



**ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**Πτυχιακή Εργασία**

# **«Εφαρμογή Διαχείρισης Ηλεκτρονικού Πρωτοκόλλου»**



του φοιτητή

**Ιωαννίδη Σάββα**

Αρ. Μητρώου:04/2595

Επιβλέπων Καθηγητής

**κ. Γιακουστίδης Κων/νος**

**Θεσσαλονίκη 2011**

## Πρόλογος

Αρκετές επιχειρήσεις και οργανισμοί ακόμη και σήμερα χρησιμοποιούν έναν χειρόγραφο τρόπο παρακολούθησης των εισερχόμενων και εξερχόμενων εγγράφων τους με αποτέλεσμα να δημιουργούνται σοβαρά προβλήματα στην οργάνωση τους. Ενδεικτικά κάποια από τα προβλήματα που μπορεί να προκύψουν κατά τον χειρισμό μεγάλου όγκου αλληλογραφίας είναι η άσκοπη σπατάλη χρόνου λόγω της μεταφοράς πλήθους εγγράφων καθημερινά μεταξύ του προσωπικού, η απαίτηση μεγάλου φυσικού χώρου αποθήκευσης για την αρχειοθέτηση των εγγράφων καθώς επίσης και το σχετικά μεγάλο χρονικό διάστημα που απαιτείται για την αναζήτηση και ανάκληση ενός εγγράφου από το φυσικό αρχείο.

Η ηλεκτρονική διαχείριση της αλληλογραφίας που έχει αρχίσει σιγά σιγά να υιοθετείται από τις επιχειρήσεις προσπαθεί να δώσει λύσει σε αυτά τα προβλήματα αντικαθιστώντας το χειρογραφικό σύστημα πρωτοκόλλου με ηλεκτρονικό. Στόχος της πτυχιακής είναι η ανάπτυξη μιας εφαρμογής διαχείρισης ηλεκτρονικού πρωτοκόλλου που να παρέχει ένα πλήθος δυνατοτήτων στον χρήστη κάνοντας πιο αποδοτική την πρωτοκόλληση της αλληλογραφίας ενός οργανισμού.

Η παρούσα πτυχιακή εργασία με τίτλο «Εφαρμογή Διαχείρισης Ηλεκτρονικού Πρωτοκόλλου» εκπονήθηκε στο πλαίσιο της πτυχιακής μου εργασίας για το τμήμα Πληροφορικής του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης.



## Περίληψη

Η διαχείριση της εισερχόμενης και εξερχόμενης αλληλογραφίας συνιστά σημαντική διαδικασία για την εύρυθμη λειτουργία κάθε οργανισμού και επιχείρησης.

Καθεστώς στο χώρο των επιχειρήσεων αποτελεί η χειρόγραφη παρακολούθηση των εισερχόμενων και εξερχόμενων εγγράφων με αποτέλεσμα σοβαρά προβλήματα και δυσλειτουργίες. Η ηλεκτρονική διαχείριση πρωτοκόλλου έρχεται να δώσει λύση σε αρκετά από αυτά τα προβλήματα.

Η ηλεκτρονική διαχείριση πρωτοκόλλου, προκειμένου να εξυπηρετεί τον σκοπό για τον οποίο αναπτύχθηκε, θα πρέπει να προσφέρει μια σειρά από δυνατότητες. Στα εισερχόμενα και εξερχόμενα έγγραφα του οργανισμού θα πρέπει να εκχωρείται αυτόματα ένας μοναδικός αριθμός πρωτοκόλλου ενώ θα πρέπει να αποθηκεύονται και σε συγκεκριμένη ηλεκτρονική τοποθεσία από την οποία θα μπορούν να ανακτηθούν. Επιπλέον, η εφαρμογή θα πρέπει να παρέχει την δυνατότητα δημιουργίας εγγράφων πρωτοκόλλου τα οποία θα βασίζονται σε πρότυπα εγγράφων. Τέλος, μια εξίσου σημαντική λειτουργία που πρέπει να παρέχεται είναι η αναζήτηση και η ανάκτηση οποιουδήποτε εγγράφου με βάση συγκεκριμένα κριτήρια.

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας υλοποιεί τις παραπάνω λύσεις με στόχο να καλύψει τα κενά της διαχείρισης αλληλογραφίας των σύγχρονων επιχειρήσεων.



## Abstract

The incoming and outgoing document management, constitutes an important process of every company and organization.

For many years, this process has been manual, something that has been creating a lot of problems and malfunctions. Electronic protocol management is able to provide solutions to a lot of these problems.

In order for electronic protocol management to serve its purpose, it needs to offer a series of functionalities. The incoming and outgoing documents should be assigned a unique protocol number and should be saved in specific areas from where they can be retrieved. Moreover, the application should provide the ability to create documents based on pre-defined templates. Finally, an equally important functionality is the ability to search and find any document based on specific criteria.

The software application that has been developed as part of the present dissertation, provides the above solutions in order to fill the gaps of protocol management in modern companies and organizations.



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Κωνσταντίνο Γιακουστίδη για την στήριξη και τη βοήθεια του στις δύσκολες στιγμές που προέκυψαν κατά την εκπόνηση αυτής της πτυχιακής εργασίας.

Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου προσέφερε σε όλη την διάρκεια των σπουδών μου.

Τέλος, νοιώθω την ανάγκη να ευχαριστήσω τον αδερφό μου που διαπίστωσε από νωρίς την κλίση μου στην πληροφορική και με «έσπρωξε» προς την επιστήμη αυτή.





## Περιεχόμενα

Πρόλογος .....	2
Περίληψη .....	4
Abstract .....	6
Ευχαριστίες .....	8
Εισαγωγή .....	12
Κεφάλαιο 1 .....	14
Εργαλεία Υλοποίησης .....	14
1.1 Visual Studio 2008 (Professional) .....	14
1.2 Visual Basic .NET .....	15
1.3 .NET Framework .....	16
1.4 Microsoft SQL Server 2008 .....	17
1.5 SQL Server Management Studio .....	19
Κεφάλαιο 2 .....	22
Σχεδίαση και υλοποίηση της βάσης δεδομένων .....	22
2.1 Πίνακες της Βάσης Δεδομένων .....	22
2.1.1 Πίνακας Document .....	22
2.1.2 Πίνακας Organization .....	24
2.1.3 Πίνακας Contacts .....	26
2.1.4 Πίνακας Department .....	28
2.1.5 Πίνακας Filetype .....	29
2.1.6 Πίνακας Configuration .....	29
2.2 Διάγραμμα ER .....	31
2.3 Συσχετίσεις Πινάκων .....	32
2.3.1 Συσχέτιση Document –Contacts .....	33
2.3.2 Συσχέτιση Document-Filetype .....	34
2.3.3 Συσχέτιση Contacts- Organization .....	34
2.3.4 Συσχέτιση Contact – Department .....	35
Κεφάλαιο 3 .....	36
Ο κώδικας της εφαρμογής .....	36
3.1 Διαχείριση Εισερχομένων Εγγράφων .....	36
3.2 Παραλαβή Εισερχόμενου Εγγράφου .....	44

3.3 Διαχείριση Εξερχόμενων Εγγράφων.....	50
3.4 Δημιουργία εξερχόμενου Εγγράφου .....	53
3.5 Διαχείριση Οργανισμών .....	55
3.6 Επεξεργασία Οργανισμού.....	58
3.7 Διαχείριση Επαφών .....	59
3.8 Επεξεργασία Επαφής .....	61
3.9 Διαχείριση Τμημάτων.....	62
3.10 Επεξεργασία Τμήματος .....	63
3.11 Διαχείριση Τύπων Αρχείου .....	64
3.12 Επεξεργασία Τύπου Αρχείων .....	66
3.13 Εισαγωγή Προτύπου .....	66
3.14 Διαμόρφωση Εφαρμογής.....	68
Προτάσεις – Συμπεράσματα .....	74
Αναφορές.....	76
Βιβλιογραφία.....	78
Οδηγός Χρήσης .....	80
Διαχείριση Οργανισμών .....	82
Δημιουργία – Επεξεργασία Οργανισμού.....	85
Διαχείριση Τμημάτων.....	87
Προσθήκη – Επεξεργασία Τμήματος .....	88
Διαχείριση Επαφών .....	90
Δημιουργία - Επεξεργασία Επαφής.....	93
Διαχείριση Τύπων Αρχείου .....	94
Δημιουργία – Επεξεργασία Τύπου Αρχείων .....	96
Διαμόρφωση Εφαρμογής.....	97
Καταχώρηση Προτύπου .....	98
Διαχείριση Εισερχόμενων Εγγράφων .....	100
Παραλαβή Εγγράφου.....	101
Διαχείριση Εξερχόμενων Εγγράφων.....	103
Δημιουργία Εγγράφου .....	105

## Εισαγωγή

Ένα σημαντικό ζήτημα που λαμβάνει χώρα σε κάθε επιχείρηση ή οργανισμό είναι η σωστή διαχείριση της εισερχόμενης και εξερχόμενης αλληλογραφίας του. Η διαχείριση πραγματοποιείται με χρήση πρωτοκόλλου στο οποίο καταγράφονται στοιχεία των εγγράφων και αποδίδεται μοναδικός αριθμός αναφοράς. Είναι γεγονός ότι το πρωτόκολλο έχει αποδειχθεί –και το έχουν αποδεχθεί όλοι- ότι είναι ο μόνος τρόπος εξασφάλισης των συναλλασσόμενων για την κατάθεση εγγράφων και για τους υπαλλήλους το μέσο παρακολούθησης και διεκπεραίωσης υποθέσεων.

Η κλασική χειρογραφική διαχείριση της εισερχόμενης και εξερχόμενης αλληλογραφίας δημιουργεί αρκετά προβλήματα. Κατ' αρχάς κάθε έγγραφο για να διακινηθεί μεταξύ του προσωπικού απαιτεί αρκετό χρόνο με αποτέλεσμα η όλη διαδικασία να γίνεται χρονοβόρα. Επίσης όταν ένα έγγραφο χρειάζεται να διανεμηθεί σε περισσότερους από έναν υπαλλήλους προϋποθέτει την δημιουργία μεγάλου φυσικού αριθμού αντιγράφων που έχει ως συνέπεια την αλόγιστη σπατάλη χαρτιού. Ένα εξίσου σημαντικό πρόβλημα που αντιμετωπίζει το χειρογραφικό σύστημα πρωτοκόλλου είναι η απαίτηση ενός σημαντικού φυσικού χώρου αποθήκευσης για την αρχειοθέτηση των εγγράφων καθώς επίσης υπάρχουν σχετικά αργές διαδικασίες αναζήτησης και ανάκλησής τους από το φυσικό αρχείο.

Πολλές φορές χάνονται έγγραφα λόγω του χειρισμού μεγάλου όγκου αλληλογραφίας και διακίνησης μεταξύ πολλαπλών εργαζομένων. Επιπρόσθετα ένας σημαντικός αριθμός προσωπικού απασχολείται για την διαχείριση του πρωτοκόλλου αφού καθημερινά διακινείται μεγάλος όγκος αλληλογραφίας μεταξύ των εργαζομένων. Γενικότερα, το εργασιακό περιβάλλον υποβαθμίζεται σημαντικά, λόγω του μεγάλου αριθμού εγγράφων και φακέλων που διακινούνται από γραφείο σε γραφείο.

Μια αποτελεσματική λύση στην χειρογραφική διαχείριση των εγγράφων έρχεται να δώσει η ηλεκτρονική διαχείριση πρωτοκόλλου. Η ηλεκτρονική διαχείριση πρωτοκόλλου επιτυγχάνει αυτοματοποίηση της διαχείρισης των εξερχομένων και εισερχομένων εγγράφων, άμεση παρακολούθηση και εύκολη ανάκλησή τους

μέσω του σχήματος αρχειοθέτησης που σχετικά προσαρμόζεται και χρησιμοποιείται.

Η υιοθέτηση εφαρμογής ενός συστήματος ηλεκτρονικής διαχείρισης πρωτοκόλλου προσφέρει τόσο στον οργανισμό όσο και στο προσωπικό αρκετά πλεονεκτήματα. Αρχικά, ελαχιστοποιείται σε πολύ μεγάλο βαθμό η διακίνηση έντυπου υλικού μεταξύ των τμημάτων ενός οργανισμού. Επίσης, υπάρχει σημαντική βελτίωση της ταχύτητας αναζήτησης των εγγράφων καθώς και ελαχιστοποίηση των τυχών καθυστερήσεων λαμβάνοντας υπόψη τον όγκο των διακινούμενων εγγράφων. Η αποθήκευση των εγγράφων γίνεται με πιο ασφαλή τρόπο και διασφαλίζεται το περιεχόμενο σημαντικών εγγράφων. Επιπλέον μειώνεται ο απαιτούμενος χώρος αποθήκευσης των εγγράφων αφού πλέον τα έγγραφα υπάρχουν σε ηλεκτρονική μορφή.

Η εφαρμογή που αναπτύχθηκε κατά την εκπόνηση της πτυχιακής έχει σαν στόχο την υλοποίηση ενός συστήματος ηλεκτρονικής διαχείρισης πρωτοκόλλου το οποίο παρέχει τα πλεονεκτήματα που αναφέρθηκαν παραπάνω.

Τα κεφάλαια της πτυχιακής εργασίας διαμορφώνονται ως εξής:

Στο κεφάλαιο 1 γίνεται αναλυτική περιγραφή των εργαλείων που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής. Το κεφάλαιο 2 παρουσιάζει την βάση δεδομένων που χρησιμοποιείται από την ίδια την εφαρμογή κάνοντας αναφορά στους πίνακες καθώς και στις συσχετίσεις μεταξύ των πινάκων ενώ στο κεφάλαιο 3 περιγράφονται τα σπουδαιότερα κομμάτια κώδικα της εφαρμογής

## Κεφάλαιο 1

### Εργαλεία Υλοποίησης

Στο κεφάλαιο αυτό παρουσιάζονται συνοπτικά τα εργαλεία και οι τεχνολογίες που χρειάστηκαν για να αναπτυχθεί η εφαρμογή. Για κάθε ένα από αυτά αναφέρονται οι βασικές λειτουργίες τους καθώς και τα χαρακτηριστικά τους. Το Visual Studio 2008 είναι το εργαλείο με το οποίο δημιουργήθηκε ο κώδικας της εφαρμογής και ο SQL Server 2008 το εργαλείο που χρησιμοποιήθηκε για την «κατασκευή» της βάσης δεδομένων που χρειάζεται για τη λειτουργία της.

#### 1.1 Visual Studio 2008 (Professional)

Για την σύνταξη του κώδικα της εφαρμογής επιλέχθηκε το Visual Studio 2008. Με τη βοήθεια του Visual Studio μπορεί κάποιος να δημιουργήσει εφαρμογές οι οποίες τρέχουν σε περιβάλλοντα Windows και διαδικτυακές εφαρμογές. Οι εφαρμογές που δημιουργούνται με την χρήση του είναι φιλικές προς τον χρήστη. Σε αυτό συμβάλλει το γεγονός ότι το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment) – IDE.



Από την πλευρά του προγραμματιστή το Visual Studio παρέχει πολλά εργαλεία και δυνατότητες ώστε να μπορεί να κάνει εύκολα εφαρμογές με γραφικό και φιλικό περιβάλλον. Εκτός από τον μεγάλο αριθμό εργαλείων που διαθέτει και είναι ήδη υλοποιημένα, αν ένας προγραμματιστής θέλει να προσθέσει περισσότερα εργαλεία ή να δημιουργήσει κάποια δικά του, έχει τη δυνατότητα να το κάνει, αφού το Visual Studio επιτρέπει την προσθήκη επιπλέον plug –ins.

Μια άλλη δυνατότητα που προσφέρει το Visual Studio στους προγραμματιστές που το επιλέγουν είναι ευελιξία στην επιλογή της γλώσσας προγραμματισμού

που θέλουν. Το Visual Studio δεν περιορίζει τον προγραμματιστή να γράψει σε κάποια συγκεκριμένη γλώσσα προγραμματισμού αλλά ο προγραμματιστής μπορεί να επιλέξει την γλώσσα προγραμματισμού που θέλει από αυτές που υποστηρίζει το Visual Studio. Η γλώσσα που θα επιλέξει ο κάθε προγραμματιστής δεν θα πρέπει να τον απασχολεί από την πλευρά των δυνατοτήτων που έχει αφού το Visual Studio δίνει της ίδιες δυνατότητες σε όλες τις γλώσσες που υποστηρίζει. Κάποιες από τις γλώσσες προγραμματισμού που υποστηρίζει είναι : η Visual C#, η Visual Basic.NET και η Visual C/C++, από πλευράς εφαρμογών, η HTML/XHTML, JavaScript και CSS από πλευράς διαδικτυακών εφαρμογών και τέλος από πλευράς βάσεων δεδομένων υποστηρίζεται η XML.

## 1.2 Visual Basic .NET

Η Visual Basic .NET ανήκει στις αντικειμενοστραφείς γλώσσες προγραμματισμού και θεωρείται η εξέλιξη της κλασσικής Visual Basic. Στις ονομασία της προστέθηκε και το .NET γιατί οι προγραμματιστές που την προτιμούν χρησιμοποιούν τις βιβλιοθήκες του .NET για να προγραμματίσουν, αφού πλέον δεν διαθέτει δικές της βιβλιοθήκες.



Για να προγραμματίσει κάποιος σε Visual Basic .NET θα πρέπει να διαθέτει και κάποια έκδοση του Visual Studio. Ο λόγος που πρέπει να γίνεται αυτό είναι γιατί η Visual Basic .NET δεν είναι μια ανεξάρτητη γλώσσα αλλά πλέον ανήκει στην «οικογένεια» του Visual Studio και πιο συγκεκριμένα του .NET. Το γεγονός αυτό δεν επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν την Visual Basic

.NET για να αναπτύξουν εφαρμογές οι οποίες θα τρέχουν σε άλλα λειτουργικά συστήματα εκτός από τα Windows και το διαδίκτυο.

Πολλοί είναι αυτοί που διαφωνούν με τη θεωρία ότι η Visual Basic .NET είναι εξέλιξη της κλασσικής Visual Basic. Αρκετοί πιστεύουν ότι είναι δύο διαφορετικές γλώσσες προγραμματισμού. Η άποψη αυτή προέρχεται από το γεγονός ότι η δύο γλώσσες προγραμματισμού δεν έχουν κανέναν κοινό εκτός από την σύνταξη του κώδικα τους. Όπως έχει αναφερθεί και προηγουμένως η Visual Basic .NET χρησιμοποιεί τη βιβλιοθήκη του .NET και όχι τη βιβλιοθήκη της κλασσικής Visual Basic. Εκτός από αυτό η Visual Basic .NET δεν ανήκει πλέον στην κατηγορία των even –driven γλωσσών προγραμματισμού, στις οποίες ανήκει η Visual Basic, αλλά μετατράπηκε σε μια αντικειμενοστραφή γλώσσα προγραμματισμού.

### 1.3 .NET Framework

Το .NET Framework είναι ένα πλαίσιο λογισμικού το οποίο έχει αναπτυχθεί από την Microsoft και μπορεί να εγκατασταθεί σε υπολογιστές που φέρουν λειτουργικό σύστημα Microsoft Windows. Το .NET Framework αποτελείται από δύο κύρια στοιχεία: Το Common Language Runtime (CLR) και τη βιβλιοθήκη κλάσεων του .NET Framework. Τα προγράμματα που είναι γραμμένα με το .NET Framework εκτελούνται σε ένα περιβάλλον λογισμικού που είναι γνωστό ως Common Language Runtime (CLR). Το CLR είναι μια εφαρμογή εικονικής μηχανής με την οποία οι προγραμματιστές δεν χρειάζεται να εξετάσουν τις δυνατότητες του συγκεκριμένου επεξεργαστή που θα εκτελεστεί το πρόγραμμα. Παρέχει βέβαια και άλλες εξίσου σημαντικές υπηρεσίες όπως είναι η ασφάλεια, η διαχείριση μνήμης και ο χειρισμός εξαιρέσεων.





Η βιβλιοθήκη κλάσεων, το άλλο βασικό στοιχείο του .NET Framework, παρέχει μια πληθώρα χαρακτηριστικών και συγκεκριμένα περιλαμβάνει διεπαφές χρήστη, πρόσβαση σε δεδομένα, σύνδεση με βάση δεδομένων, κρυπτογραφία, ανάπτυξη εφαρμογών διαδικτύου, αριθμητικούς αλγορίθμους και δίκτυα επικοινωνίας. Χρησιμοποιείται από τους προγραμματιστές για να δημιουργήσουν τις δικές του εφαρμογές συνδυάζοντας τα παραπάνω χαρακτηριστικά με τον δικό τους κώδικα.

Το .NET Framework έχει ως σκοπό του να χρησιμοποιείται από τις περισσότερες νέες εφαρμογές που δημιουργούνται για την πλατφόρμα των Windows. Για την ανάπτυξη των εφαρμογών αυτών απαιτείται να έχει εγκατασταθεί μια νέα έκδοση του (.NET Framework 4 ή νεότερη).

#### **1.4 Microsoft SQL Server 2008**

Για την σχεδίαση της βάσης δεδομένων της εφαρμογής, επιλέχθηκε σαν εργαλείο το Microsoft SQL Server 2008. Ο SQL Server είναι ένα Σχεσιακό Μοντέλο Διαχείρισης Βάσεων Δεδομένων (RDBMS). Η κύρια μονάδα αποθήκευσης του SQL Server είναι μια βάση δεδομένων η οποία αποτελείται από μια συλλογή πινάκων και κώδικα. Ο SQL Server διαθέτει τις δικές τους γλώσσες προγραμματισμού, με τις οποίες μπορούν οι διαχειριστές τους να τον χειριστούν. Οι γλώσσες αυτές είναι η T-SQL και η ANSI SQL.

Στη βάση δεδομένων του SQL Server μπορούν να αποθηκευτούν δεδομένα πολλών διαφορετικών τύπων. Αν για κάποιο λόγο η πληροφορία που θέλει ο διαχειριστής της βάσης δεδομένων να αποθηκεύσει δεν υποστηρίζεται από τους

βασικούς τύπους, μπορεί να δημιουργήσει τους δικούς του σύνθετους τύπους δεδομένων. Οι τύποι αυτοί (UDTs) βασίζονται στους βασικούς τύπους του «SQL Server» αλλά μπορούν να τροποποιηθούν. Τα στοιχεία μιας βάσης δεδομένων στον SQL Server αποθηκεύονται σε ένα ή περισσότερα αρχεία με κατάληξη .mdf. Ο χώρος αποθήκευσης που δίνεται σε μια βάση δεδομένων διαιρείται διαδοχικά σε αριθμημένες σελίδες των 8 KB.



Ο κεντρικός διακομιστής SQL επιτρέπει να χρησιμοποιούν την ίδια βάση πολλοί χρήστες ταυτόχρονα. Για να το πετύχει αυτό χρησιμοποιεί τις λεγόμενες «συναλλαγές» (transactions), οι οποίες εξασφαλίζουν ότι οποιαδήποτε λειτουργία γίνει πάνω στα δεδομένα είτε ολοκληρώνεται συνολικά είτε αποτυγχάνει συνολικά. Ο κεντρικός διακομιστής SQL δεν επιτρέπει ποτέ η βάση να μείνει σε μια ενδιάμεση κατάσταση ώστε να μπορεί να εγγυηθεί την ακεραιότητα των δεδομένων που υπάρχουν στη βάση.

Ο SQL Server επιπλέον περιλαμβάνει μια σειρά από πρόσθετες υπηρεσίες. Οι υπηρεσίες αυτές δεν είναι απαραίτητες για τη λειτουργία του συστήματος της βάσης δεδομένων αλλά παρέχουν υπηρεσίες προστιθέμενης αξίας πάνω από το κεντρικό σύστημα διαχείρισης της βάσης δεδομένων. Αυτές οι υπηρεσίες λειτουργούν είτε ως μέρος κάποιου στοιχείου (Component) του SQL Server, είτε αυτόνομα ως υπηρεσία των Windows η οποία χρησιμοποιεί το δικό της API για να αλληλεπιδρά με αυτές. Οι υπηρεσίες του SQL Server είναι οι ακόλουθες:

- **Service Broker:** τρέχει ως μέρος της μηχανής βάσεως δεδομένων και παρέχει μια αξιόπιστη πλατφόρμα ανταλλαγής μηνυμάτων για τις εφαρμογές SQL Server.
- **Replication Services:** χρησιμοποιούνται από τον SQL Server για να συγχρονίσουν τις βάσεις δεδομένων του είτε ολοκληρωτικά, είτε ένα

υποσύνολο των βασικών αντικείμενων. Υπάρχουν τουλάχιστον 4 διαφορετικοί μέθοδοι για τον συγχρονισμό. Οι υπηρεσίες συγχρονισμού βοηθούν και σε περιπτώσεις ολοκληρωτικής καταστροφής της μητρικής βάσης δεδομένων, δίνοντας την δυνατότητα ακόμα και για αυτόματη λειτουργία της δευτερεύουσας συγχρονισμένης βάσης δεδομένων.

- **Analysis Services:** προσφέρει την online επεξεργασία δεδομένων και υποστηρίζει τη λειτουργία εξόρυξης δεδομένων (data mining) για business intelligent εφαρμογές.
- **Reporting Services:** επιβλέπει μέσω μιας διαδικτυακής διεπαφής (web interface) τα δεδομένα που έχουν συγκεντρωθεί από μια SQL Server βάση δεδομένων.
- **Notification Services:** δίνουν τη δυνατότητα στο χρήστη να παράγει μηνύματα ηλεκτρονικού ταχυδρομείου με πολύ λίγο κόστος.
- **Integration Services:** χρησιμοποιούνται για να ενσωματώσουν στοιχεία από διαφορετικές πηγές δεδομένων.
- **Full Text Search Engine:** προσφέρει εξειδικευμένη αναζήτηση κειμένου που είναι αποθηκευμένο σε μια SQL Server βάση δεδομένων. Για παράδειγμα κείμενα Word και pdf που είναι αποθηκευμένα στον SQL Server μπορούν να αναζητηθούν πολύ γρήγορα.

## 1.5 SQL Server Management Studio

Οι εκδόσεις του SQL Server από την έκδοση του 2005 και έπειτα συνοδεύονται με ένα εργαλείο. Το εργαλείο αυτό είναι το SQL Server Management Studio, το οποίο δημιουργήθηκε από τη Microsoft ώστε να βοηθήσει τους χρήστες του SQL Server να μπορέσουν να διαχειριστούν αυτόν και τα συστατικά του. Ένα SQL Server Management Studio δεν μπορεί να διαχειριστεί οποιαδήποτε έκδοση του SQL Server αλλά μόνο μια συγκεκριμένη. Για κάθε έκδοση του SQL Server υπάρχει και το αντίστοιχο SQL Server Management Studio.

Το SQL Server Management Studio διαθέτει έναν scrip editor και διάφορα άλλα γραφικά εργαλεία τα οποία εργάζονται με τα αντικείμενα και τα χαρακτηριστικά του SQL Server. Ο εξερευνητής που υπάρχει στο SQL Server Management

Studio είναι ένα από τα βασικά χαρακτηριστικά του. Με τη βοήθεια του εξερευνητή αυτού ο χρήστης μπορεί εύκολα να αναζητήσει κάποιο αντικείμενο μέσα στον SQL Server ώστε να ενεργήσει πάνω σε αυτό.



Πριν την εμφάνιση του SQL Server Management Studio η διαχείριση του SQL Server ήταν αρκετά πιο δύσκολη υπόθεση, αφού όλα έπρεπε να γίνονται «χειροκίνητα» χωρίς να υπάρχουν κάποια εργαλεία για βοήθεια. Το γραφικό περιβάλλον του SQL Server Management Studio έκανε την δουλειά των διαχειριστών του SQL Server εύκολη και γρήγορη υπόθεση.



## Κεφάλαιο 2

### Σχεδίαση και υλοποίηση της βάσης δεδομένων

Μια βάση δεδομένων είναι ένα σύστημα που προορίζεται για να οργανώνει, να αποθηκεύει και να ανακτά μεγάλες ποσότητες δεδομένων εύκολα. Τις περισσότερες φορές η βάση δεδομένων αποτελεί ένα από τα κύρια συστατικά μιας εφαρμογής της οποίας ο σκοπός είναι να χειρίζεται δεδομένα. Η σχεδίαση της αποτελεί μια επίπονη εργασία και πρέπει να γίνεται με προσεκτικό και μελετημένο τρόπο.

Στο κεφάλαιο που ακολουθεί θα εξετάσουμε αναλυτικά την βάση δεδομένων που δημιουργήθηκε για την συγκεκριμένη εφαρμογή. Η βάση δεδομένων υλοποιήθηκε με γνώμονα την απλότητα, την ευχρηστία, την πληρότητα αλλά και την έλλειψη επανάληψης πληροφορίας. Αρχικά, θα ερμηνευτεί η χρησιμότητα κάθε πίνακα και ακολούθως θα παρατεθεί το σχεσιακό μοντέλο στο οποίο φαίνονται όλοι οι πίνακες με τις επιμέρους ιδιότητές τους καθώς και οι σχέσεις μεταξύ των πινάκων.

#### 2.1 Πίνακες της Βάσης Δεδομένων

Κάθε βάση δεδομένων αποτελείται από πίνακες στους οποίους αποθηκεύονται τα δεδομένα που χρειάζεται η εφαρμογή για την λειτουργία της. Στο υποκεφάλαιο αυτό θα περιγραφούν αναλυτικά οι πίνακες που χρειάστηκαν για την υλοποίηση της εφαρμογής.

##### 2.1.1 Πίνακας Document

**Πεδία:** IdProtocol, NumberProtocol, DocumentType, IdSender, IdReceiver, Date, Subject, Keywords, Path, Filename, IdFiletype.

Ο πίνακας document αποτελεί τον βασικό πίνακα της βάσης δεδομένων. Στον πίνακα αυτό καταχωρούνται τα στοιχεία (οι πληροφορίες) που σχετίζονται με

κάθε έγγραφο της εφαρμογής . Το έγγραφο μπορεί να είναι είτε εισερχόμενο είτε εξερχόμενο χωρίς όμως να απαιτείται ξεχωριστός πίνακας για κάθε τύπο εγγράφου.

Επειδή η εφαρμογή έχει σαν κύριο χαρακτηριστικό της την διαχείριση των εγγράφων η δομή και τα πεδία του συγκεκριμένου πίνακα αποτέλεσαν έντονο σημείο προβληματισμού. Αναλυτικά τα πεδία του πίνακα και ο ρόλος τους στην βάση δεδομένων :

- **IdProtocol:** Το πεδίο αυτό ορίζεται ως πρωτεύον κλειδί. Η τιμή του θέτεται αυτόματα από την βάση και είναι ένας αύξων αριθμός. Η ύπαρξη του έγκειται στο γεγονός ότι πρέπει να υπάρχουν μοναδικές εγγραφές στον πίνακα των εγγράφων.
- **NumberProtocol:** Αποτελεί τον αριθμό πρωτοκόλλου που δίνεται σε κάθε έγγραφο. Έχει την μορφή «αύξων αριθμός – χρονολογία». Κάθε έγγραφο που δημιουργείται (εξερχόμενο) παίρνει τον αμέσως επόμενο διαθέσιμο αριθμό και σε συνδυασμό με την χρονολογία δημιουργίας του σχηματίζεται ο αριθμός πρωτοκόλλου που θα του δοθεί. Με τον ίδιο τρόπο σχηματίζεται και ο αριθμός πρωτοκόλλου των εισερχόμενων εγγράφων. Εδώ θα πρέπει να αναφέρουμε ότι η αρίθμηση των εγγράφων είναι ξεχωριστή για τα εισερχόμενα και τα εξερχόμενα. Δηλαδή μπορεί να υπάρχει εισερχόμενο και εξερχόμενο έγγραφο με τον ίδιο αριθμό πρωτοκόλλου. Αυτό δεν δημιουργεί κάποιο πρόβλημα αφού ο τύπος του εγγράφου είναι διαφορετικός.
- **DocumentType:** Παίρνει την τιμή 1 (ένα) ή 0 (μηδέν) ανάλογα με τον τύπο του εγγράφου που καταχωρείται στον πίνακα. Αν το έγγραφο είναι εισερχόμενο τότε ο τύπος του εγγράφου είναι 1 (ένα) ενώ αν το έγγραφο είναι εξερχόμενο ο τύπος του είναι 0 (μηδέν).
- **IdSender:** Κάθε έγγραφο είτε εισερχόμενο είτε εξερχόμενο έχει ένα παραλήπτη και ένα αποστολέα. Στο πεδίο αυτό καταχωρείτε ο κωδικός του αποστολέα που σχετίζεται με το έγγραφο. Τόσο ο αποστολέας, όσο και ο παραλήπτης του εγγράφου αποτελούν επαφές του οργανισμού ή της επιχείρησης που χρησιμοποιεί την εφαρμογή. Περισσότερα για τις επαφές θα αναφερθούν στο υποκεφάλαιο Πίνακας Contacts.

- **IdReceiver:** Αποτελεί τον κωδικό του παραλήπτη του εγγράφου. Δεν μπορεί ο παραλήπτης ενός εγγράφου να είναι ίδιος με τον αποστολέα του.
- **Date:** Η ημερομηνία κατά την οποία αποθηκεύεται το έγγραφο στη βάση δεδομένων, καταχωρείται στο πεδίο αυτό. Την ημερομηνία δεν τη δίνει ο χρήστης αλλά δίνεται αυτόματα από τον σύστημα.
- **Subject:** Είναι ένα κείμενο με το οποίο ο χρήστης δίνει ένα τίτλο στο έγγραφο που θέλει να καταχωρήσει.
- **Keywords:** Για να είναι πιο εύκολη η αναζήτηση μεταξύ των εγγράφων δημιουργήθηκε στον πίνακα ένα πεδίο στο οποίο θα καταχωρούνται διάφορες λέξεις - κλειδιά που σχετίζονται με το έγγραφο και θα βοηθήσουν τον χρήστη να το αναζητήσει και να το βρει πιο εύκολα. Οι λέξεις - κλειδιά που θα καταχωρούνται στο πεδίο αυτό είναι αποκλειστική ευθύνη του χρήστη. Το πεδίο μπορεί να μην πάρει τιμές.
- **Path:** Τα έγγραφα δεν αποθηκεύονται αυτούσια στη βάση δεδομένων, αλλά σε φακέλους που υπάρχουν σε κάποιον υπολογιστή. Στη βάση δεδομένων αποθηκεύεται η διεύθυνση (διαδρομή) στην οποία βρίσκεται αποθηκευμένο το έγγραφο, ώστε σε περίπτωση που θέλει ο χρήστης να το χειριστεί να μπορεί οποιαδήποτε στιγμή να το βρει.
- **Filename:** Στο πεδίο αυτό καταχωρείται το όνομα το οποίο δόθηκε στο έγγραφο όταν αποθηκεύεται στη διεύθυνση που αναφέραμε προηγουμένως. Το όνομα που δίνεται στο κάθε αρχείο είναι ο αριθμός πρωτοκόλλου του ώστε να υπάρχει μια ομοιομορφία στην ονομασία των εγγράφων.
- **IdFiletype:** Το πεδίο αυτό αποτελεί τον κωδικό ο οποίος αναφέρεται στον τύπο του αρχείου (εγγράφου). Το πεδίο αυτό σχετίζεται με τον πίνακα Filetype που θα αναλυθεί στη συνέχεια.

### 2.1.2 Πίνακας Organization

**Πεδία:** IdOrganization, OrgName, OrgAddress, OrgTelephone, OrgTown, OrgPostCode, OrgCountry.



Κάθε επιχείρηση ή οργανισμός που διαχειρίζεται έγγραφα (εισερχόμενα ή εξερχόμενα) συνεργάζεται, επικοινωνεί με κάποιους άλλους οργανισμούς ή επιχειρήσεις. Οι οργανισμοί αυτοί ομαδοποιούνται και καταχωρούνται στον πίνακα organization. Στον πίνακα καταχωρούνται τα στοιχεία του κάθε οργανισμού τα οποία χρειάζεται η εταιρεία για να επικοινωνεί μαζί του. Επειδή επικοινωνία και ανταλλαγή εγγράφων μπορεί να υπάρχει και μέσα στην ίδια την εταιρεία καλό είναι στον πίνακα των οργανισμών να καταχωρείται και ο οργανισμός ή η εταιρεία που χρησιμοποιεί την εφαρμογή.

Ο πίνακας organization έχει και μια επιπλέον χρήση. Με την βοήθεια του μπορούμε να ομαδοποιήσουμε τις επαφές μας με βάση τους οργανισμούς στους οποίους ανήκουν. Τα πεδία που υπάρχουν στον πίνακα των Οργανισμών είναι τα ακόλουθα:

- **IdOrganization:** Αποτελεί το κύριο κλειδί του πίνακα Organization. Η τιμή δεν δίνεται από τον χρήστη της εφαρμογής αλλά αυτόματα από τη βάση δεδομένων. Η τιμή του είναι αριθμητική και αποτελεί τον αμέσως επόμενο αριθμό από αυτόν που έχει ο τελευταίος οργανισμός που καταχωρήθηκε.
- **OrgName:** Ο κάθε οργανισμός με τον οποίο συνεργάζεται η εταιρεία που χρησιμοποιεί την εφαρμογή έχει κάποια επωνυμία. Η επωνυμία αυτή καταχωρείται στο συγκεκριμένο πεδίο. Είναι σημαντικό να υπάρχει η πραγματική ονομασία του οργανισμού και όχι κάποιο τυχαίο κείμενο στο πεδίο αυτό, για να είναι πιο εύκολη η επεξεργασία των οργανισμών, των επαφών και των εγγράφων που σχετίζονται με τον κάθε οργανισμό.
- **OrgAddress:** Στο πεδίο αυτό αποθηκεύεται η διεύθυνση του εκάστοτε οργανισμού. Το πεδίο αυτό δεν είναι υποχρεωτικό.
- **OrgTelephone:** Το πεδίο αυτό αντιπροσωπεύει ένα από τα τηλέφωνα του οργανισμού. Όπως και το προηγούμενο πεδίο δεν είναι υποχρεωτικό.
- **OrgTown:** Απαραίτητο δεν είναι ούτε το πεδίο αυτό, γιατί δεν έχει κάποιο λειτουργικό ρόλο στην ίδια την εφαρμογή, αλλά είναι καθαρά πεδίο πληροφοριών.
- **OrgPostCode:** Μια επιπλέον πληροφορία που μπορεί ο χρήστης της εφαρμογής να καταχωρήσει για τον οργανισμό είναι ο ταχυδρομικός κωδικός. Από μόνο του το πεδίο αυτό δεν δίνει κάποια σημαντική πληροφορία αλλά σε συνδυασμό με την διεύθυνση του οργανισμού

μπορεί να αποτελέσει κάποιο χρήσιμο στοιχείο. Το πεδίο δεν είναι υποχρεωτικό.

- **OrgCountry:** Στο πεδίο αυτό, το οποίο δεν είναι υποχρεωτικό, καταχωρείται η χώρα στην οποία εδρεύει ο οργανισμός. Η πληροφορία αυτή θα βοηθήσει τον χρήστη της εφαρμογής, συνδυάζοντας την με την διεύθυνση, την πόλη, και τον ταχυδρομικό κωδικό, να έρθει σε επαφή με τον οργανισμό μέσω ταχυδρομείου ή να αποστείλει κάτι στον οργανισμό αυτό.

### 2.1.3 Πίνακας Contacts

**Πεδία:** IdContact, SurnameCon, NameCon, EmailCon, AddressCon, TownCon, PostCodeCon, MobilePhoneCon, HomePhoneCon, IdOrganization, IdDepartment.

Η «ανταλλαγή» εγγράφων και η επικοινωνία γίνεται είτε μέσα στην ίδια την εταιρεία είτε με άλλους εξωτερικούς οργανισμούς. Αν το δούμε όμως λίγο πιο ειδικά, η επικοινωνία δεν γίνεται μεταξύ των οργανισμών αλλά μεταξύ των υπαλλήλων τους. Για τον λόγο αυτό δημιουργήθηκε ο πίνακας των επαφών. Μέσα στον πίνακα αυτό αποθηκεύονται πληροφορίες που αφορούν τις επαφές (τα πρόσωπα) με τις οποίες σχετίζεται η εταιρεία.

Στον πίνακα αυτό πρέπει να καταχωρηθούν και οι υπάλληλοι της εταιρείας που χειρίζονται την εφαρμογή ώστε να μπορεί η εταιρεία να γνωρίζει ποιος είναι ο αποστολέας ενός εξερχόμενου εγγράφου και ποιος ο παραλήπτης ενός εισερχόμενου. Εκτός από αυτό με την εισαγωγή των υπαλλήλων της εταιρείας στον πίνακα μπορεί να υλοποιηθεί και η αποστολή εγγράφων εσωτερικά στην εταιρεία. Αναλυτικά τα πεδία του πίνακα είναι:

- **IdContact:** Το πεδίο αυτό είναι το κύριο κλειδί του πίνακα. Οι τιμές που παίρνει δίνονται αυτόματα από τη βάση δεδομένων και η μια είναι συνέχεια της άλλης. Ο σκοπός που υπάρχει το πεδίο αυτό είναι για να μπορούμε να αναφερθούμε σε κάποια επαφή από άλλον πίνακα της βάσης δεδομένων.

- **SurnameCon:** Σε μια επαφή το πιο σημαντικό στοιχείο που χρειάζεται να καταχωρείται είναι φυσικά το ονοματεπώνυμο της. Στο πεδίο αυτό εισάγεται το επώνυμο της επαφής και είναι ένα από τα πεδία του πίνακα αυτού που είναι υποχρεωτικά να καταχωρηθούν.
- **NameCon:** Εκτός από το επώνυμο της επαφής είναι σημαντικό να γνωρίζουμε και το όνομα της. Αυτή η πληροφορία καταχωρείται στο πεδίο αυτό. Όπως είναι λογικό αυτή η πληροφορία είναι απαραίτητη να καταχωρηθεί στον πίνακα.
- **EmailCon:** Αν η επικοινωνία που έχει ο χρήστης με κάποια επαφή είναι μέσω ηλεκτρονικού ταχυδρομείου τότε είναι χρήσιμο να καταχωρήσει στον πίνακα το email της επαφής. Το email της κάθε επαφής υπάρχει στο πεδίο αυτό του πίνακα.
- **AddressCon:** Στο πεδίο αυτό του πίνακα καταχωρείται η διεύθυνση της επαφής. Σε αρκετές περιπτώσεις, ειδικά όταν η επαφή δεν ανήκει στον οργανισμό που χειρίζεται την εφαρμογή, η διεύθυνση της επαφής πιθανόν είναι ίδια με την διεύθυνση του οργανισμού στον οποίο ανήκει. Η πληροφορία αυτή όμως θεωρήθηκε καλό να βρίσκεται και σε αυτόν τον πίνακα για δύο λόγους. Ο πρώτος είναι για τις επαφές του οργανισμού που χειρίζεται την εφαρμογή και ο δεύτερος για να υπάρχει πιο εύκολη πρόσβαση στην πληροφορία όταν ο χρήστης γνωρίζει σε ποιά επαφή θέλει να στείλει κάποιο έγγραφο.
- **TownCon:** Το πεδίο αυτό από μόνο του, ίσως να μην αποτελεί κάποια σημαντική πληροφορία, αφού κρατά την πόλη της επαφής. Αν όμως συνδυαστεί με την διεύθυνση της επαφής τότε δημιουργείται μια ίσως χρήσιμη πληροφορία. Για τις επαφές που ανήκουν σε άλλους οργανισμούς (όχι στην ίδια την εταιρεία), η τιμή που θα πάρει το πεδίο αυτό πιθανών να είναι η ίδια με αυτή που θα πάρει το αντίστοιχο πεδίο του οργανισμού στον οποίο ανήκουν.
- **PostCodeCon:** Ο ταχυδρομικός κώδικας που καταχωρείται στο πεδίο αυτό είναι μια πληροφορία που είναι συνέχεια των δύο προηγούμενων και ισχύουν αντίστοιχα όσα αναφέρθηκαν σε αυτά.
- **MobilePhoneCon:** Ένα σημαντικό πεδίο αν η επικοινωνία με μια επαφή είναι στενή και συχνή, αφού σε αυτό το πεδίο αποθηκεύεται το κινητό

τηλέφωνο της επαφής. Πολύ σημαντικό πεδίο για τις επαφές του ίδιου του οργανισμού.

- **HomePhoneCon:** Η τιμή που παίρνει το πεδίο αυτό είναι ένα σταθερό τηλέφωνο στο οποίο μπορεί κάποιος να βρει την επαφή. Το πεδίο αυτό μπορεί να μείνει κενό αν δεν υπάρχει η συγκεκριμένη πληροφορία.
- **IdOrganization:** Κάθε επαφή ανήκει σε κάποιον οργανισμό. Ελάχιστες φορές (μπορεί και καθόλου) υπάρχουν επαφές με τις οποίες ένας οργανισμός ανταλλάσσει έγγραφα και επικοινωνεί, που δεν ανήκουν σε κάποιον άλλον οργανισμό. Στο πεδίο αυτό μπαίνει το κύριο κλειδί του οργανισμού στον οποίο ανήκει η επαφή.
- **IdDepartment:** Εκτός από τον οργανισμό στον οποίο ανήκει, μια επαφή ανήκει και σε κάποιο τμήμα. Με το πεδίο αυτό, όπως και με το προηγούμενο μπορούν εύκολα να ομαδοποιηθούν οι επαφές για καλύτερη αναζήτηση και διαχείριση. Το πεδίο αυτό παίρνει σαν τιμή τον κωδικό του τμήματος που έχει στον αντίστοιχο πίνακα.

#### 2.1.4 Πίνακας Department

**Πεδία:** IdDepartment, DepName.

Πολλές φορές ο οργανισμός στον οποίο ανήκει κάποια επαφή δεν είναι αρκετή πληροφορία. Ειδικά για τις επαφές που ανήκουν στο οργανισμό που χρησιμοποιεί την εφαρμογή. Για τον λόγο αυτό δημιουργήθηκε ο πίνακας Department στον οποίο καταχωρούνται τα διάφορα τμήματα που μπορεί να έχει ένας οργανισμός. Δεν θεωρήθηκε αναγκαίο να διαχωριστούν τα τμήματα ανάλογα με τους οργανισμούς στους οποίους ανήκουν επειδή είναι λίγο δύσκολο να γνωρίζει ο χρήστης της εφαρμογής τα τμήματα που διαθέτει ο κάθε οργανισμός με τον οποίο συνεργάζεται η εταιρεία.

Ο κύριος σκοπός της δημιουργίας του πίνακα αυτού είναι για να γίνεται καλύτερη ομαδοποίηση των επαφών καθώς και των εγγράφων. Με την βοήθεια του πίνακα αυτού, ο οποίος σχετίζεται άμεσα με τον πίνακα των επαφών, ο χρήστης μπορεί να παρακολουθεί τα έγγραφα που δημιουργούνται από κάποιο τμήμα καθώς και αυτά που εισέρχονται σε αυτό. Τα πεδία του πίνακα αναλυτικά είναι:

- **IdDepartment:** Αποτελεί τον κωδικό του τμήματος και το κύριο κλειδί του πίνακα. Όπως και στους υπόλοιπους πίνακες, έτσι και εδώ ο κωδικός δίνεται αυτόματα από τη βάση δεδομένων.
- **DepName:** Στο πεδίο αυτό καταχωρείται το όνομα του τμήματος. Το όνομα είναι υποχρεωτικό μιας και είναι η μόνη πληροφορία που κρατάμε για το κάθε τμήμα.

### 2.1.5 Πίνακας Filetype

**Πεδία:** IdFiletype, Extension, Application.

Ο πίνακας Filetype αποτελεί ένα πληροφοριακό πίνακα. Ο πίνακας αυτός δίνει πληροφορίες για τον τύπο του κάθε αρχείου που είναι αποθηκευμένο στη βάση δεδομένων. Η δημιουργία του πίνακα αυτού έγινε για να βοηθήσει την ομαδοποίηση των εγγράφων. Τα δεδομένα που υπάρχουν στον πίνακα αυτό θα πρέπει να καταχωρηθούν από τον χρήστη της εφαρμογής. Τα πεδία του πίνακα είναι τα ακόλουθα:

- **IdFiletype:** το πεδίο αυτό είναι ο κωδικός του τύπου αρχείων και το κύριο κλειδί του πίνακα. Ο κωδικός αυτός δεν δίνεται από τον χρήστη αλλά αυτόματα από τη βάση δεδομένων. Ο πίνακας αυτός με τη βοήθεια του συγκεκριμένου πεδίου συνδέεται με τον πίνακα Document.
- **Extension:** Ο τύπος του κάθε αρχείου συνήθως χωρίζεται με βάση την κατάληξη που έχει. Αυτή η πληροφορία βρίσκεται στο πεδίο αυτό .
- **Application:** Κάθε αρχείο ανάλογα με την κατάληξη που έχει ανοίγει και μπορεί να επεξεργαστεί με την βοήθεια κάποιας εφαρμογής. Οι διάφορες εφαρμογές που μπορεί να ανοίγουν τα αρχεία της εταιρείας αποθηκεύονται στο πεδίο αυτό.

### 2.1.6 Πίνακας Configuration

**Πεδία:** PKey, SavePath, IncomingFolderName, OutgoingFolderName, GeneralFolderName, TemplateFolderName.

Τα έγγραφα που καταχωρούνται ή δημιουργούνται από την εφαρμογή, πρέπει να αποθηκεύονται σε κάποια διεύθυνση ενός υπολογιστή. Η αποθήκευση αυτή είναι αναγκαία για να μπορεί ο χρήστης, οποιανδήποτε στιγμή να ανοίξει το έγγραφο αυτό και να το δει ή να το εκτυπώσει. Επειδή ανάλογα με την εταιρεία που χρησιμοποιεί την εφαρμογή η διεύθυνση που θέλει να αποθηκεύσει τα έγγραφα της, αλλάζει, αφήσαμε στην ίδια την εταιρεία τη δυνατότητα να επιλέξει τη διεύθυνση αποθήκευσης των εγγράφων και την ονομασία που θα πάρουν οι φάκελοι που θα τα ομαδοποιήσουν. Οι πληροφορίες αυτές αποθηκεύονται στον πίνακα Configuration ο οποίος δεν σχετίζεται με κανένα άλλο πίνακα της βάσης δεδομένων αλλά υπάρχει για να κρατάει της διαμορφώσεις της κάθε εταιρείας που χρησιμοποιεί την εφαρμογή. Ένα χαρακτηριστικό του πίνακα αυτού είναι ότι έχει μόνο μια γραμμή και πολλές στήλες. Ο πίνακας αναλυτικά έχει τα παρακάτω πεδία:

- **PKey:** Το πεδίο αυτό αποτελεί το κύριο κλειδί του πίνακα. Δεν χρησιμοποιείται για αναφορά σε κανένα άλλο πίνακα στη βάση δεδομένων, ούτε για να ξεχωρίσει τις εγγραφές του πίνακα αυτού (αφού υπάρχει μόνο μια εγγραφή στον πίνακα αυτό). Ο λόγος ύπαρξης του είναι γιατί χρειάζεται ο πίνακας ένα κύριο κλειδί.
- **SavePath:** Η διεύθυνση στην οποία επιλέξει η εταιρεία ότι θέλει να αποθηκεύει τα αρχεία της καταχωρείται στο πεδίο αυτό. Στην διεύθυνση αυτή θα δημιουργηθεί από την εφαρμογή ένας γενικός φάκελος στον οποίο θα υπάρχει ολόκληρο το πρωτόκολλο της εταιρείας.
- **GeneralFolderName:** Στην διεύθυνση που θα επιλέξει η εταιρεία να δημιουργηθεί το πρωτόκολλό της, θα δημιουργηθεί ένας γενικός φάκελος μέσα στον οποίο θα χωρίζονται τα εισερχόμενα και τα εξερχόμενα έγγραφα. Ο λόγος που γίνεται αυτό είναι για να είναι τα έγγραφα τις εταιρείας πιο ομαδοποιημένα. Μπορεί στη διεύθυνση που θα επιλεγεί να υπάρχουν πολλοί και διάφοροι φάκελοι. Με τον τρόπο αυτό δεν θα υπάρχουν απλά δυο φάκελοι με τα εισερχόμενα και τα εξερχόμενα έγγραφα αλλά θα υπάρχει ένας γενικός φάκελος που θα βοηθήσει στην καλύτερη ομαδοποίηση. Το όνομα του γενικού αυτού φακέλου θα το δώσει η εταιρεία.

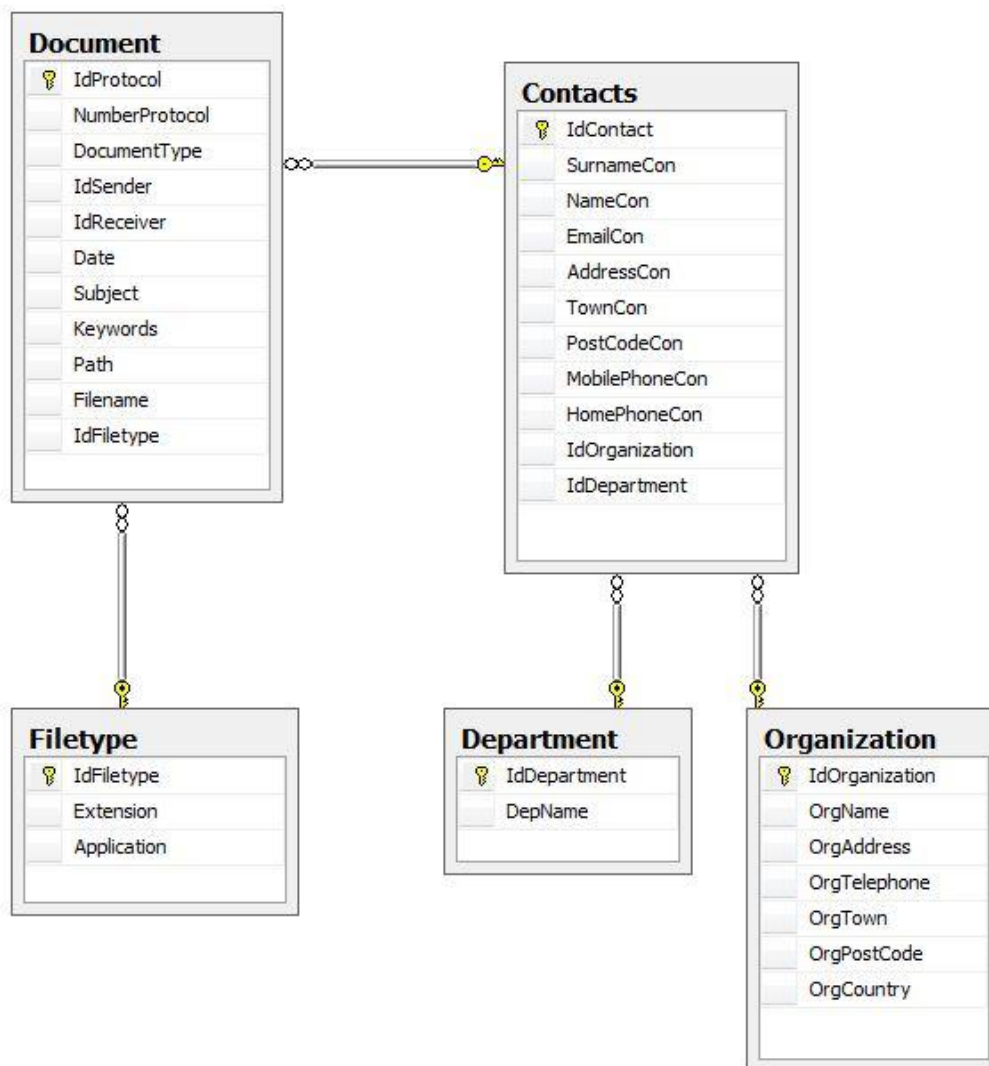
- **IncomingFolderName:** Μέσα στον γενικό φάκελο που θα αποθηκεύεται το πρωτόκολλο θα υπάρχουν και δύο φάκελοι. Ο ένας για τα εισερχόμενα έγγραφα και ο άλλος για τα εξερχόμενα. Στο πεδίο αυτό του πίνακα θα καταχωρείται το όνομα του φακέλου που θα επιλέξει η εταιρεία που χρησιμοποιεί την εφαρμογή, να δώσει στον φάκελο των εισερχομένων εγγράφων.
- **OutgoingFolderName:** Το πεδίο αυτό αποθηκεύει το όνομα του φακέλου των εξερχόμενων εγγράφων. Όπως και τα προηγούμενα πεδία, έτσι και αυτό, το όνομα το επιλέγει η ίδια η εταιρεία.
- **TemplateFolderName:** Για να δημιουργηθεί ένα εξερχόμενο έγγραφο, ο χρήστης της εφαρμογής, θα πρέπει αν επιλέξει ένα από τα πρότυπα εγγράφων που υπάρχουν. Τα πρότυπα εγγράφων δεν αποθηκεύονται στη βάση δεδομένων αλλά σε μια διεύθυνση κάποιου υπολογιστή. Η διεύθυνση στην οποία θέλει η εταιρεία να αποθηκεύονται τα πρότυπα των εγγράφων της, υπάρχει στον πεδίο αυτό.

## 2.2 Διάγραμμα ER

Για να δημιουργηθούν οι πίνακες της βάσης δεδομένων θα πρέπει προηγουμένως να γίνει κάποια προεργασία. Η προεργασία αυτή προϋποθέτει την ανάλυση του προβλήματος που καλείται να αντιμετωπίσει η εφαρμογή και η προσπάθεια να βρεθεί ο τρόπος για να λυθεί το πρόβλημα αυτό. Ένα από τα σημεία που πρέπει να δημιουργηθούν κατά την ανάλυση του προβλήματος είναι το διάγραμμα ER της βάσης δεδομένων. Δηλαδή ποιοί πίνακες θα πρέπει να δημιουργηθούν, τι πεδία θα έχει ο κάθε ένας και πως θα συσχετίζονται μεταξύ τους.

Το διάγραμμα ER δημιουργήθηκε μετά από αρκετή σκέψη και η τελική μορφή που έχει δεν ήταν η μορφή που είχε στην αρχή. Παράλληλα με την ανάπτυξη της εφαρμογής χρειάστηκε αρκετές φορές να αλλαχθεί και το διάγραμμα ώστε να προσαρμοστεί στα δεδομένα και στις απαιτήσεις της.

Η τελική μορφή του διαγράμματος ER της βάσης δεδομένων είναι η ακόλουθη, στην οποία φαίνονται όλοι οι πίνακες της βάσης και τα πεδία που έχουν.



Θα παρατηρήσετε ότι στο διάγραμμα δεν εμφανίζεται ο πίνακας configuration. Ο λόγος που δεν εμφανίζεται είναι γιατί ο πίνακας αυτός υπάρχει αποκλειστικά για βοηθητικούς λόγους και δεν επηρεάζει τα δεδομένα και το σχήμα της βάσης δεδομένων.

### 2.3 Συσχετίσεις Πινάκων

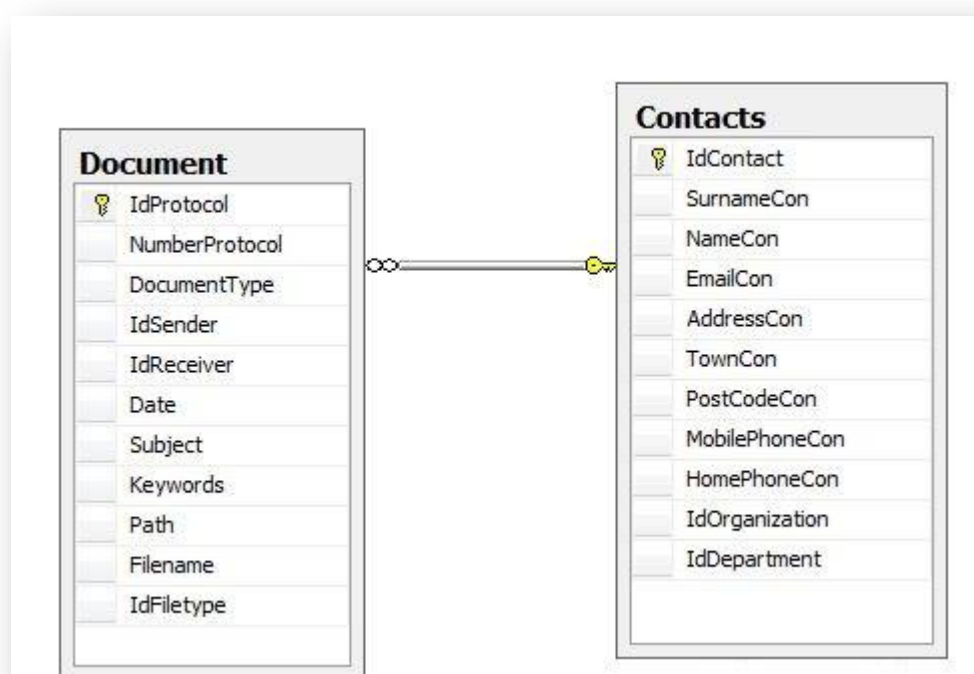
Στο προηγούμενο υποκεφάλαιο παρουσιάστηκε συνοπτικά το διάγραμμα ER της βάσης δεδομένων, χωρίς να γίνει αναφορά στις συσχετίσεις που υπάρχουν



μεταξύ των πινάκων. Παρόλο που έγινε αναλυτική αναφορά στους πίνακες και στα πεδία του στην αρχή του κεφαλαίου, θεωρήθηκε καλό να αναλυθούν περισσότερο οι συσχετίσεις που υπάρχουν ανάμεσα τους. Η ανάλυση αυτή θα γίνει στο παρόν υποκεφάλαιο.

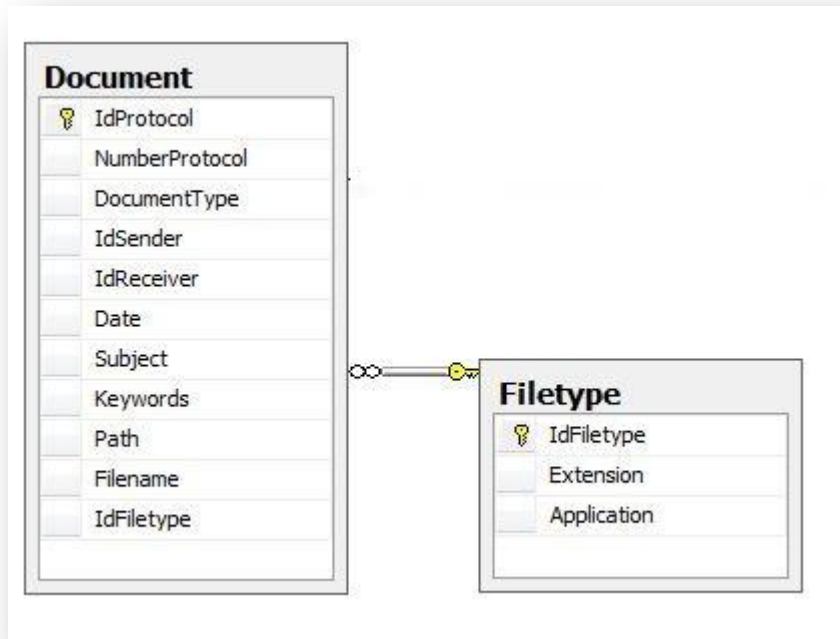
### 2.3.1 Συσχέτιση Document –Contacts

Η εικόνα που εμφανίζεται παρακάτω παρουσιάζει την συσχέτιση που γίνεται μεταξύ του πίνακα των εγγράφων (Document) και του πίνακα των επαφών (Contacts). Όπως αναφέραμε και στην αρχή του κεφαλαίου αυτού (όταν αναλύσαμε τους πίνακες της βάσης δεδομένων), κάθε έγγραφο έχει ένα παραλήπτη και έναν αποστολέα. Ο συσχετισμός αυτός του εγγράφου με κάποια/κάποιες επαφές φαίνεται στην παρακάτω εικόνα. Όπως εύκολα μπορεί να εννοηθεί, με τη συσχέτιση ανάμεσα στους δύο πίνακες δεν μπορεί να διαγραφεί καμία επαφή, από την στιγμή που σχετίζεται με κάποιο έγγραφο.



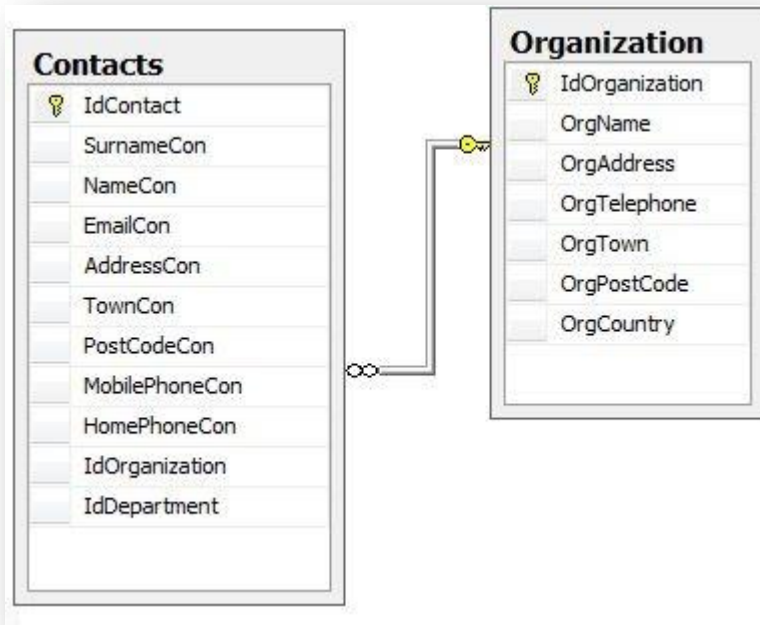
### 2.3.2 Συσχέτιση Document-Filetype

Εκτός από τον παραλήπτη και τον αποστολέα, ένα έγγραφο (ηλεκτρονικό) έχει και κάποιον τύπο. Ο τύπος αυτός αφορά στην εφαρμογή με την οποία μπορεί κάποιος να ανοίξει το ηλεκτρονικό έγγραφο. Στην εικόνα παρακάτω φαίνεται η συσχέτιση των δύο πινάκων.



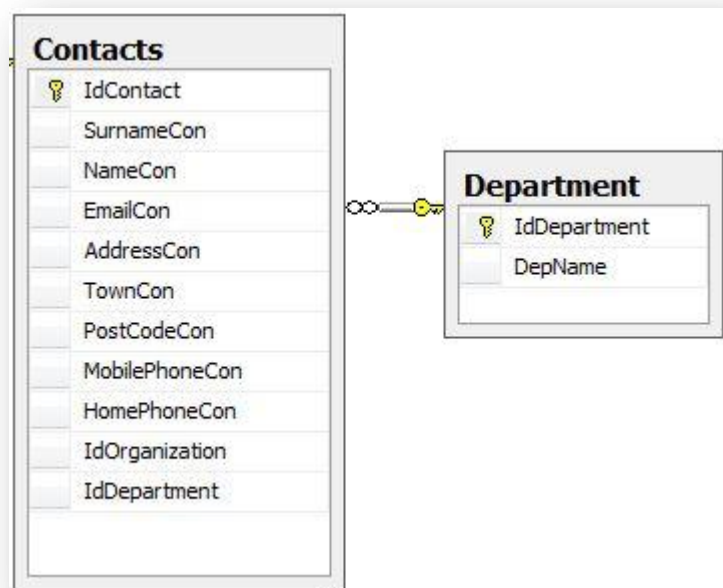
### 2.3.3 Συσχέτιση Contacts- Organization

Η κάθε επαφή που καταχωρείται στην εφαρμογή θα πρέπει να αντιστοιχεί σε κάποια εταιρεία ή οργανισμό με τον οποίο συνεργάζεται η εταιρεία που χρησιμοποιεί την εφαρμογή. Κάθε επαφή μπορεί να ανήκει μόνο σε έναν οργανισμό. Ενώ από την πλευρά του οργανισμού, πολλές επαφές μπορούν να σχετίζονται με αυτό. Η συσχέτιση που υπάρχει ανάμεσα στους πίνακες Contacts και Organization φαίνεται στην εικόνα που ακολουθεί:



### 2.3.4 Συσχέτιση Contact - Department

Εκτός από τον οργανισμό με τον οποίο σχετίζεται, μια επαφή ανήκει και σε ένα τμήμα. Όπως και στον οργανισμό, ένα τμήμα μπορεί να έχει πολλές επαφές, ενώ μια επαφή μπορεί να ανήκει μόνο σε ένα τμήμα.



## Κεφάλαιο 3

### Ο κώδικας της εφαρμογής

Στα προηγούμενο κεφαλαίο έγινε αναφορά στους πίνακες που δημιουργήθηκαν στη βάση δεδομένων για να μπορέσουν να διαχειριστούν τα έγγραφα μιας εταιρείας. Η διαχείριση όμως των εγγράφων δεν γίνεται από τους πίνακες της βάσης δεδομένων αλλά από την εφαρμογή που δημιουργήθηκε και χρησιμοποιεί μια βάση δεδομένων για να πετύχει τους στόχους της.

Η εφαρμογή αυτή στην ουσία μέσω του κώδικα που υλοποιήθηκε ενημερώνει τους διάφορους πίνακες της βάσης δεδομένων, με τα αντίστοιχα στοιχεία που αποθηκεύουν. Για την ενημέρωση του κάθε πίνακα δημιουργήθηκαν ξεχωριστές φόρμες στη εφαρμογή. Επίσης φόρμες υπάρχουν και για την εμφάνιση των στοιχείων του κάθε πίνακα της βάσης δεδομένων.

Στο κεφάλαιο αυτό θα δούμε πώς οι διάφορες φόρμες της εφαρμογής χειρίζονται τα διάφορα δεδομένων των πινάκων. Από κάθε φόρμα θα γραφτεί ο κώδικας που κρίνεται πιο σημαντικός.

#### 3.1 Διαχείριση Εισερχομένων Εγγράφων

Για την καλύτερη διαχείριση και παρακολούθηση των εγγράφων δημιουργήθηκαν διαφορετικές φόρμες για τα εισερχόμενα και τα εξερχόμενα έγγραφα αντίστοιχα. Στην φόρμα της διαχείρισης των εισερχομένων εγγράφων, ο χρήστης μπορεί να βλέπει και να χειρίζεται τα εισερχόμενα έγγραφα που έχει η εταιρεία. Ο κώδικας με τον οποίο γίνονται όλες αυτές οι ενέργειες θα περιγραφεί παρακάτω.

Στην φόρμα αυτή υπάρχει ένα Data Grid View, μια Toolbar και ένα πλαίσιο αναζήτησης. Μέσα στο Data Grid View εμφανίζονται τα εισερχόμενα έγγραφα που υπάρχουν στη βάση δεδομένων και συγκεκριμένα στον πίνακα Document. Το Data Grid View είναι συνδεδεμένο με τον πίνακα Document μέσω ενός adapter. Για να μπορέσουν να εμφανιστούν μόνο τα εισερχόμενα έγγραφα, με το που ανοίξει η φόρμα αυτή, καλείται μια μέθοδος η οποία με ένα μικρό Select στον

πίνακα Document παίρνει τα έγγραφα που έχουν ένα συγκεκριμένο τύπο (DocumentType) και ενημερώνει τον adapter που είναι συνδεδεμένος με το Data Grid View.

```
Dim SelectQuery As String = "Select * From Document WHERE  
DocumentType=1"  
  
Private Sub LoadIncomingFiles()  
  
    Try  
        Dim cmd As SqlCommand  
  
        cmd = New SqlCommand(SelectQuery, connection)  
  
        Me.DocumentTableAdapter.Adapter.SelectCommand = cmd  
        Me.ProtocolDataSet.Document.Clear()  
        Me.DocumentTableAdapter.Adapter.Fill(Me.ProtocolDataSet, "Document")  
  
        Me.DocumentBindingSource.DataSource = Me.ProtocolDataSet.Document  
  
        Me.DocumentDataGridView.DataSource = Me.DocumentBindingSource  
  
        Me.DocumentBindingNavigator.BindingSource =  
Me.DocumentBindingSource  
  
        Me.DocumentBindingNavigator.Update()  
    Catch ex As Exception  
        MessageBox.Show(ex.Message)  
    End Try  
  
    data_are_loaded = True  
  
End Sub
```

Η Toolbar που υπάρχει στη φόρμα είναι και αυτή συνδεδεμένη με το πίνακα Document και το Data Grid View της φόρμας. Στην toolbar υπάρχουν τα βελάνια μετακίνησης στα δεδομένα που εμφανίζονται στο Data Grid View. Επιπλέον υπάρχει κουμπί για την προσθήκη νέου εγγράφου. Το πλήκτρο αυτό ανοίγει μια ξεχωριστή φόρμα στην οποία γίνεται αναλυτικά η παραλαβή του εγγράφου και δημιουργεί μια καινούργια γραμμή στον πίνακα Document.

```
Private Sub ToolStripButtonAddNew_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ToolStripButtonAddNew.Click  
    If IncomingFileEdit.Visible = True Then  
        IncomingFileEdit.WindowState = FormWindowState.Normal  
        IncomingFileEdit.Focus()  
        ToolStripButtonAddNew.Enabled = False  
        ToolStripButtonEdit.Enabled = False  
    Else
```

```

IncomingFileEdit.MdiParent = MainForm
IncomingFileEdit.Show()
IncomingFileEdit.NewIncomingFile(SelectQuery)
End If

```

```
End Sub
```

Στην φόρμα που ανοίγει για την παραλαβή του εγγράφου, καλείται η δημόσια μέθοδος `NewIncomingFile` η οποία παίρνει σαν όρισμα ένα `select` ερώτημα. Το ερώτημα αυτό είναι για να εμφανιστούν στη φόρμα `IncomingFileEdit` τα ίδια έγγραφα που εμφανίζονται και στο `Data Grid View`.

Στην διαχείριση των εγγράφων δεν επιτρέπεται στον χρήστη να διαγράψει κάποιο έγγραφο. Για αυτό τον λόγο το πλήκτρο της διαγραφής πάνω στην `toolbar` δεν υπάρχει. Επιτρέπεται όμως η τροποποίηση κάποιων στοιχείων που σχετίζονται με το έγγραφο. Το πλήκτρο της τροποποίησης ανοίγει την ίδια φόρμα που ανοίγει και το πλήκτρο της παραλαβής νέου εγγράφου. Η διαφορά εδώ είναι ότι καλεί διαφορετική μέθοδο στη φόρμα αυτή, ώστε να θέσει τη φόρμα σε κατάσταση επεξεργασίας εγγράφου.

```

Private Sub ToolStripButtonEdit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripButtonEdit.Click

    If DocumentDataGridView.SelectedRows.Count > 0 Then
        Dim SelectedRowsQuery As String
        SelectedRowsQuery = "SELECT * FROM Document WHERE DocumentType=1
AND IdProtocol IN ("
        For i As Integer = 0 To DocumentDataGridView.SelectedRows.Count - 1
            SelectedRowsQuery &=
DocumentDataGridView.SelectedRows(i).Cells(ColumnIdProtocol.Index).Value
& ", "
        Next

        If SelectedRowsQuery.EndsWith(",") Then
            SelectedRowsQuery =
SelectedRowsQuery.Remove(SelectedRowsQuery.Length - 1, 1)
        End If
        SelectedRowsQuery &= ")"
        IncomingFileEdit.MdiParent = MainForm
        IncomingFileEdit.Show()
        IncomingFileEdit.LoadEditIncomingFile(SelectedRowsQuery, 0)
    Else
        IncomingFileEdit.MdiParent = MainForm
        IncomingFileEdit.Show()
        IncomingFileEdit.LoadEditIncomingFile(SelectQuery, 0)
    End If
End Sub

```

Ένα άλλο εργαλείο που υπάρχει στην toolbar είναι το Refresh το οποίο έχει παρόμοια λειτουργία με αυτή που έχει η μέθοδος LoadIncomingFiles. Επειδή δεν υπάρχει κάποια σημαντική αλλαγή δεν θα δείξουμε τον κώδικα.

Ο χρήστης μπορεί από την φόρμα αυτή να ανοίξει οποιοδήποτε έγγραφο επιλέξει και να το δει. Επειδή τα έγγραφα είναι πιθανόν να είναι διαφορετικού είδους, δηλαδή να ανοίγουν με διαφορετική εφαρμογή, η εντολή που γράφτηκε καλύπτει αυτή την δυνατότητα.

```
Private Sub ToolStripButtonOpenFile_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripButtonOpenFile.Click

    If DocumentDataGridView.SelectedRows.Count = 1 Then
        Dim FileFullName As String
        FileFullName =
DocumentDataGridView.SelectedRows(0).Cells(ColumnPath.Index).Value & "\"
        FileFullName &=
DocumentDataGridView.SelectedRows(0).Cells(ColumnFilename.Index).Value
        Try
            System.Diagnostics.Process.Start(FileFullName)
        Catch ex As Exception
            MessageBox.Show("Δημιουργήθηκε κάποιο πρόβλημα κατά την εμφάνιση
του εγγράφου", "Εμφάνιση Εγγράφου", MessageBoxButtons.OK)
        End Try
    End If

    If DocumentDataGridView.SelectedRows.Count < 1 Then
        MessageBox.Show("Δεν έχετε επιλέξει κάποιο έγγραφο", "Εμφάνιση
Εγγράφου", MessageBoxButtons.OK)
    End If

    If DocumentDataGridView.SelectedRows.Count > 1 Then
        MessageBox.Show("Έχετε επιλέξει περισσότερα από ένα έγγραφα.
Επιλέξτε μόνο ένα έγγραφο για να συνεχίσετε.", "Εμφάνιση Εγγράφου",
MessageBoxButtons.OK)
    End If
End Sub
```

Εκτός από αυτά που περιγράψαμε πιο πάνω για καλύτερη παρακολούθηση των εγγράφων υπάρχει η δυνατότητα της αναζήτησης. Στη φόρμα υπάρχει ένα πλαίσιο με αρκετά πεδία που μπορεί ο χρήστης να κάνει αναζήτηση. Ο κώδικας της αναζήτησης ανάλογα με τα πεδία που επιλέξει ο χρήστης, είναι το πιο δύσκολο και ταυτόχρονα πιο μεγάλο κομμάτι κώδικα της φόρμας. Στον κώδικα

αυτό γίνεται προσπάθεια να δημιουργηθεί το select ερώτημα που θα γίνει στη βάση δεδομένων ανάλογα με τις επιλογές του χρήστη.

```
Private Sub BtnSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnSearch.Click

    Dim cmd As SqlCommand
    Dim query As String
    query = "SELECT Document.* FROM Document"

    If LBoxOrganization.SelectedItems.Count > 0 Then
        query = query & ",Contacts "
    End If

    query = query & " WHERE DocumentType=1 AND "
```

Στο ερώτημα που θέλουμε να δημιουργήσουμε θέλουμε να πάρουμε όλα τα πεδία των εγγραφών από τον πίνακα Document όπου ο τύπος του εγγράφου είναι «1», δηλαδή τα έγγραφα που είναι εισερχόμενα. Αν ο χρήστης έχει επιλέξει στην αναζήτηση κάποιον οργανισμό, δηλαδή θέλει να βρει τα έγγραφα που ήρθαν στην εταιρεία από κάποιον/κάποιους συγκεκριμένους οργανισμούς, τότε θα πρέπει στην αναζήτηση να ενταχθεί και ο πίνακας Contacts ώστε η αναζήτηση με βάση τον οργανισμό να γίνει με την βοήθεια του οργανισμού στον οποίο ανήκει ο αποστολέας.

Αφού δημιουργηθεί το κομμάτι From από το select (η δημιουργία του φαίνεται και περιγράφεται παραπάνω) πρέπει να δημιουργηθεί και το κομμάτι Where. Το κομμάτι αυτό δημιουργείται ανάλογα με τις τιμές που έδωσε ο χρήστης στα ανάλογα πεδία. Τα πεδία αυτά, που εξετάζονται για τις τιμές τους στη συνέχεια, αφορούν πεδία του πίνακα Document.

```
If txtProtocolNumber.Text <> "" Then
    Dim input As String = txtProtocolNumber.Text
    input = input.Replace("*", "%")
    query = query & "NumberProtocol Like '" & input & "' AND "
End If

If txtFileName.Text <> "" Then
    Dim input As String = txtFileName.Text
    input = input.Replace("*", "%")
    query = query & "Filename Like '" & input & "' AND "
End If

If TxtSubject.Text <> "" Then
    Dim input As String = TxtSubject.Text
```



```

input = input.Replace("*", "%")
query = query & "Subject Like '" & input & "' AND "
End If

If TxtKeyWords.Text <> "" Then
    Dim KeyWords As String
    KeyWords = TxtKeyWords.Text
    KeyWords = KeyWords.Replace("*", "%")
    KeyWords = KeyWords.Replace(", ", "%")
    KeyWords = KeyWords.Replace("-", "%")
    KeyWords = KeyWords.Replace("_", "%")
    KeyWords = KeyWords.Replace("/", "%")
    KeyWords = KeyWords.Replace(" ", "%")
    query = query & " Keywords Like '%" & KeyWords & "%' AND "
End If

```

Σε αυτό το σημείο θα πρέπει να αναφέρουμε στο πώς δημιουργούμε το κομμάτι του select που αφορά το πεδίο KeyWords. Επειδή δεν δόθηκε συγκεκριμένο μοτίβο με το οποίο ο χρήστης, κατά την παραλαβή ή την δημιουργία του εγγράφου, να εισάγει λέξεις κλειδιά προσπαθήσαμε να δημιουργήσουμε όλες τις περιπτώσεις με τις οποίες υπάρχει περίπτωση να διαχωρίσει τις λέξεις κλειδιά του. Έτσι αντικαθιστούμε κάθε πιθανό τελεστή διαχωρισμού των λέξεων κλειδιών, που σκεφτήκαμε ότι μπορεί να χρησιμοποιήσει ο χρήστης, με το σύμβολο «%». Το σύμβολο αυτό στη βάση δεδομένων του SQL Server προσδιορίζει οποιονδήποτε χαρακτήρα.

```

If TxtDateFrom.Text <> "" Then
    Dim input As String = Format(Convert.ToDateTime(TxtDateFrom.Text),
"yyyy-MM-dd ")
    query &= "Date >= '" & input & "' AND "
End If

If TxtDateTo.Text <> "" Then
    Dim input As String = Format(Convert.ToDateTime(TxtDateFrom.Text),
"yyyy-MM-dd ")
    query &= "Date <= '" & input & "' AND "
End If

```

Στον χρήστη δίνεται η δυνατότητα να κάνει αναζήτηση με βάση την ημερομηνία την οποία καταχωρήθηκε το έγγραφο στη βάση δεδομένων. Την ημερομηνία δεν την γράφει με το χέρι αλλά την επιλέγει από ένα Month Calendar. Το Month Calendar όμως επιστρέφει την ημερομηνία με την μορφή που την έχει ο υπολογιστής του εκάστοτε χρήστη. Στην SQL Server όμως η ημερομηνία καταχωρείται με έναν συγκεκριμένο τρόπο. Έτσι για να αποφευχθούν τυχόν

προβλήματα που μπορεί να προκύψουν, η ημερομηνία που επιλέγει ο χρήστης τροποποιείται και γίνεται στο ίδιο Format με αυτό της βάσης δεδομένων.

```
If LBoxOrganization.SelectedItems.Count > 0 Then
    query = query & " Contacts.IdContact=Document.IdSender AND"
    query = query & " Contacts.IdOrganization IN ("

    For Each obj In LBoxOrganization.SelectedItems
        query = query & obj.item(0).ToString & ","
    Next

    If query.EndsWith(",") Then
        query = query.Remove(query.Length - 1)
    End If

    query = query & ")"
End If
```

Ο κώδικας που φαίνεται παραπάνω εκτελείται μόνο όταν επιλεγεί για αναζήτηση ένας ή περισσότεροι οργανισμοί. Αν γίνει μια τέτοια επιλογή θα πρέπει όπως αναφέρθηκε και προηγουμένως να συσχετιστούν οι οργανισμοί που επιλέχθηκαν με τον οργανισμό των αποστολών των εισερχόμενων εγγράφων. Η συσχέτιση αυτή γίνεται με τον κώδικα που εμφανίζεται πιο πάνω. Αυτό που κάνει ο κώδικας είναι να παίρνει τους αποστολείς των εισερχόμενων εγγράφων και να ελέγχει αν ο οργανισμός στον οποίο ανήκουν υπάρχει μέσα στους επιλεγμένους για αναζήτηση οργανισμούς.

Αν ο χρήστης εκτός από οργανισμός επιλέξει για την αναζήτηση και τμήμα τότε θα πρέπει να εκτελεστεί ανάλογος κώδικας ώστε να εμφανιστούν τα εισερχόμενα έγγραφα των οποίων οι παραλήπτες ανήκουν στο/στα τμήματα που επέλεξε ο χρήστης. Θα προσέξετε ότι για τον έλεγχο ως προς το τμήμα του παραλήπτη δημιουργείται ένα ξεχωριστό Select το οποίο με την βοήθεια του Intersect εμφανίζει τα σωστά έγγραφα. Ο κώδικας που εκτελεί σε αυτή την περίπτωση φαίνεται παρακάτω.

```
If query.EndsWith("AND ") Then
    query = query.Remove(query.Length - 4)
End If

If LBoxDepartment.SelectedItems.Count > 0 Then

    query = query & " INTERSECT "
```

```

query = query & " SELECT Document.* FROM Document,Contacts WHERE
DocumentType=1 AND "

query = query & " Contacts.IdContact=Document.IdReceiver AND"
query = query & " Contacts.IdDepartment IN ("

For Each obj In LBoxDepartment.SelectedItems
    query = query & obj.item(0).ToString & ","
Next

If query.EndsWith(",") Then
    query = query.Remove(query.Length - 1)
End If

query = query & ")"
End If

```

Αφού δημιουργηθεί το Select ερώτημα με βάση τα κριτήρια που έδωσε ο χρήστης στην αναζήτηση, θα πρέπει το ερώτημα αυτό να εκτελεστεί στη βάση δεδομένων. Τα αποτελέσματα που θα επιστρέψει το ερώτημα αυτό θα πρέπει να συνδεθούν με τους σωστούς adapter και το Data Grid View ώστε να μπορούν να εμφανιστούν στην φόρμα.

```

SelectQuery = query
cmd = New SqlCommand(query, connection)

Try

    Me.DocumentTableAdapter.Adapter.SelectCommand = cmd
    Me.ProtocolDataSet.Document.Clear()
    Me.DocumentTableAdapter.Adapter.Fill(Me.ProtocolDataSet, "Document")

    Me.DocumentBindingSource.DataSource = Me.ProtocolDataSet.Document

    Me.DocumentDataGridView.DataSource = Me.DocumentBindingSource

    Me.DocumentBindingNavigator.BindingSource = Me.DocumentBindingSource

    Me.DocumentBindingNavigator.Update()

Catch ex As Exception
    MessageBox.Show("Δημιουργήθηκε κάποιο πρόβλημα κατά την εμφάνιση των
αποτελεσμάτων της αναζήτησης.", "Πρόβλημα εμφάνισης Αποτελεσμάτων",
MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

End Sub

```

Με το κομμάτι αυτό τελειώνει και ο κώδικας για την φόρμα διαχείρισης των εισερχομένων. Ο κώδικας που αναφέρθηκε δεν είναι ολόκληρος ο κώδικας της φόρμας αλλά αυτός που θεωρήθηκε πιο σημαντικός.

### 3.2 Παραλαβή Εισερχόμενου Εγγράφου

Για την παραλαβή ενός εισερχόμενου εγγράφου δημιουργήθηκε μια ξεχωριστή φόρμα. Στη φόρμα αυτή ο χρήστης πρέπει να συμπληρώσει όλα τα πεδία που είναι υποχρεωτικά ώστε να μπορέσει να καταχωρήσει τα στοιχεία του εγγράφου στη βάση δεδομένων και να αποθηκεύσει το έγγραφο στο φάκελο των εισερχομένων.

Στην φόρμα αυτή υπάρχει μια Toolbar και τα στοιχεία που πρέπει να καταχωρήσει ο χρήστης εμφανίζονται με την μορφή Details. Τα δεδομένα και τα στοιχεία που υπάρχουν στη φόρμα είναι συνδεδεμένα με τη βοήθεια των adapter με τον πίνακα Document. Η ίδια φόρμα χρησιμοποιείται και για τροποποίηση κάποιων στοιχείων ενός εγγράφου. Η διαφορά είναι ότι όταν η φόρμα ανοίγει για παραλαβή εγγράφου, καλείται μια συγκεκριμένη μέθοδος ενώ όταν ανοίγει για επεξεργασία εγγράφου, καλείται μια άλλη μέθοδος. Οι μέθοδοι αυτοί είναι και οι δύο δημόσιες και καλούνται από τις φόρμες ή τα μενού τα οποία ζητούν να ανοιχτεί η φόρμα παραλαβής/επεξεργασίας εγγράφου.

```
Public Sub NewIncomingFile(ByVal SelectQuery As String)
    Try
        Dim cmd As SqlCommand
        cmd = New SqlCommand(SelectQuery, connection)
        Me.DocumentTableAdapter.Adapter.SelectCommand = cmd
        Me.DocumentTableAdapter.Adapter.Fill(Me.ProtocolDataSet.Document)
        Me.DocumentBindingSource.AddNew()

        idSender = Nothing
        idReceiver = Nothing

        idSenderComboBox.SelectedValue = 0
        idReceiverComboBox.SelectedValue = 0

        DateDateTimePicker.Enabled = True
        btnBrowseFile.Enabled = True

        setProtocolNumber()
    End Try
End Sub
```

```

        DocumentTypeCheckBox.CheckState = CheckState.Checked
        DateDateTimePicker.Value = Now
        data_are_Loaded = True
        moved = False
        DisableNavigationItems()
    Catch ex As Exception

    End Try

End Sub

```

Η παραπάνω μέθοδος καλείται για να προσαρμόσει την φόρμα ώστε να γίνει μια καινούργια καταχώρηση εγγράφου. Η μέθοδος παίρνει σαν όρισμα ένα Select ερώτημα για να εμφανίζει τα ίδια έγγραφα με αυτά που εμφανίζονται στη φόρμα της διαχείρισης των εισερχόμενων εγγράφων. Αν η φόρμα αυτή ανοιχτεί από το μενού της εφαρμογής τότε θα εμφανίζονται όλα τα εισερχόμενα έγγραφα που υπάρχουν στη βάση δεδομένων.

Στο σώμα της μεθόδου δημιουργείται μια νέα γραμμή στη βάση δεδομένων στην οποία θα καταχωρηθούν τα στοιχεία του εγγράφου. Επιπλέον στο νέο έγγραφο που θα καταχωρηθεί δίνεται ένας αριθμός πρωτοκόλλου. Ο αριθμός αυτός δίνεται μέσα στη μέθοδο setProtocolNumber η οποία θα εξηγηθεί στη συνέχεια. Τέλος για να αποφευχθούν κάποια προβλήματα που μπορεί να υπάρχουν με την ταυτόχρονη παραλαβή περισσότερων του ενός εγγράφου, καλείται αντίστοιχη μέθοδος για να απενεργοποιήσει κάποια από τα πλήκτρα που υπάρχουν στην toolbar.

```

Public Sub LoadEditIncomingFile(ByVal SelectQuery As String, ByVal
RowIndex As Integer)
    DateDateTimePicker.Enabled = False
    btnBrowseFile.Enabled = False

    Try
        Dim cmd As SqlCommand
        cmd = New SqlCommand(SelectQuery, connection)

        Me.DocumentTableAdapter.Adapter.SelectCommand = cmd
        Me.DocumentTableAdapter.Adapter.Fill(Me.ProtocolDataSet.Document)
        Me.DocumentBindingSource.Position = RowIndex

    Try
        idSenderComboBox.SelectedValue = idSenderTextBox.Text
        idReceiverComboBox.SelectedValue = idReceiverTextBox.Text
    Catch ex As Exception

    End Try

```

```
Catch ex As Exception

End Try

data_are_Loaded = True
moved = False
ToolStripButtonAddNew.Enabled = True
End Sub
```

Παρόμοια με την μέθοδο που καλείται όταν είναι να γίνει παραλαβή καινούργιου εγγράφου, είναι και η μέθοδος που καλείται όταν ο χρήστης θέλει να τροποποιήσει κάποια από τα στοιχεία του εγγράφου. Όπως φαίνεται στον κώδικα παραπάνω όταν είναι να γίνει επεξεργασία του εγγράφου απενεργοποιούνται κάποια στοιχεία, τα οποία δεν αφήνει η εφαρμογή να αλλάξουν.

Κάθε φορά που παραλαμβάνεται κάποιο έγγραφο για να περαστεί στη βάση δεδομένων πρέπει να του δοθεί ένας μοναδικός αριθμός πρωτοκόλλου. Ο αριθμός πρωτοκόλλου είναι ένας αύξων αριθμός που σχετίζεται με τον τύπο του εγγράφου (εισερχόμενο / εξερχόμενο). Περισσότερα για τον αριθμό πρωτοκόλλου υπάρχουν στο υποκεφάλαιο 2.1.1. Στον κώδικα που ακολουθεί με ένα select παίρνουμε τον αριθμό πρωτοκόλλου του εγγράφου που έχει το μεγαλύτερο idProtocol (βλ. υποκεφάλαιο 2.1.1) από τα έγγραφα που είναι εισερχόμενα (δλδ έχουν DocumentType=1). Το έγγραφο αυτό είναι και το τελευταίο που περάστηκε. Στη συνέχεια χωρίζουμε τον αριθμό πρωτοκόλλου σε δύο μέρη για να διαχωρίσουμε τον αύξοντα αριθμό του από την χρονολογία εισαγωγής του στη βάση. Έπειτα στον αύξοντα αριθμό προσθέτουμε το ένα (1) για να πάρουμε τον επόμενο αριθμό και βάζουμε δίπλα του την χρονολογία. Με αυτό τον τρόπο δίνουμε στο έγγραφο τον επόμενο διαθέσιμο αριθμό πρωτοκόλλου.

```
Private Sub setProtocolNumber()
    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String

    query = "SELECT NumberProtocol FROM Document WHERE DocumentType =1
and IdProtocol=(SELECT MAX(idProtocol) FROM Document WHERE
DocumentType=1) "

    Try
        cmd = New SqlCommand(query, connection)
        rdr = cmd.ExecuteReader
        rdr.Read()
        Dim protocolNumber As String
```

```

If IsDBNull(rdr.Item("NumberProtocol")) Then
    NumberProtocolTextBox.Text = "1-" & Now.Year
Else
    protocolNumber = rdr.Item("NumberProtocol")
    protocolNumber = protocolNumber.Split("-")(0)
    NumberProtocolTextBox.Text = CInt(Val(protocolNumber)) + 1
    NumberProtocolTextBox.Text &= "-" & Now.Year
End If

rdr.Close()
rdr = Nothing
Catch ex As Exception
    If Not (rdr Is Nothing) Then
        rdr.Close()
        rdr = Nothing
    End If
    MsgBox(ex.Message)
End Try
End Sub

```

Αφού δοθεί στο έγγραφο ο αριθμός πρωτοκόλλου ο χρήστης θα πρέπει να συμπληρώσει τα στοιχεία του εγγράφου, τουλάχιστον αυτά που είναι υποχρεωτικά. Υποχρεωτικά στοιχεία μεταξύ άλλων είναι ο παραλήπτης και ο αποστολέας. Για ευκολία του χρήστη και για να είμαστε σίγουροι ότι το όνομα του παραλήπτη και του αποστολέα είναι σωστά, τα στοιχεία αυτά δεν γράφονται με το χέρι αλλά ανοίγει μια βοηθητική φόρμα από την οποία ο χρήστης διαλέγει τον αποστολέα και τον παραλήπτη του εγγράφου. Με την επιλογή οποιουδήποτε από αυτούς, γίνεται ένας μικρός έλεγχος ώστε να αποφευχθεί το γεγονός να είναι ίδιος ο αποστολέας και ο παραλήπτης.

```

Private Sub BtnSelectSender_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnSelectSender.Click

    Dim FormResult As System.Windows.Forms.DialogResult
    FormResult = SelectContact.ShowDialog()
    If FormResult = Windows.Forms.DialogResult.OK Then
        idSenderComboBox.SelectedValue = idSender
        idSenderTextBox.Text = idSender
    End If
    If idSender = idReceiverComboBox.SelectedValue And
idReceiverComboBox.SelectedValue <> 0 Then

        MessageBox.Show("Δεν μπορεί ο αποστολέας και ο παραλήπτης του
εγγράφου να είναι ο ίδιος. Παρακαλώ επιλέξτε διαφορετικές επαφές.",
"Ορισμός Παραλήπτη Αποστολέα", MessageBoxButtons.OK,
MessageBoxIcon.Warning)

        idSenderComboBox.SelectedValue = 0

```

```
End If  
End Sub
```

Ο παραπάνω κώδικας εκτελείται κατά την επιλογή του αποστολέα του εγγράφου. Αντίστοιχος κώδικας υπάρχει και για την επιλογή του παραλήπτη.

Ένα σημαντικό μέρος της φόρμας αυτής είναι όταν ο χρήστης πρέπει να επιλέξει το έγγραφο ή το αντίγραφο του ώστε να μεταφερθεί στον φάκελο των εισερχομένων. Αφού πατήσει το πλήκτρο και βρει το έγγραφο στον υπολογιστή του, το έγγραφο αυτό θα πρέπει να πάρει το νέο του όνομα (το όνομα είναι ο αριθμός πρωτοκόλλου που του δόθηκε) και να δημιουργηθεί ένα αντίγραφο το οποίο θα αποθηκευτεί στον φάκελο των εισερχομένων. Η διεύθυνση και ο φάκελος που θα αποθηκεύονται τα διάφορα έγγραφα ορίζονται από τον χρήστη. Η διαδικασία φαίνεται στον κώδικα που ακολουθεί.

```
Private Sub RenameAndCopyFile()  
    If SavePath = Nothing OrElse IncomingFolderName = Nothing OrElse  
    OutgoingFolderName = Nothing OrElse GeneralFolderName = Nothing Then  
  
        MessageBox.Show("Δεν έχετε καθορίσει το χώρο στον οποίο θα  
αποθηκεύονται τα αρχεία σας. Θα πρέπει να τον καθορίσετε για να  
μπορέσετε να συνεχίζετε.", "Χώρος Αποθήκευσης", MessageBoxButtons.OK)  
        ConfigurationManagment.ShowDialog()  
        RenameAndCopyFile()  
    End If  
End Sub
```

Στην αρχή της μεθόδου ελέγχεται αν έχουν οριστεί από τον χρήστη η διεύθυνση αποθήκευσης των εγγράφων καθώς και τα ονόματα των φακέλων αποθήκευσης τους. Αν κάποιο από αυτά δεν έχει οριστεί τότε ανοίγει η φόρμα στην οποία ο χρήστης μπορεί να τα ορίσει και στη συνέχεια ξανακαλείται η ίδια μέθοδος ώστε να είμαστε σίγουροι ότι ο χρήστης έδωσε όλα τα στοιχεία που χρειάζονται για την αποθήκευση των εγγράφων.

Αφού βεβαιωθούμε ότι ορίστηκαν τα στοιχεία αποθήκευσης ελέγχουμε αν όντως υπάρχουν οι φάκελοι της αποθήκευσης και αν δεν υπάρχουν τους δημιουργούμε ξανά. Η διαδικασία αυτή γίνεται στη μέθοδο CheckFoldersAndCreate η οποία είναι δημόσια μέθοδος και υπάρχει στο Module1.

Στη συνέχεια παίρνουμε τη διεύθυνση του εγγράφου που επέλεξε ο χρήστης και την χωρίζουμε ώστε να πάρουμε την κατάληξη του εγγράφου. Την κατάληξη τη θέλουμε για να δημιουργήσουμε σωστά το καινούργιο όνομα του εγγράφου που



θα μεταφέρουμε στο φάκελο που ορίστηκε για τα εισερχόμενα αρχεία. Με τα στοιχεία της αποθήκευσης που όρισε ο χρήστης δημιουργείται η πλήρης διεύθυνση που θα αντιγραφεί το αρχείο και στη συνέχεια γίνεται η αντιγραφή του αρχείου.

```
Else
    CheckFoldersAndCreate()

    Dim FullNameOfFile As String =
txtOriginalFile.Text.Split("\")(txtOriginalFile.Text.Split("\").Length -
1)
    Dim FileExtension As String = FullNameOfFile.Split(".")(1)
    NewFileName = NumberProtocolTextBox.Text & "." & FileExtension
    Dim yearFolder As String = Now.Date.Year
    Dim monthFolder As String = Now.Month.ToString
    FullDirectoryName = SavePath & "\" & GeneralFolderName & "\" &
IncomingFolderName & "\" & yearFolder & "\" & monthFolder

    Try
        If Not (Directory.Exists(FullDirectoryName)) Then
            Directory.CreateDirectory(FullDirectoryName)
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

    PathTextBox.Text = FullDirectoryName
    FilenameTextBox.Text = NewFileName

    Try
        File.Copy(txtOriginalFile.Text, FullDirectoryName & "\" &
NewFileName)
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

End If

End Sub
```

Τελειώνοντας με τα στοιχεία που χρειάζεται το έγγραφο πρέπει να πατηθεί το πλήκτρο Αποθήκευση Εγγράφου, ώστε να περαστούν στη βάση δεδομένων τα στοιχεία αυτά. Με το πάτημα του πλήκτρου αυτού, γίνεται ένας μικρός έλεγχος αν ο χρήστης καταχώρησε όλα τα απαιτούμενα στοιχεία του εγγράφου. Αν τα έχει καταχωρήσει τότε πραγματοποιείται η ενημέρωση της βάσης δεδομένων.

```
Private Sub saveData()
    Try
```

```
Me.Validate()  
Me.DocumentBindingSource.EndEdit()  
Me.DocumentTableAdapter.Update(Me.ProtocolDataSet.Document)  
  
data_are_Saved = True  
moved = False  
  
MessageBox.Show("Τα δεδομένα αποθηκεύτηκαν επιτυχώς.", "Αποθήκευση  
Δεδομένων", MessageBoxButtons.OK)  
Catch ex As Exception  
    MessageBox.Show("Δημιουργήθηκε κάποιο πρόβλημα κατά την αποθήκευση  
των δεδομένων", "Αποθήκευση Δεδομένων", MessageBoxButtons.OK)  
End Try  
  
btnBrowseFile.Enabled = False  
DateDateTimePicker.Enabled = False  
EnableNavigationItems()  
End Sub
```

Με την αποθήκευση του εγγράφου στον πίνακα Document ενεργοποιούνται ξανά τα διάφορα πλήκτρα που απενεργοποιήθηκαν κατά την παραλαβή του εγγράφου και ρυθμίζεται η φόρμα ώστε να δεχτεί τυχών επιπλέον έγγραφο. Παράλληλα απενεργοποιείται το πλήκτρο για επιλογή εγγράφου και η επιλογή της ημερομηνίας αφού αυτά είναι δύο στοιχεία που δεν επιτρέπεται να τροποποιηθούν μετά την αποθήκευση του εγγράφου στη βάση δεδομένων.

### 3.3 Διαχείριση Εξερχόμενων Εγγράφων

Για την διαχείριση των εξερχόμενων εγγράφων δημιουργήθηκε ξεχωριστή φόρμα. Η φόρμα αυτή έχει την ίδια όψη με τη φόρμα διαχείρισης των εισερχομένων εγγράφων. Διαθέτει δηλαδή, ένα Data Grid View για την εμφάνιση των εγγράφων, μια toolbar για την μετακίνηση και την διαχείριση των εγγράφων και τέλος ένα πλαίσιο για να μπορεί ο χρήστης να κάνει σύνθετη αναζήτηση στα έγγραφα.

Επειδή ο κώδικας της φόρμας αυτής δεν διαφέρει κατά πολύ από τον κώδικα που υπάρχει στη φόρμα της διαχείρισης των εισερχομένων (υποκεφάλαιο 3.1), θα αναφέρουμε μόνο της διαφοροποιήσεις που υπάρχουν στους κώδικες των δύο φορμών.

Η πρώτη διαφορά που υπάρχει είναι το Select ερώτημα με το οποίο φορτώνονται τα εξερχόμενα έγγραφα. Εξερχόμενα είναι τα έγγραφα τα οποία έχουν DocumentType=0.

```
Dim SelectQuery As String = "Select * From Document WHERE DocumentType=0"
```

Για την δημιουργία ή την επεξεργασία κάποιου εξερχόμενου εγγράφου ανοίγει διαφορετική φόρμα από αυτή που ανοίγει για την παραλαβή εισερχόμενου εγγράφου. Ενδεικτικά παρατίθεται ο κώδικας για την επεξεργασία εξερχόμενου εγγράφου.

```
Private Sub ToolStripButtonEdit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripButtonEdit.Click

    If DocumentDataGridView.SelectedRows.Count > 0 Then
        Dim RowIndex As Integer
        RowIndex = DocumentDataGridView.SelectedRows(0).Index
        OutgoingFile.MdiParent = MainForm
        OutgoingFile.Show()
        OutgoingFile.LoadEditOutgoingFile(SelectQuery, RowIndex)
    Else
        OutgoingFile.MdiParent = MainForm
        OutgoingFile.Show()
        OutgoingFile.LoadEditOutgoingFile(SelectQuery, 0)
    End If
End Sub
```

Μια άλλη διαφορά στον κώδικα των δύο φορμών υπάρχει στη μέθοδο της αναζήτησης των εγγράφων. Εκτός από το DocumentType που στην συγκεκριμένη περίπτωση είναι μηδέν, υπάρχουν αλλαγές στον τρόπο που γίνεται η αναζήτηση με βάση τον οργανισμό και το τμήμα που σχετίζεται με το έγγραφο.

```
...
If LBoxOrganization.SelectedItems.Count > 0 Then
    query = query & " Contacts.IdContact=Document.IdSender AND"
    query = query & " Contacts.IdOrganization IN ("

For Each obj In LBoxOrganization.SelectedItems
    query = query & obj.item(0).ToString & ","
Next

If query.EndsWith(",") Then
    query = query.Remove(query.Length - 1)
End If
```

```
query = query & ") "  
End If  
...
```

Όταν ο χρήστης θέλει να κάνει αναζήτηση με βάση τον οργανισμό τότε ελέγχεται ο οργανισμός του παραλήπτη του εγγράφου. Ο λόγος είναι γιατί ένα εξερχόμενο έγγραφο γνωρίζουμε ότι έφυγε σίγουρα από την ίδια την εταιρεία και λογικά αυτό που θέλουμε να μάθουμε είναι σε ποιά εταιρεία στάλθηκε. Φυσικά με αυτό τον τρόπο δεν αποκλείεται το γεγονός να σταλεί ένα έγγραφο εσωτερικά στην εταιρεία.

Αντίστοιχη αλλαγή έγινε και στην αναζήτηση με βάση το τμήμα. Στη φόρμα αυτή το τμήμα σχετίζεται με τον αποστολέα του εγγράφου. Δεδομένου ότι ξέρουμε ότι το έγγραφο έφυγε από την «δική» μας εταιρεία, αν θέλουμε να μάθουμε περισσότερες πληροφορίες, αυτές θα ήταν το τμήμα το οποίο απέστειλε το έγγραφο.

```
...  
If LBoxDepartment.SelectedItems.Count > 0 Then  
    query = query & " INTERSECT "  
    query = query & " SELECT Document.* FROM Document,Contacts WHERE  
DocumentType=1 AND "  
  
    query = query & " Contacts.IdContact=Document.IdReceiver AND"  
    query = query & " Contacts.IdDepartment IN ("  
  
    For Each obj In LBoxDepartment.SelectedItems  
        query = query & obj.item(0).ToString &  
    Next  
  
    If query.EndsWith(",") Then  
        query = query.Remove(query.Length - 1)  
    End If  
  
    query = query & ") "  
End If  
...
```

Οι διαφορές που αναφέρθηκαν παραπάνω ήταν και οι σημαντικότερες (ίσως και όλες) που υπάρχουν μεταξύ της φόρμας για διαχείριση των εισερχομένων και αυτής για διαχείριση των εξερχομένων.

### 3.4 Δημιουργία εξερχόμενου Εγγράφου

Η δημιουργία καθώς και η επεξεργασία κάποιου εξερχόμενου εγγράφου γίνεται αντίστοιχα σε μια φόρμα η οποία και αυτή με τη σειρά της έχει πολλές ομοιότητες με την αντίστοιχη φόρμα για τα εισερχόμενα έγγραφα. Εκτός όμως από τις ομοιότητες έχουν και κάποιες διαφορές. Αυτές τις διαφορές θα αναλύσουμε στο υποκεφάλαιο αυτό.

Στην δημιουργία καινούργιου εξερχόμενου εγγράφου δίνεται ένας αριθμός πρωτοκόλλου. Η μέθοδος για τον ορισμό του αριθμού πρωτοκόλλου που θα δοθεί διαφέρει από αυτή που βρίσκεται στη φόρμα παραλαβής εισερχόμενου εγγράφου μόνο στο Select ερώτημα το οποίο παίρνει τον τελευταίο αριθμό πρωτοκόλλου που δόθηκε σε εξερχόμενο έγγραφο.

```
...  
query = "SELECT NumberProtocol FROM Document WHERE DocumentType =0 and  
IdProtocol=(SELECT MAX(idProtocol) FROM Document WHERE DocumentType=0)"  
...
```

Η σημαντική διαφορά που υπάρχει στην παραλαβή εισερχόμενου και στην δημιουργία εξερχόμενου εγγράφου είναι το ίδιο το έγγραφο. Στα εισερχόμενα το έγγραφο υπάρχει και ο χρήστης πρέπει να το βρει μέσα από τον υπολογιστή του ώστε να καταχωρηθεί στον φάκελο των εισερχομένων. Τα εξερχόμενα όμως έγγραφα δεν υπάρχουν ήδη αλλά πρέπει να δημιουργηθούν από την εφαρμογή. Η δημιουργία τους βασίζεται σε διάφορα πρότυπα με τα οποία ο ίδιος ο χρήστης έχει ενημερώσει την εφαρμογή. Η λογική που ακολουθείται είναι ο χρήστης να επιλέξει ένα από τα διαθέσιμα πρότυπα που υπάρχουν και να το τροποποιήσει σύμφωνα με τις ανάγκες του. Αφού το τροποποιήσει, το πρότυπο θα γίνει πλέον εξερχόμενο έγγραφο (αρχείο) και θα αποθηκευτεί στην κατάλληλη διεύθυνση αποθήκευσης των εξερχόμενων εγγράφων.

Για την επιλογή του προτύπου ανοίγει μια νέα φόρμα στην οποία εμφανίζονται τα πρότυπα που υπάρχουν.

```
Private Sub BtnChooseTemplate_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles BtnChooseTemplate.Click  
  
If SelectTemplate.ShowDialog() = Windows.Forms.DialogResult.OK Then
```

```

TxtTemplateName.Text = SelectedTemplateName
RenameAndCopyFile ()
Try
    BtnChooseTemplate.Enabled = False
    FilePathForViewer = FullDirectoryName & "\" & NewFileName
    System.Diagnostics.Process.Start (FilePathForViewer)
Catch ex As Exception
    MessageBox.Show ("Δημιουργήθηκε κάποιο πρόβλημα κατά την
εμφάνιση του εγγράφου", "Εμφάνιση Εγγράφου", MessageBoxButtons.OK)
End Try
End If
End Sub

```

Ο χρήστης επιλέγει το πρότυπο που θέλει και στη συνέχεια καλείται η μέθοδος `RenameAndCopyFile` ώστε να δημιουργήσει ένα αντίγραφο του προτύπου, να του δώσει όνομα τον αριθμό πρωτοκόλλου του εγγράφου και να το αποθηκεύσει στον φάκελο των εξερχομένων. Η μέθοδος `RenameAndCopyFile` έχει τον ίδιο κώδικα με αυτόν που έχει στη φόρμα παραλαβής των εισερχομένων. Η μόνη διαφορά είναι ο φάκελος στον οποίο αποθηκεύεται το έγγραφο.

Αφού δημιουργηθεί και αποθηκευτεί, το έγγραφο ανοίγει ώστε ο χρήστης να κάνει τις τροποποιήσεις που θέλει. Η εφαρμογή δεν μπορεί να ελέγξει αν ο χρήστης τροποποίησε και αποθήκευσε τις αλλαγές πριν την καταχώρηση του εγγράφου στη βάση. Αυτό όμως δεν δημιουργεί κανένα πρόβλημα στα δεδομένα του εγγράφου. Το μόνο που δεν μπορεί να κάνει ο χρήστης χωρίς να κλείσει το έγγραφο είναι να ακυρώσει τη διαδικασία δημιουργίας του εγγράφου.

Αν για κάποιο λόγο ο χρήστης έκανε κάποιο λάθος και δεν θέλει τελικά να δημιουργήσει το εξερχόμενο έγγραφο τότε μπορεί να πατήσει απλά το πλήκτρο Ακύρωση. Πατώντας το πλήκτρο αυτό, δεν αποθηκεύονται τα πεδία του εγγράφου στη βάση δεδομένων και το έγγραφο (αρχείο) που δημιουργήθηκε διαγράφεται από τον φάκελο των εξερχομένων.

```

Private Sub BtnCancel_Click (ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnCancel.Click
    If Not IdProtocolTextBox.Text.Contains ("-") Then
        Exit Sub
    End If

    Dim dlgRslt As DialogResult
    dlgRslt = MessageBox.Show ("Το έγγραφο δεν έχει ακόμα αποθηκευτεί. Αν
ακυρώσετε τη διαδικασία τα δεδομένα θα χαθούν. Είστε σίγουροι ότι θέλετε
να ακυρώσετε τη διαδικασία παραλαβής εγγράφου;", "Ακύρωση Παραλαβής
Εγγράφου", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question)

```

```
If dlgRslt = Windows.Forms.DialogResult.Yes Then
    Try
        If TxtTemplateName.Text <> "" Then
            File.Delete(FullDirectoryName & "\" & NewFileName)
        End If
        CancelOutgoing = True
        Me.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End If
End Sub
```

Με το πάτημα του πλήκτρου Ακύρωση εκτός από την διαγραφή του αρχείου που δημιουργήθηκε κλείνει και η φόρμα της δημιουργίας του εγγράφου.

### 3.5 Διαχείριση Οργανισμών

Η φιλοσοφία που ακολουθείται για τη φόρμα της διαχείρισης των οργανισμών είναι η ίδια με αυτήν που ακολουθείται σε όλες τις φόρμες διαχείρισης των πινάκων της βάσης δεδομένων που υπάρχουν. Η φόρμα αυτή διαθέτει ένα Data Grid View για την εμφάνιση των εγγραφών του πίνακα Organization, ένα πλαίσιο για αναζήτηση και μια toolbar για επεξεργασία του πίνακα.

Παρόμοια φόρμα με αυτή είναι και η φόρμα διαχείρισης των εισερχομένων (υποκεφάλαιο 3.1) η οποία αναλύθηκε αρκετά στην αρχή του κεφαλαίου. Επειδή οι διαφοροποιήσεις δεν είναι πολύ σημαντικές ανάμεσα στις δύο φόρμες θα αναφέρουμε μόνο αυτές που θεωρούμε περισσότερο σημαντικές.

Επειδή όταν ανοίξει η φόρμα θέλουμε ο χρήστης να βλέπει όλους τους οργανισμούς που υπάρχουν δεν χρησιμοποιούμε κάποιο Select όπως στα εισερχόμενα και στα εξερχόμενα για να πάρουμε μόνο αυτά που θέλουμε.

Σε αντίθεση με τα έγγραφα (εισερχόμενα και εξερχόμενα) που δεν μπορούσε ο χρήστης, αφού τα έχει καταχωρήσει να τα διαγράψει, η διαγραφή μπορεί να γίνει στους οργανισμούς. Η διαγραφή όμως γίνεται μόνο όταν στον συγκεκριμένο οργανισμό δεν υπάρχουν επαφές που να σχετίζονται σε αυτό. Ο έλεγχος αυτός γίνεται κάθε φορά που ο χρήστης θέλει να διαγράψει ένα ή περισσότερους οργανισμούς.

```
Private Function OrganizationCanDelete (ByVal OrganizationId As Integer)
As Boolean

    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String
    query = "SELECT idContact FROM Contacts WHERE idOrganization=" &
OrganizationId

    cmd = New SqlCommand(query, connection)
    Try
        rdr = cmd.ExecuteReader
        If rdr.HasRows() Then
            Return False
        Else
            Return True
        End If
    Catch ex As Exception

    End Try
End Function
```

Αφού γίνει ο έλεγχος αυτός και βεβαιωθούμε ότι μπορούμε να διαγράψουμε τον οργανισμό, τότε με την βοήθεια μιας άλλης μεθόδου πραγματοποιούμε την διαγραφή του.

```
Private Sub DeleteOrganization (ByVal OrganizationId As Integer)

    Dim cmd As SqlCommand
    Dim query As String
    query = "DELETE FROM Organization WHERE IdOrganization=" &
OrganizationId

    Try
        cmd = New SqlCommand(query, connection)
        cmd.ExecuteNonQuery()
    Catch ex As Exception
        MessageBox.Show("Δημιουργήθηκε πρόβλημα κατά τη διαγραφή του
οργανισμού.")
    End Try
End Sub
```

Ο χρήστης έχει την δυνατότητα να επιλέξει περισσότερους από έναν οργανισμούς για διαγραφή. Αν κάποιος από τους οργανισμούς που επέλεξε δεν μπορεί να διαγραφεί τότε εμφανίζεται κατάλληλο μήνυμα που τον ενημερώνει.

Όπως και στις άλλες φόρμες διαχείρισης, έτσι και σε αυτήν ο χρήστης δεν μπορεί να αλλάξει τα στοιχεία του οργανισμού μέσα από την συγκεκριμένη φόρμα. Για να



κάνει κάποιες αλλαγές στα στοιχεία ανοίγει μια διαφορετική φόρμα. Η ίδια φόρμα ανοίγει και όταν ο χρήστης θέλει να δημιουργήσει καινούργιο οργανισμό.

Τέλος, θα παρουσιάσουμε τον τρόπο με τον οποίο γίνεται η αναζήτηση στη φόρμα αυτή. Η λογική δεν διαφέρει από τις αναζητήσεις που έγιναν στις άλλες φόρμες. Η διαφορά είναι στο Select ερώτημα που δημιουργείται, το οποίο είναι προσαρμοσμένο στον πίνακα Organization.

```
Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearch.Click

    Dim cmd As SqlCommand
    Dim query As String
    query = "Select * From Organization Where "
    If txtBoxOrgName.Text <> "" Then
        Dim input As String = txtBoxOrgName.Text
        input = input.Replace("*", "%")
        query = query & "OrgName Like '" & input & "' AND "
    End If
    ...
    ...
    ...
    If query.EndsWith("Where ") Then
        query = query.Replace("Where ", "")
    End If

    If query.EndsWith("AND ") Then
        query = query.Remove(query.Length - 4)
    End If

    SelectQuery = query
    cmd = New SqlCommand(query, connection)

    Me.OrganizationTableAdapter.Adapter.SelectCommand = cmd
    Me.ProtocolDataSet.Organization.Clear()
    Me.OrganizationTableAdapter.Adapter.Fill(Me.ProtocolDataSet,
"Organization")

    Me.OrganizationBindingSource.DataSource =
Me.ProtocolDataSet.Organization
    Me.OrganizationDataGridView.DataSource =
Me.OrganizationBindingSource

    Me.OrganizationBindingNavigator.BindingSource =
Me.OrganizationBindingSource
    Me.OrganizationBindingNavigator.Update()

End Sub
```

### 3.6 Επεξεργασία Οργανισμού

Στην φόρμα της επεξεργασίας του οργανισμού ο χρήστης εκτός από το να αλλάξει τα στοιχεία του οργανισμού μπορεί και να προσθέσει ένα νέο οργανισμό. Η φόρμα αυτή έχει την μορφή Details καθώς και μια toolbar η οποία βοηθά τον χρήστη στην επεξεργασία του κάθε οργανισμού και στην πλοήγηση του ανάμεσα τους.

Τα εργαλεία που υπάρχουν στην toolbar είναι τα ίδια με αυτά που υπάρχουν στην φόρμα για την διαχείριση των οργανισμών, για αυτό δεν θα αναφερθεί αναλυτικά ο κώδικας που υπάρχει σε αυτά.

Ένα από τα σημαντικά κομμάτια κώδικα σε αυτήν την εφαρμογή είναι ο έλεγχος που γίνεται αν ο οργανισμός είναι ήδη καταχωρημένος.

```
Private Function OrganizationAlreadyExists() As Boolean
    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String
    query = "SELECT * FROM Organization WHERE OrgName='" &
    OrgNameTextBox.Text & "'"
    cmd = New SqlCommand(query, connection)
    Try
        rdr = cmd.ExecuteReader
        If rdr.HasRows() Then
            Return True
        Else
            Return False
        End If
    Catch ex As Exception

    End Try
End Function
```

Όταν ο χρήστης προσπαθήσει να περάσει έναν καινούργιο οργανισμό, τότε εκτός από τον έλεγχο των υποχρεωτικών πεδίων (σε αυτή την φόρμα είναι μόνο το όνομα του οργανισμού), γίνεται και έλεγχος για το αν υπάρχει ήδη στον πίνακα των οργανισμών με το ίδιο όνομα.

### 3.7 Διαχείριση Επαφών

Η διαχείριση του πίνακα Contacts γίνεται από την φόρμα Διαχείριση Επαφών. Οι διάφορες επαφές που υπάρχουν στον πίνακα εμφανίζονται μέσα σε ένα Data Grid View και η διαχείριση τους γίνεται με τη βοήθεια των εργαλείων που υπάρχουν στην toolbar. Όπως και στις άλλες φόρμες έτσι και σε αυτή, το Data Grid View είναι συνδεδεμένο με τον πίνακα Contacts με τη βοήθεια ενός adapter.

Τα εργαλεία που υπάρχουν στην toolbar είναι τα ίδια που υπάρχουν και στις toolbars των άλλων φορμών. Οι διαφορές που υπάρχουν είναι στην φόρμα που ανοίγεται όταν ο χρήστης θέλει να προσθέσει κάποια επαφή ή να επεξεργαστεί ήδη υπάρχουσα καθώς και στην διαδικασία της διαγραφής.

```
Private Function ContactCanDelete (ByVal ContactId As Integer) As Boolean
    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String
    query = "SELECT idProtocol FROM Document WHERE idSender=" & ContactId
    & " OR idReceiver=" & ContactId
    cmd = New SqlCommand(query, connection)
    Try
        rdr = cmd.ExecuteReader
        If rdr.HasRows() Then
            Return False
        Else
            Return True
        End If
    Catch ex As Exception

    End Try
End Function
```

Ο χρήστης μπορεί να διαγράψει μια ή περισσότερες επαφές, αρκεί αυτές να μην σχετίζονται με κανένα έγγραφο. Ο έλεγχος αυτός γίνεται με την βοήθεια του IdContact και της μεθόδου που εμφανίζεται παραπάνω. Αφού η επαφή δεν σχετίζεται με κανένα έγγραφο, τότε μπορεί να διαγραφεί.

```
Private Sub DeleteContact (ByVal idContact As Integer)
    Dim cmd As SqlCommand
    Dim query As String
    query = "DELETE FROM Contacts WHERE IdContact=" & idContact

    Try
        cmd = New SqlCommand(query, connection)
        cmd.ExecuteNonQuery()
    End Try
End Sub
```

```

Catch ex As Exception
    MessageBox.Show("Δημιουργήθηκε πρόβλημα κατά τη διαγραφή της
επαφής")
End Try
End Sub

```

Ένα άλλο κομμάτι που θα θέλαμε να αναφέρουμε είναι το κομμάτι της αναζήτησης. Η λογική δεν διαφέρει από τις άλλες αναζητήσεις αλλά θα δείξουμε ένα μικρό κομμάτι του κώδικα που είναι λίγο πιο περίπλοκο.

```

Dim cmd As SqlCommand
Dim query As String
query = "Select * From Contacts Where "
...
...
...
If LBoxOrganization.SelectedItems.Count > 0 Then
    query = query & "IdOrganization IN ("

    For Each obj In LBoxOrganization.SelectedItems
        query = query & obj.item(0).ToString & ","
    Next

    If query.EndsWith(",") Then
        query = query.Remove(query.Length - 1)
    End If

    query = query & ") AND "
End If

If LBoxDepartment.SelectedItems.Count > 0 Then
    query = query & "IdDepartment IN ("

    For Each obj In LBoxDepartment.SelectedItems
        query = query & obj.item(0).ToString & ","
    Next

    If query.EndsWith(",") Then
        query = query.Remove(query.Length - 1)
    End If

    query = query & ")"
End If
...
...
...

```

Στην αναζήτηση των επαφών δίνεται η δυνατότητα στον χρήστη να ζητήσει να εμφανιστούν οι επαφές που ανήκουν σε κάποιον/κάποιους συγκεκριμένους

οργανισμούς ή τμήματα. Οι οργανισμοί και τα τμήματα εμφανίζονται μέσα σε δύο List Box από τα οποία ο χρήστης μπορεί να επιλέξει αυτά που θέλει.

Η δημιουργία του select ερωτήματος στην περίπτωση της αναζήτησης με βάση τους οργανισμούς και τα τμήματα εμφανίζεται στον παραπάνω κώδικα. Στον κώδικα δημιουργείται δυναμικά ένα κομμάτι του Where το οποίο σχετίζεται με τον οργανισμό ή το τμήμα (ανάλογα) στα οποία ανήκει μια επαφή.

### 3.8 Επεξεργασία Επαφής

Η φόρμα για την επεξεργασία και την δημιουργία μιας επαφής είναι η φόρμα η οποία ανοίγει όταν πατηθεί επεξεργασία ή προσθήκη στη φόρμα της διαχείρισης των επαφών. Η φόρμα αυτή εμφανίζει τα στοιχεία με την μορφή details και έχει την ίδια toolbar και τα ίδια εργαλεία με αυτά που υπάρχουν στην φόρμα της διαχείρισης επαφών.

Στη φόρμα αυτή το μόνο κομμάτι κώδικα που θα αναφέρουμε είναι ο έλεγχος για τα υποχρεωτικά πεδία κατά την αποθήκευση της εγγραφής στον πίνακα Contacts.

```
Private Function CheckRequiredFields() As Boolean
    If SurnameConTextBox.Text = "" Then
        Return False
    ElseIf NameConTextBox.Text = "" Then
        Return False
    End If
    Return True
End Function
```

Οι υπόλοιπες λειτουργίες που υπάρχουν στη φόρμα αυτή ακολουθούν τον ίδιο κώδικα όπως και στην φόρμα της διαχείρισης των επαφών και δεν κρίνεται αναγκαίο να παρουσιαστούν.

### 3.9 Διαχείριση Τμημάτων

Στην φόρμα αυτή ο χρήστης μπορεί να διαχειριστεί τα δεδομένα του πίνακα Department. Η φόρμα αυτή διαφέρει σε σχέση με τις αντίστοιχες φόρμες διαχείρισης που περιγράψαμε παραπάνω στο γεγονός ότι δεν υπάρχει αναζήτηση.

Κατά τα άλλα το Data Grid View και η Toolbar υπάρχουν. Τα εργαλεία είναι τα ίδια και διαχειρίζονται τα δεδομένα του Data Grid View όπως και στις άλλες φόρμες.

Ο κώδικας που αξίζει να αναφερθεί στην φόρμα αυτή είναι η διαγραφή ενός ή περισσότερων τμημάτων. Για να διαγραφεί ένα τμήμα πρέπει πρώτα να γίνει έλεγχος ότι καμία επαφή δεν υπάρχει σε αυτό.

```
Private Function DepartmentCanDelete (ByVal DepartmentId As Integer) As Boolean

    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String
    query = "SELECT idContact FROM Contacts WHERE idDepartment=" &
DepartmentId
    cmd = New SqlCommand(query, connection)
    Try
        rdr = cmd.ExecuteReader
        If rdr.HasRows() Then
            Return False
        Else
            Return True
        End If
    Catch ex As Exception

    End Try
End Function
```

Αν η μέθοδος που φαίνεται παραπάνω, επιστρέψει την τιμή true τότε το τμήμα μπορεί να διαγραφεί και τότε καλείται η μέθοδος για την διαγραφή του.

```
Private Sub DeleteDepartment (ByVal DepartmentId As Integer)
    Dim cmd As SqlCommand
    Dim query As String
    query = "DELETE FROM Department WHERE IdDepartment=" & DepartmentId
```

```
Try
    cmd = New SqlCommand(query, connection)
    cmd.ExecuteNonQuery()
Catch ex As Exception
    MessageBox.Show("Δημιουργήθηκε πρόβλημα κατά τη διαγραφή του
    τμήματος")
End Try
End Sub
```

### 3.10 Επεξεργασία Τμήματος

Όταν ο χρήστης επιλέξει από τη φόρμα της διαχείρισης των τμημάτων να δημιουργήσει ένα καινούργιο τμήμα ή να τροποποιήσει τα στοιχεία κάποιου ήδη υπάρχοντος τότε ανοίγει αυτή η φόρμα.

Ο χρήστης στην φόρμα αυτή βλέπει τα στοιχεία του τμήματος με τη μορφή Details. Στη φόρμα αυτή υπάρχει και η toolbar η οποία θα βοηθήσει τον χρήστη στην επεξεργασία των τμημάτων του.

Τα εργαλεία και ο κώδικας τους είναι ο ίδιος με αυτόν που περιγράψαμε στη φόρμα Διαχείρισης Τμημάτων και δεν θεωρείται αναγκαίο να αναφερθεί. Για να αναφέρουμε όμως κάτι θα πούμε ότι υποχρεωτικά πεδία για ένα τμήμα είναι μόνο το όνομα του και ο έλεγχος γίνεται κάθε φορά που ο χρήστης πατήσει το πλήκτρο της αποθήκευσης.

```
Private Sub BtnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnSave.Click

    If DepNameTextBox.Text = "" Then
        MessageBox.Show("Δεν έχετε καταχωρήσει το όνομα του τμήματος",
        "Αποθήκευση Τμήματος", MessageBoxButtons.OK)
    Else
        If newDepartment Then
            If DepartmentAlreadyExists() Then
                MessageBox.Show("Το τμήμα με όνομα '" &
                DepNameTextBox.Text & "' υπάρχει ήδη καταχωρημένο", "Αποθήκευση
                τμήματος", MessageBoxButtons.OK)
            Else
                saveData()
            End If
        Else
    
```

```
        saveData()  
    End If  
End If  
End Sub
```

Κατά την αποθήκευση ενός νέου τμήματος γίνεται έλεγχος αν το τμήμα αυτό υπάρχει ήδη καταχωρημένο. Αν υπάρχει, τότε η εφαρμογή δεν αφήνει τον χρήστη να το περάσει ξανά. Ο έλεγχος αυτός γίνεται με τη βοήθεια της μεθόδου `DepartmentAlreadyExists`.

```
Private Function DepartmentAlreadyExists() As Boolean  
    Dim cmd As SqlCommand  
    Dim rdr As SqlDataReader = Nothing  
    Dim query As String  
    query = "SELECT * FROM Department WHERE DepName='" &  
    DepNameTextBox.Text & "'" &  
    cmd = New SqlCommand(query, connection)  
    Try  
        rdr = cmd.ExecuteReader  
        If rdr.HasRows() Then  
            Return True  
        Else  
            Return False  
        End If  
    Catch ex As Exception  
    End Try  
End Function
```

### 3.11 Διαχείριση Τύπων Αρχείου

Για τη διαχείριση των εγγραφών που υπάρχουν στη πίνακα `Filetype` ο χρήστης θα πρέπει να ανοίξει τη φόρμα αυτή. Η φόρμα αυτή όπως και αυτή της διαχείρισης των τμημάτων δεν έχει αναζήτηση. Διαθέτει όμως το κλασσικό `Data Grid View` για την εμφάνιση των εγγραφών του πίνακα και την `toolbar` για την διαχείριση τους.

Το σημαντικό κομμάτι σε αυτή τη φόρμα είναι η διαγραφή που μπορεί να κάνει ο χρήστης στους τύπους αρχείου. Αφού ο χρήστης επιλέξει τον ή τους τύπους αρχείου που θέλει να διαγράψει, γίνεται ένας έλεγχος.



Ο έλεγχος αυτός δεν αφήνει τον χρήστη να διαγράψει κάποιον τύπο αρχείου ο οποίος σχετίζεται με κάποιο έγγραφο.

```
Private Function FileTypeCanDelete(ByVal FileTypeid As Integer) As Boolean
    Dim cmd As SqlCommand
    Dim rdr As SqlDataReader = Nothing
    Dim query As String
    query = "SELECT idProtocol FROM Document WHERE idFileType=" &
FileTypeid
    cmd = New SqlCommand(query, connection)
    Try
        rdr = cmd.ExecuteReader
        If rdr.HasRows() Then
            Return False
        Else
            Return True
        End If
    Catch ex As Exception

    End Try
End Function
```

Αν η μέθοδος που φαίνεται παραπάνω μας επιτρέπει να διαγράψουμε το τύπο εγγράφου, τότε εκτελείται η ακόλουθη μέθοδος η οποία πραγματοποιεί τη διαγραφή.

```
Private Sub DeleteFileType(ByVal idFileType As Integer)
    Dim cmd As SqlCommand
    Dim query As String
    query = "DELETE FROM Filetype WHERE IdFiletype=" & idFileType

    Try
        cmd = New SqlCommand(query, connection)
        cmd.ExecuteNonQuery()
    Catch ex As Exception
        MessageBox.Show("Δημιουργήθηκε πρόβλημα κατά τη διαγραφή της εγγραφής")
    End Try
End Sub
```

### 3.12 Επεξεργασία Τύπου Αρχείων

Η δημιουργία ενός τύπου αρχείου ή η τροποποίηση ενός αρχείου που είναι ήδη καταχωρημένο γίνεται από αυτή την φόρμα. Η φόρμα αυτή ανοίγει μέσα από τη φόρμα για την διαχείριση των τύπων αρχείων.

Επειδή ο κώδικας στα σημαντικά σημεία είναι όμοιος με αυτόν που υπάρχει στη φόρμα της διαχείρισης δεν θα αναφέρουμε πολλά πράγματα. Στην φόρμα αυτήν δεν ελέγχουμε αν κάποιος τύπος αρχείου υπάρχει ήδη. Ο μόνος έλεγχος που γίνεται είναι αν έχει καταχωρήσει ο χρήστης όλα τα υποχρεωτικά πεδία. Υποχρεωτικά πεδία ορίστηκαν όλα τα πεδία στη φόρμα αυτή.

### 3.13 Εισαγωγή Προτύπου

Στο υποκεφάλαιο για την δημιουργία εξερχόμενου εγγράφου, αναφέραμε ότι για να δημιουργηθεί ένα έγγραφο πρέπει πρώτα ο χρήστης της εφαρμογής να επιλέξει κάποιο πρότυπο από αυτά που έχει καταχωρήσει στην εφαρμογή.

Η εισαγωγή του προτύπου στην εφαρμογή γίνεται από τη φόρμα αυτή. Τα πρότυπα δεν αποθηκεύονται στη βάση δεδομένων αλλά σε ένα φάκελο ενός υπολογιστή τον οποίο βλέπουν όλοι οι χρήστες της εφαρμογής. Η επιλογή της διεύθυνσης στην οποία θα αποθηκεύονται τα πρότυπα γίνεται από τον ίδιο το χρήστη μέσα από τη φόρμα της Διαμόρφωσης της εφαρμογής.

Πώς λειτουργεί τώρα η φόρμα εισαγωγής του προτύπου. Στην φόρμα αυτή υπάρχει ένα open file dialog με το οποίο ο χρήστης επιλέγει το αρχείο που θα αποτελέσει το πρότυπο του. Στη συνέχεια θέτει ένα χαρακτηριστικό όνομα στο αρχείο, ανεξάρτητα αν το έχει κάνει ήδη πρέπει να περάσει και μέσα από την εφαρμογή έναν τίτλο στο πρότυπο. Επίσης αν το επιθυμεί μπορεί να καταχωρήσει και κάποια περιγραφή του προτύπου αυτού. Η περιγραφή που καταχωρεί μπαίνει στο όνομα του αρχείου, δίπλα από τον τίτλο, σε μια παρένθεση.

Αφού επιλέξει αρχείο και συμπληρώσει τουλάχιστον ένα τίτλο για τον πρότυπο τότε πρέπει να πατήσει το πλήκτρο καταχώρηση ώστε το πρότυπο να μεταφερθεί στη διεύθυνση που αποθηκεύονται τα πρότυπα. Ο κώδικας που αναπτύχθηκε στο πλήκτρο της καταχώρησης φαίνεται παρακάτω.

```
Private Sub InsertTemplate()  
  
    If isAllData() Then  
  
        RenameAndCopyTemplateFile()  
  
        templateSaved = True  
    Else  
        MessageBox.Show("Δεν έχετε εισάγει όλα τα απαιτούμενα πεδία για την  
καταχώρηση του προτύπου", "Καταχώρηση Προτύπου", MessageBoxButtons.OK)  
    End If  
End Sub
```

Στην αρχή γίνεται έλεγχος ότι έχουν περαστεί όλα τα απαιτούμενα πεδία για την μεταφορά του προτύπου και στη συνέχεια καλείται η μέθοδος για να πραγματοποιήσει την μεταφορά αυτή.

```
Private Sub RenameAndCopyTemplateFile()  
    Dim fileName As String  
    Dim NewFileName As String  
  
    If TemplateFullPath = Nothing Then  
        MessageBox.Show("Δεν έχετε καθορίσει το χώρο στον οποίο θα  
αποθηκεύονται τα πρότυπα σας. Θα πρέπει να τον καθορίσετε για να  
μπορέσετε να συνεχίζεται.", "Χώρος Αποθήκευσης Προτύπων",  
MessageBoxButtons.OK)  
        ConfigurationManagment.ShowDialog()  
        RenameAndCopyTemplateFile()  
    Else  
        CheckFoldersAndCreate()  
  
        fileName = TxtTitle.Text  
        If TxtDescription.Text <> "" Then  
            fileName &= "(" & TxtDescription.Text & ")"  
        End If  
  
        Dim FullNameOfFile As String =  
TxtFile.Text.Split("\")(TxtFile.Text.Split("\").Length - 1)  
        Dim FileExtension As String =  
FullNameOfFile.Split(".")(FullNameOfFile.Split(".").Length - 1)  
  
        NewFileName = fileName & "." & FileExtension  
  
    Try
```

```
File.Copy(TxtFile.Text, TemplateFullPath & "\" & NewFileName)
MessageBox.Show("Το πρότυπο αποθηκεύτηκε επιτυχώς.",
"Aποθήκευση Προτύπου", MessageBoxButtons.OK)
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try

End If

End Sub
```

Θα παρατηρήσετε ότι ο κώδικας αυτός θυμίζει κατά πολύ τον κώδικα που υπάρχει στη δημιουργία του εξερχόμενου εγγράφου και στην παραλαβή του εισερχόμενου. Στην αρχή γίνεται έλεγχος αν έχει οριστεί διεύθυνση για την αποθήκευση προτύπου και αφού βεβαιωθούμε ότι ο χρήστης έχει καταχωρήσει τέτοια διεύθυνση δημιουργούμε το νέο όνομα του αρχείου και το μεταφέρουμε στην διεύθυνση αυτή.

### 3.14 Διαμόρφωση Εφαρμογής

Στα προηγούμενα υποκεφάλαια αναφέραμε συχνά ότι τα εισερχόμενα, τα εξερχόμενα έγγραφα και τα πρότυπα αποθηκεύονται σε μια διεύθυνση ενός υπολογιστή (πιθανόν σε κάποιον χώρο που μπορούν να δουν όλοι οι χρήστες της εφαρμογής). Οι διευθύνσεις αυτές καθορίζονται από τον χρήστη με τη βοήθεια αυτής της φόρμας.

Αποφασίσαμε να δίνεται στον χρήστη αυτή η δυνατότητα ώστε να προσαρμόσει τις διευθύνσεις και τις ονομασίες των διάφορων φακέλων σύμφωνα με τα δικά του δεδομένα. Αρχικά θα πρέπει να επιλέξει τη διεύθυνση που θα δημιουργηθεί ο γενικός φάκελος του πρωτοκόλλου. Στη συνέχεια θα πρέπει να δώσει το όνομα που θέλει στον γενικό φάκελο του πρωτοκόλλου, στο φάκελο των εισερχομένων και στον φάκελο των εξερχομένων. Επιπλέον πρέπει να επιλέξει τη διεύθυνση που θα καταχωρούνται τα διάφορα πρότυπα του. Στα πρότυπα αρκεί να επιλέξει τη διεύθυνση που θέλει. Το όνομα του φακέλου το δίνει απευθείας η εφαρμογή.

Ο περισσότερος κώδικας της φόρμας αυτής βρίσκεται στο κουμπί της αποθήκευσης των αλλαγών. Οι αλλαγές που γίνονται στη φόρμα αυτή αποθηκεύονται στον πίνακα Configuration και πολλές φορές επηρεάζουν και τα

δεδομένων του πίνακα Document στο πεδίο της διεύθυνσης που βρίσκεται το αρχείο.

Όταν ο χρήστης πατήσει αποθήκευση το πρώτο πράγμα που εκτελείται είναι ο έλεγχος αν οι διάφοροι φακέλοι υπάρχουν. Αν δεν υπάρχουν τότε δημιουργούνται. Δηλαδή ο κώδικας αυτός εκτελείται μόνο την πρώτη φορά που ο χρήστης θα περάσει δεδομένα σε κάποιο πεδίο.

```
If (SavePathTextBox.Text <> "") AndAlso (GeneralFolderName = Nothing And
GeneralFolderNameTextBox.Text <> "") Then
    Try
        Directory.CreateDirectory(SavePathTextBox.Text & "\" &
GeneralFolderNameTextBox.Text)
    Catch ex As Exception
        MessageBox.Show("Δημιουργήθηκε σφάλμα κατά τη δημιουργία του
φακέλου", "Δημιουργία φακέλου", MessageBoxButtons.OK)
    End Try
End If

Try
    If IncomingFolderName = Nothing And IncomingFolderNameTextBox.Text <>
"" Then
        Directory.CreateDirectory(SavePathTextBox.Text & "\" &
GeneralFolderNameTextBox.Text & "\" & IncomingFolderNameTextBox.Text)
    End If
Catch ex As Exception
    MessageBox.Show("Δημιουργήθηκε σφάλμα κατά τη δημιουργία του
φακέλου", "Δημιουργία φακέλου", MessageBoxButtons.OK)
End Try

Try
    If OutgoingFolderName = Nothing And OutgoingFolderNameTextBox.Text <>
"" Then
        Directory.CreateDirectory(SavePathTextBox.Text & "\" &
GeneralFolderNameTextBox.Text & "\" & OutgoingFolderNameTextBox.Text)
    End If
Catch ex As Exception
    MessageBox.Show("Δημιουργήθηκε σφάλμα κατά τη δημιουργία του
φακέλου", "Δημιουργία φακέλου", MessageBoxButtons.OK)
End Try

Try
    If TemplateFullPath = Nothing And TemplateFolderNameTextBox.Text <>
"" Then
        Directory.CreateDirectory(TemplateFolderNameTextBox.Text)
    End If
Catch ex As Exception
```

```

    MessageBox.Show("Δημιουργήθηκε σφάλμα κατά τη δημιουργία του φακέλου", "Δημιουργία φακέλου", MessageBoxButtons.OK)
End Try

```

Όταν ο χρήστης για κάποιο λόγο αλλάξει την διεύθυνση αποθήκευσης των εγγράφων τότε εκτελείται ένας διαφορετικό κώδικας, ο οποίος ρωτάει τον χρήστη αν θέλει να μεταφέρει όλα τα έγγραφα στην νέα του διεύθυνση.

```

If SavePathChanged Then
    Dim dResult As DialogResult
    If (Not (SavePath = Nothing)) And SavePathTextBox.Text <> "" And (SavePath <> SavePathTextBox.Text) Then
        dResult = MessageBox.Show("Έχετε αλλάξει τη διεύθυνση αποθήκευσης των εγγράφων. Θα θέλατε να μεταφερθούν τα έγγραφα με τους φακέλους τους στη διεύθυνση αυτή;", " Αλλαγή Διεύθυνσης Αποθήκευσης Εγγράφων", MessageBoxButtons.YesNoCancel)

        If dResult = Windows.Forms.DialogResult.Yes Then
            If GeneralFolderNameChanged Then
                Directory.Move(SavePath & "\" & GeneralFolderNameTextBox.Text, SavePathTextBox.Text & "\" & GeneralFolderNameTextBox.Text)
            Else
                Directory.Move(SavePath & "\" & GeneralFolderName, SavePathTextBox.Text & "\" & GeneralFolderName)
            End If

            UpdateDocumentSavePath()
        End If
    End If
End If

```

Αν ο χρήστης θέλει να τα μεταφέρει τα έγγραφα στην νέα διεύθυνση τότε τα έγγραφα μεταφέρονται και στη συνέχεια ενημερώνεται το πεδίο της διεύθυνσης τους στη βάση δεδομένων.

```

Private Sub UpdateDocumentSavePath()
    Dim cmd As SqlCommand
    Dim cmdUpdate As SqlCommand
    Dim rdr As SqlDataReader
    Dim query As String
    Dim UpdateQuery As String
    Dim NewPath As String
    Dim IdProtocol As Integer
    query = "SELECT IdProtocol,Path FROM Document WHERE Path LIKE '" & SavePath & "%'"
    Try
        cmd = New SqlCommand(query, connection)
        rdr = cmd.ExecuteReader
    End Try

```

```

If rdr.HasRows Then
    While rdr.Read()
        IdProtocol = rdr.Item("IdProtocol")
        NewPath = rdr.Item("Path")
        NewPath = NewPath.Replace(SavePath, SavePathTextBox.Text)
        UpdateQuery = "UPDATE Document SET Path='" & NewPath & "'
WHERE IdProtocol=" & IdProtocol
        cmdUpdate = New SqlCommand(UpdateQuery, connection)
        cmdUpdate.ExecuteNonQuery()
    End While
    rdr.Close()
End If
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
End Sub

```

Αντίστοιχα αν ο χρήστης αλλάξει το όνομα κάποιου από τους φακέλους που αποθηκεύονται τα έγγραφα γίνεται παρόμοιος έλεγχος. Αν θέλει ο χρήστης όντως να αλλάξει το όνομα του φακέλου τότε καλείται μέθοδος που ενημερώνει το πεδίο της διεύθυνσης των εγγράφων που είναι αποθηκευμένα στον φάκελο του οποίου αλλάχθηκε το όνομα.

Στο κουμπί της αποθήκευσης πριν αρχίσει να εκτελείται ο κώδικας για την δημιουργία και την αλλαγή των φακέλων ελέγχονται πρώτα τα δεδομένα. Αν υπήρχε κάποια διεύθυνση αποθηκευμένη στη βάση ή αν είχαν δοθεί ονόματα σε κάποιους φακέλους και ο χρήστης τα έσβησε τότε η εφαρμογή δεν τον αφήνει να κάνει την αποθήκευση των αλλαγών.

```

If CheckData() = False Then
    MessageBox.Show("Έχετε αφαιρέσει το όνομα από κάποιο/ους φάκελο/ους και δεν μπορεί να γίνει αποθήκευση. Παρακαλώ δώστε κάποιο όνομα", "Αποθήκευση Ρυθμίσεων", MessageBoxButtons.OK)
    Exit Sub
End If

```

Ο παραπάνω κώδικας εκτελείται στην αρχή του πλήκτρου αποθήκευσης και καλεί την μέθοδο CheckData για να ελέγξει τα δεδομένα. Η μέθοδος CheckData επιστρέφει false όταν ο χρήστης έσβησε κάποια δεδομένα που είχε αποθηκευμένα στη βάση, ενώ επιστρέφει true αν τα δεδομένα δεν έχουν σβηστεί.

```

Private Function CheckData() As Boolean
    If (Not (SavePath = Nothing)) And SavePathTextBox.Text = "" Then
        Return False
    End If

```

```
End If

If (Not (GeneralFolderName = Nothing)) And
GeneralFolderNameTextBox.Text = "" Then
    Return False
End If
If (Not (IncomingFolderName = Nothing)) And
IncomingFolderNameTextBox.Text = "" Then
    Return False
End If
If (Not (OutgoingFolderName = Nothing)) And
OutgoingFolderNameTextBox.Text = "" Then
    Return False
End If

If (Not (TemplateFullPath = Nothing)) And
TemplateFolderNameTextBox.Text = "" Then
    Return False
End If
Return True
End Function
```





## Προτάσεις – Συμπεράσματα

Η εφαρμογή που υλοποιήθηκε υιοθετεί ορισμένες βασικές λειτουργίες και υπηρεσίες που προσφέρει ένα σύστημα διαχείρισης ηλεκτρονικού πρωτοκόλλου. Ωστόσο, υπάρχουν πάντα περιθώρια βελτίωσης και επέκτασης της εφαρμογής για να ενσωματώνει περισσότερες λειτουργίες. Σε μια μεταγενέστερη έκδοση θα μπορούσαν να προστεθούν επιπλέον χαρακτηριστικά ώστε να καλύπτει περισσότερες απαιτήσεις.

Σε αυτήν την ενότητα γίνονται προτάσεις για επέκταση σε συγκεκριμένα θέματα που διαπιστώθηκαν κατά τη διάρκεια ανάπτυξης της εφαρμογής.

Θα μπορούσε έτσι να έχει δημιουργηθεί ένας πίνακας όπου να κρατά τους χρήστες που χειρίζονται την εφαρμογή δίνοντας στον καθένα τα δικαιώματα που του αναλογούν πάνω στην εφαρμογή.

Επίσης θα ήταν καλό να υπήρχε η δυνατότητα σε αυτόν που χειρίζεται την εφαρμογή να μπορεί να στέλνει e-mail σε συγκεκριμένη επαφή στέλνοντας του συγκεκριμένα έγγραφα.

Τέλος μια άλλη λειτουργία που θα μπορούσε να προστεθεί σε μια επόμενη έκδοση της εφαρμογής είναι η αναζήτηση εισερχομένων ή εξερχομένων εγγράφων με βάση συγκεκριμένο αποστολέα και παραλήπτη. Στην τρέχουσα έκδοση του προγράμματος υπάρχει η δυνατότητα αναζήτησης των εγγράφων που σχετίζονται με συγκεκριμένο οργανισμό ή και τμήμα.



## Αναφορές

1. Wikipedia, the free encyclopedia (15 Απριλίου 2009). Microsoft Visual Studio – Wikipedia, the free encyclopedia. Ανακτήθηκε 3, Φεβρουαρίου 2011, από [http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio)
2. Βικιπαίδεια, η ελεύθερη εγκυκλοπαίδεια (29 Απριλίου 2009). Visual Basic – Βικιπαίδεια. Ανακτήθηκε 3, Φεβρουαρίου 2011, από [http://el.wikipedia.org/wiki/Visual\\_Basic](http://el.wikipedia.org/wiki/Visual_Basic)
3. Wikipedia, the free encyclopedia (11 Φεβρουαρίου 2008). Visual Basic – Wikipedia, the free encyclopedia. Ανακτήθηκε 3, Φεβρουαρίου 2011, από [http://en.wikipedia.org/wiki/Visual\\_Basic](http://en.wikipedia.org/wiki/Visual_Basic)
4. Wikipedia, the free encyclopedia (20 Μαΐου 2008). .NET Framework – Wikipedia, the free encyclopedia. Ανακτήθηκε 3, Φεβρουαρίου 2011, από [http://en.wikipedia.org/wiki/.NET\\_Framework](http://en.wikipedia.org/wiki/.NET_Framework)
5. Wikipedia the free encyclopedia (11 Απριλίου 2003). Visual Basic.NET – Wikipedia, the free encyclopedia. Ανακτήθηκε 12, Φεβρουαρίου 2011, από [http://en.wikipedia.org/wiki/Visual\\_Basic\\_.NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)
6. Wikipedia the free encyclopedia (8 Σεπτεμβρίου 2008). Microsoft SQL Server – Wikipedia, the free encyclopedia. Ανακτήθηκε 12, Φεβρουαρίου 2011, από [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)
7. Wikipedia the free encyclopedia (12 Αυγούστου 2008). SQL Server Management Studio – Wikipedia the free encyclopedia. Ανακτήθηκε 12, Φεβρουαρίου 2011 από [http://en.wikipedia.org/wiki/SQL\\_Server\\_Management\\_Studio](http://en.wikipedia.org/wiki/SQL_Server_Management_Studio)

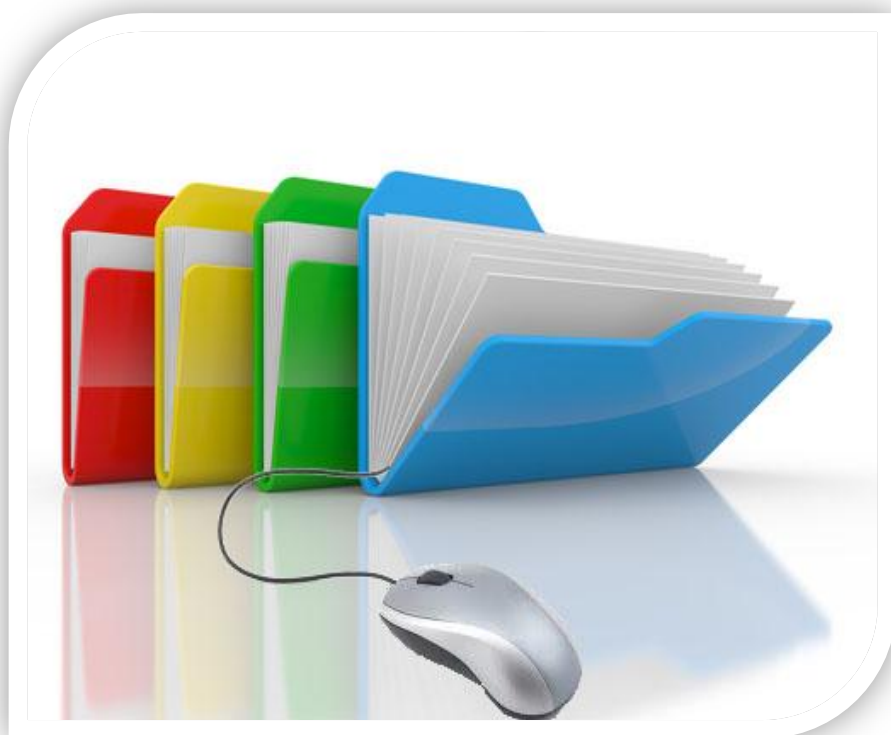


## Βιβλιογραφία

1. Halvorson, M. (2008). Microsoft Visual Basic 2008 Βήμα Βήμα. Αθήνα: Κλειδάριθμος.
2. Πετρούτσος, Ε. (2008). Πλήρες εγχειρίδιο της Microsoft Visual Basic 2008. Αθήνα: Γκιούρδας Μ.
3. Deitel, P. & Deitel, M (2009). Visual Basic 2008 How to Program. New Jersey: Pearson/Prentice Hall.
4. Nielsen, P., White, M. & Parui, U. (2009). Microsoft SQL Server 2008 Bible. Indianapolis: Wiley Publishing, Inc.
5. Dewson, R. (2008). Beginning SQL Server 2008 for Developers. New York: Apress, Inc.



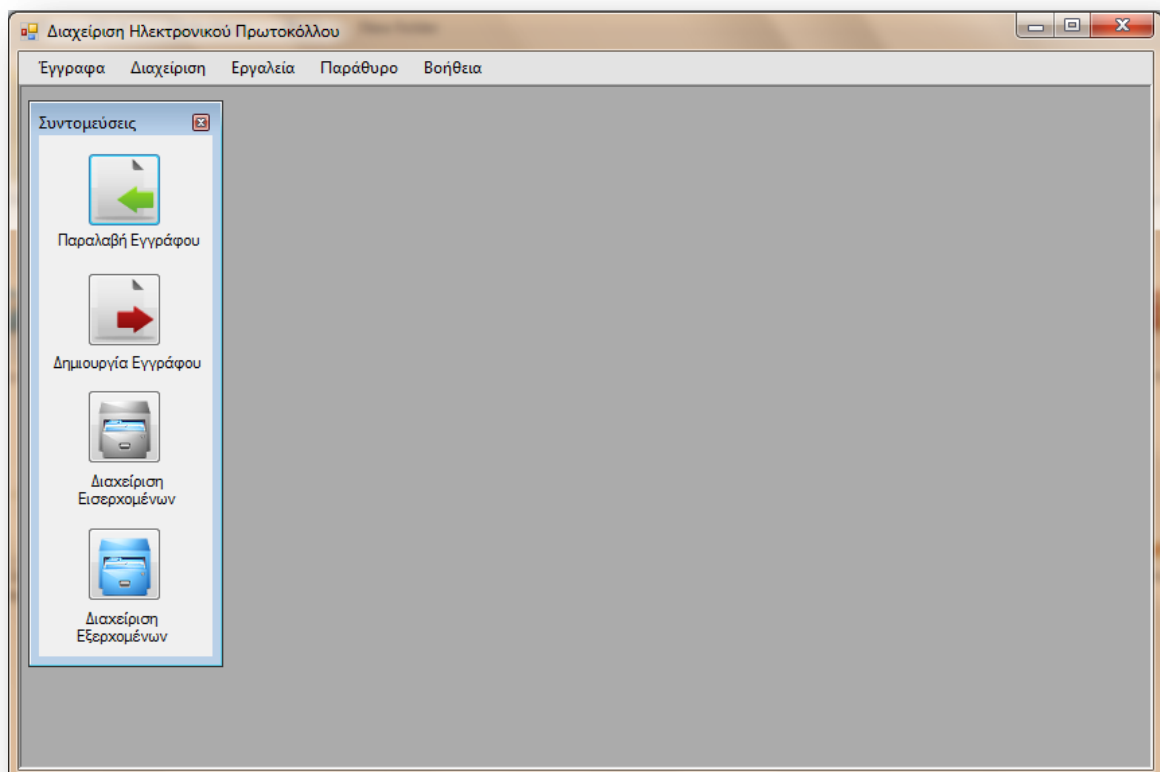
# Οδηγός Χρήσης





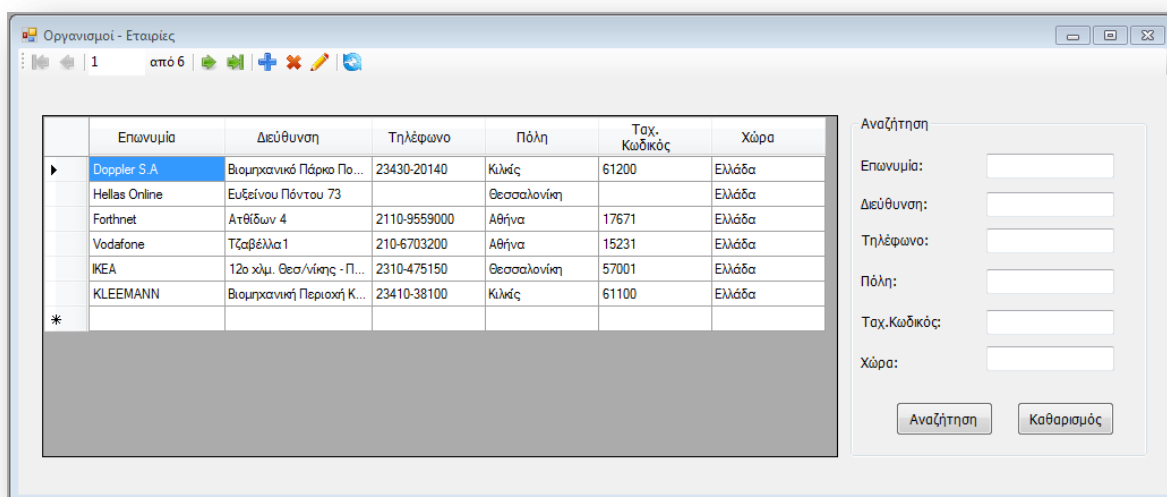
Η παρούσα εφαρμογή δημιουργήθηκε ώστε να δώσει λύση και να διαχειριστεί το πρωτόκολλο μιας εταιρείας ή ενός οργανισμού. Χρησιμοποιώντας την εφαρμογή αυτή μπορείτε να δημιουργήσετε έγγραφα βασισμένα σε πρότυπα και να τα αποθηκεύσετε στο χώρο που επιθυμείται. Αποθήκευση μπορείτε να κάνετε και για τα έγγραφα που εισέρχονται στην εταιρεία. Με τη βοήθεια της εφαρμογής αυτής η αναζήτηση και η ανάκτηση εγγράφων μπορεί να γίνει με πολύ εύκολο και γρήγορο τρόπο.

Η οθόνη που βλέπετε όταν ανοίξετε την εφαρμογή φαίνεται παρακάτω. Στο πάνω μέρος της οθόνης βρίσκεται το μενού με όλες τις λειτουργίες που προσφέρει η εφαρμογή. Όπως θα παρατηρήσετε αριστερά στην οθόνη υπάρχει ένα πλαίσιο με συντομεύσεις. Το πλαίσιο αυτό υπάρχει ώστε να έχετε εύκολη πρόσβαση σε σημαντικές λειτουργίες της εφαρμογής. Αν όμως δεν το χρειάζεστε έχετε τη δυνατότητα να το κλείσετε και μπορείτε να το ανοίξετε όταν το χρειαστείτε από το μενού και την κατηγορία *Εργαλεία*.



## Διαχείριση Οργανισμών

Για να μπορέσετε να έχετε οργανωμένους τους οργανισμούς με τους οποίους συνεργάζεστε δημιουργήθηκε ένα παράθυρο με τη βοήθεια του οποίου μπορείτε να διαχειρίζεστε τους οργανισμούς με τους οποίους συνεργάζεστε.



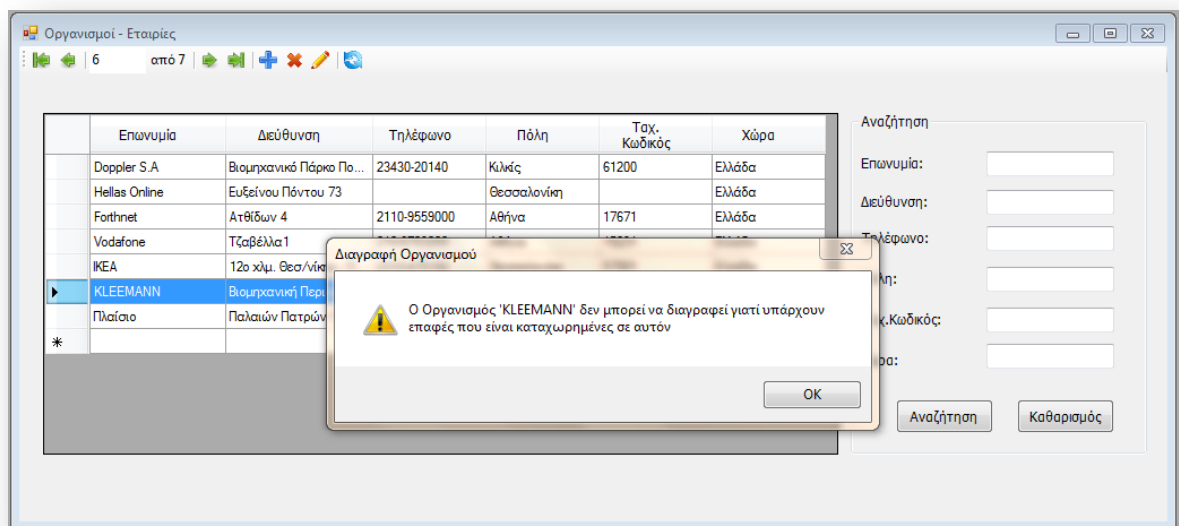
Το παράθυρο αυτό μπορείτε να το βρείτε στο κυρίως μενού της εφαρμογής και συγκεκριμένα στην κατηγορία *Διαχείριση*. Όπως φαίνεται και στη εικόνα παραπάνω, μέσα στο παράθυρο εμφανίζονται όλοι οι οργανισμοί που έχετε καταχωρήσει στην εφαρμογή μαζί με τις πληροφορίες που έχετε για αυτούς.

Στο πάνω μέρος του παραθύρου υπάρχει μια μπάρα εργαλείων που σας βοηθά στη διαχείριση των οργανισμών. Η μπάρα χωρίζεται σε τρία μέρη. Το πρώτο μέρος είναι τα πλήκτρα μετακίνησης ανάμεσα στις εγγραφές που υπάρχουν στον πίνακα. Αν θέλετε να πάτε απευθείας σε κάποια εγγραφή αρκεί να γράψετε τον αριθμό της γραμμής, στην οποία βρίσκετε η εγγραφή αυτή, στο κουτάκι της μπάρας εργαλείων και να πατήσετε το πλήκτρο *Enter*.

Η δεύτερη ομάδα εργαλείων είναι τα εργαλεία διαχείρισης. Όταν θέλετε να δημιουργήσετε ένα νέο οργανισμό αρκεί να πατήσετε το πλήκτρο *Προσθήκη* και θα ανοίξει μια νέα οθόνη μέσα στην οποία θα συμπληρώσετε τα στοιχεία του νέου οργανισμού και θα τον καταχωρήσετε.

Αν θέλετε για κάποιο λόγο να τροποποιήσετε τα στοιχεία ενός ή περισσότερων οργανισμών το μόνο που έχετε να κάνετε είναι να επιλέξετε τους οργανισμούς αυτούς και να πατήσετε το πλήκτρο *Επεξεργασία* στην μπάρα εργαλείων. Για να επιλέξετε κάποιον οργανισμό πρέπει να πατήσετε το κουμπάκι που βρίσκεται δίπλα από το όνομα του. Η ταυτόχρονη επιλογή περισσότερων από έναν οργανισμό γίνεται με την βοήθεια των πλήκτρων *Shift* και *Ctrl* του πληκτρολογίου σας. Κρατήστε ένα από τα δύο πλήκτρα πατημένα και επιλέξτε τους οργανισμούς που θέλετε. Χρησιμοποιείστε το πλήκτρο *Ctrl* για οργανισμούς που δεν είναι στη σειρά και το πλήκτρο *shift* για οργανισμούς που βρίσκεται ο ένας κάτω από τον άλλον. Αφού επιλέξετε τους οργανισμούς που θέλετε και πατήσετε το κατάλληλο πλήκτρο θα εμφανιστεί μια οθόνη, όμοια με αυτήν για την δημιουργία νέου οργανισμού στην οποία θα μπορείτε να κάνετε τις αλλαγές που θέλετε.

Εκτός από την δημιουργία και την επεξεργασία κάποιου οργανισμού, ίσως χρειαστεί να διαγράψετε έναν οργανισμό. Η διαγραφή μπορεί να γίνει αν επιλέξετε τον ή τους οργανισμούς που θέλετε να διαγράψετε και πατήσετε το πλήκτρο *Διαγραφή* της μπάρας εργαλείων. Για να σας επιτρέψει η εφαρμογή να διαγράψετε κάποιον οργανισμό θα πρέπει να μην υπάρχει καμία επαφή που να ανήκει σε αυτόν.

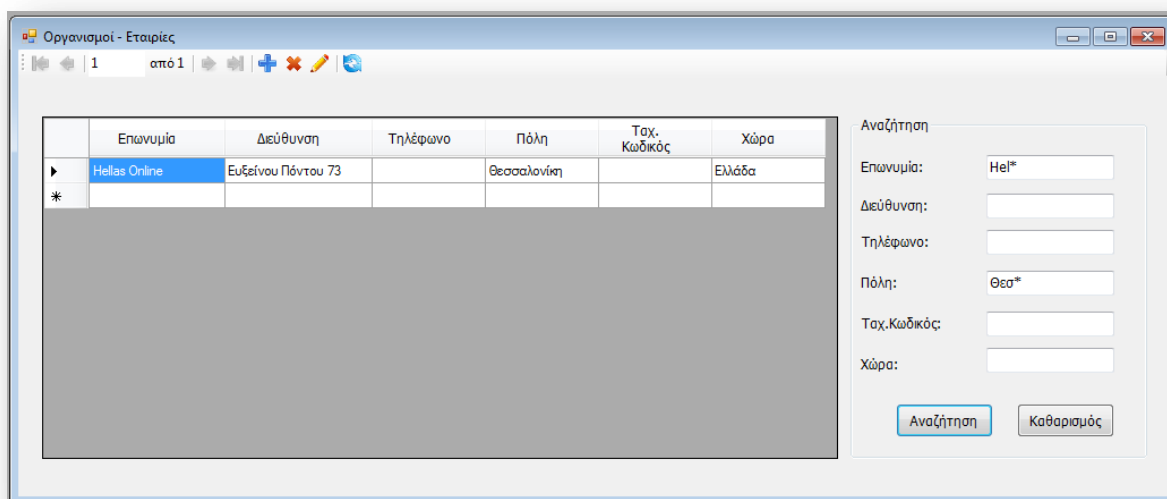


Αν επιχειρήσετε να διαγράψετε έναν οργανισμό ενώ υπάρχουν επαφές που ανήκουν σε αυτόν τότε θα δείτε στην οθόνη σας να εμφανίζεται ένα μήνυμα

αντίστοιχο με αυτό που φαίνεται στην παραπάνω εικόνα. Το μήνυμα σας ενημερώνει ότι ο οργανισμός δεν μπορεί να διαγραφεί.

Τέλος από την μπάρα εργαλείων θα πρέπει να αναφέρουμε το πλήκτρο της *Ανανέωσης*. Με το πλήκτρο αυτό μπορείτε να κάνετε ανανέωση των εγγραφών σας, ώστε να εμφανιστούν όλες οι εγγραφές που υπάρχουν στην εφαρμογή ή για να δείτε αλλαγές που κάνετε στα δεδομένα των οργανισμών σας.

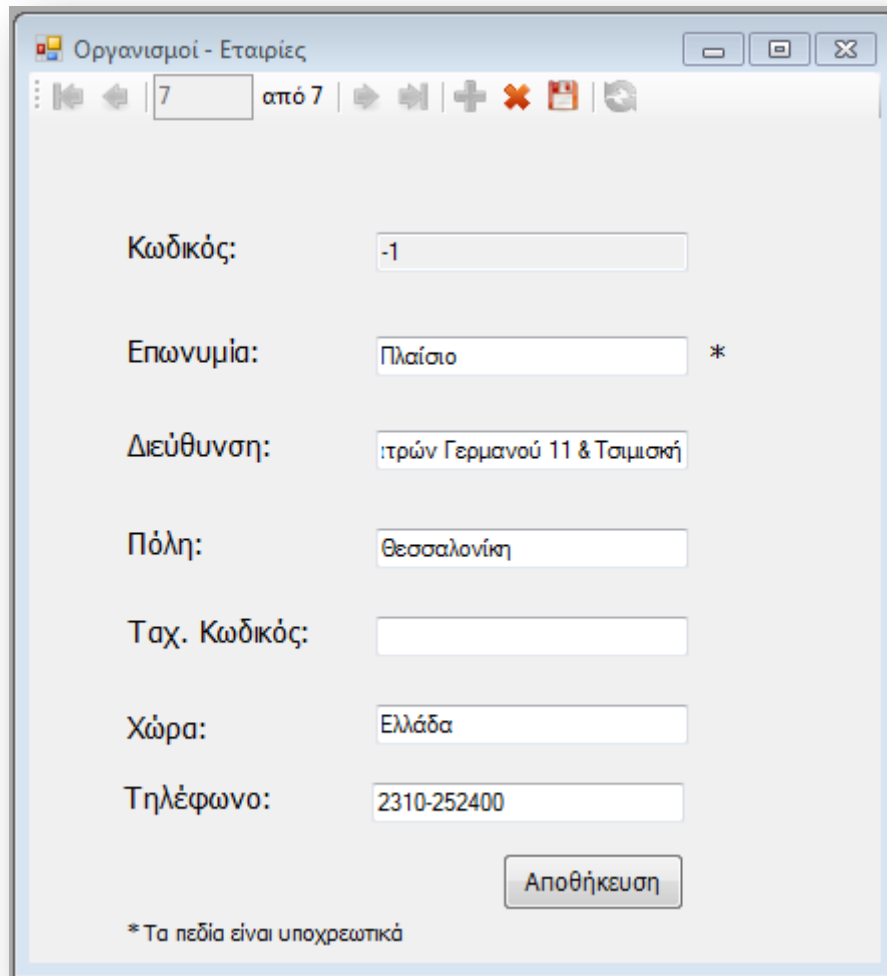
Για καλύτερη διαχείριση των οργανισμών, υπάρχει στο δεξί μέρος του παραθύρου, ένα πλαίσιο αναζήτησης. Στο πλαίσιο αυτό υπάρχουν όλα τα στοιχεία που διαθέτει ένας οργανισμός και τα οποία μπορείτε να χρησιμοποιήσετε για να κάνετε αναζήτηση. Παράδειγμα αναζήτησης φαίνεται στη εικόνα που ακολουθεί. Θα προσέξετε ότι στη αναζήτηση χρησιμοποιούμε τον χαρακτήρα «\*» (αστεράκι). Ο χαρακτήρας αυτός σημαίνει οτιδήποτε και μπορείτε να τον χρησιμοποιήσετε όσες φορές και σε όποιο σημείο της αναζήτησης σας θέλετε. Στο παράδειγμα μας ψάχνουμε να βρούμε τους οργανισμούς που αρχίζουν από Hel και ανήκουν σε κάποια πόλη η οποία αρχίζει από Θεσ.



Όπως έχουμε αναφέρει και πιο πάνω αν θέλετε να εμφανίσετε όλους του οργανισμούς μετά από κάποια αναζήτηση, αρκεί να πατήσετε το πλήκτρο *Ανανέωση* στη μπάρα εργαλείων.

## Δημιουργία – Επεξεργασία Οργανισμού

Το παράθυρο για την δημιουργία ενός οργανισμού ανοίγει μέσα από το παράθυρο της διαχείρισης των οργανισμών, πατώντας το πλήκτρο *Προσθήκη Νέου* από την μπάρα εργαλείων που υπάρχει. Η οθόνη που ανοίγει φαίνεται παρακάτω.



Οργανισμοί - Εταιρίες

7 από 7

Κωδικός: -1

Επωνυμία: Πλαίσιο \*

Διεύθυνση: τερών Γερμανού 11 & Τσιμισκή

Πόλη: Θεσσαλονίκη

Ταχ. Κωδικός:

Χώρα: Ελλάδα

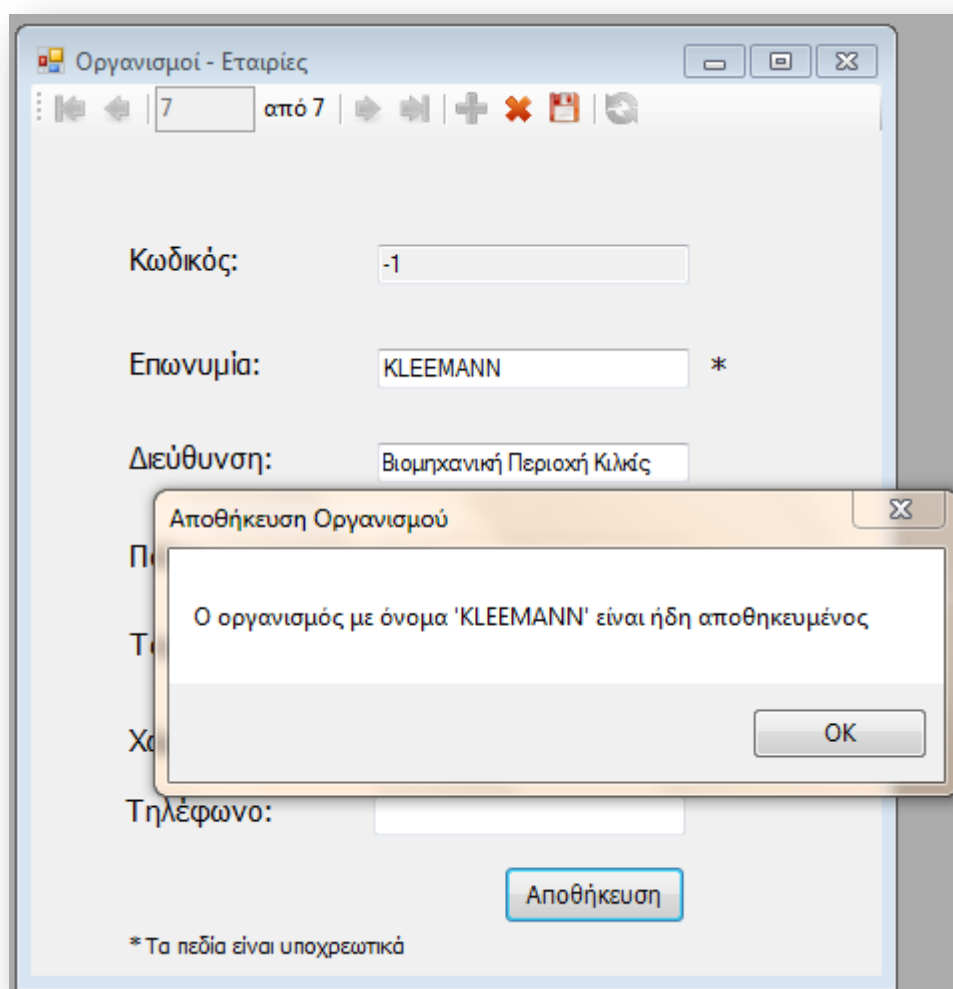
Τηλέφωνο: 2310-252400

Αποθήκευση

\* Τα πεδία είναι υποχρεωτικά

Στο πάνω μέρος του παραθύρου υπάρχει μια μπάρα εργαλείων η οποία μοιάζει με τη μπάρα που αναφέρθηκε στη διαχείριση των οργανισμών. Η μετακίνηση μεταξύ των οργανισμών, η προσθήκη νέου οργανισμού, η διαγραφή και η ανανέωση είναι λειτουργίες που μπορείτε να κάνετε και μέσα από αυτό το παράθυρο.

Κατά τη δημιουργία ενός νέου οργανισμού θα πρέπει να καταχωρήσετε τουλάχιστον τα στοιχεία που εμφανίζουν αστεράκι δίπλα από το κουτάκι τους. Τελειώνοντας την καταχώρηση των στοιχείων του οργανισμού πρέπει να πατήσετε το πλήκτρο *Αποθήκευση*, είτε αυτό που βρίσκεται στο κάτω μέρος του παραθύρου, είτε αυτό που βρίσκεται στη μπάρα εργαλείων, ώστε να περαστεί ο καινούργιος οργανισμός στη εφαρμογή. Στο σημείο αυτό θα πρέπει να γνωρίζεται ότι η εφαρμογή δεν σας αφήνει να περάσετε δύο οργανισμούς με την ίδια επωνυμία. Αν για κάποιο λόγο προσπαθήσετε να το κάνετε θα δείτε να σας εμφανίζεται ένα μήνυμα όπως αυτό που φαίνεται στη εικόνα παρακάτω.



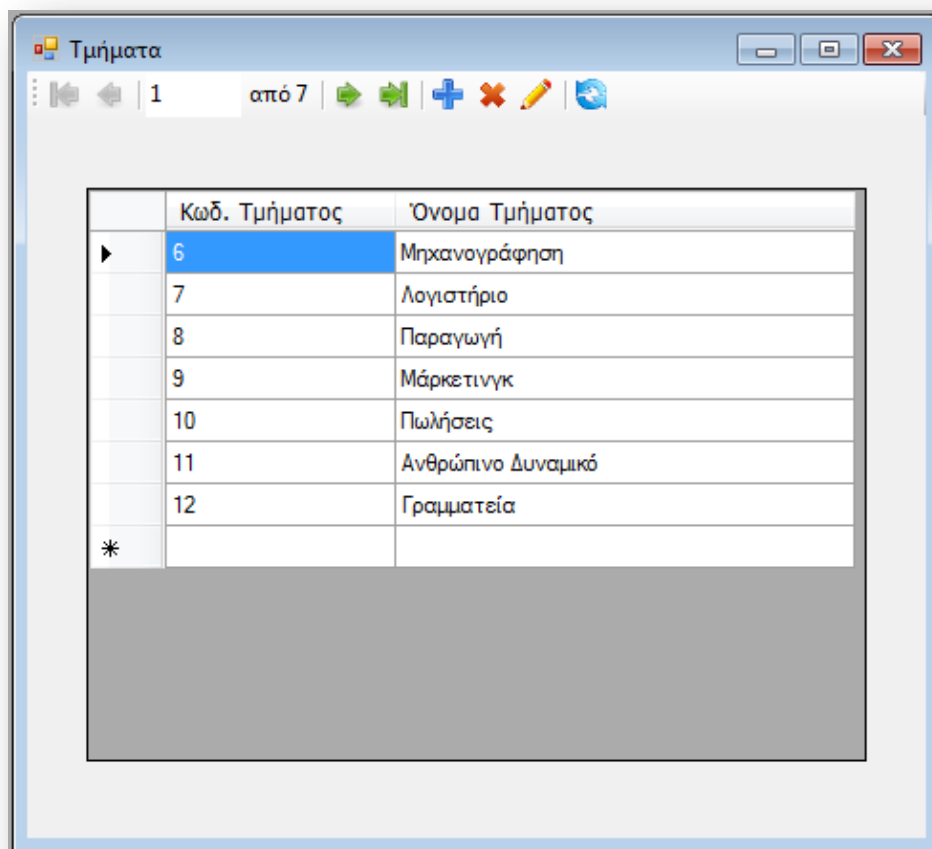
Μέσα από το παράθυρο αυτό μπορείτε να τροποποιήσετε τα στοιχεία ενός ή περισσότερων οργανισμών και να τα αποθηκεύσετε. Θα πρέπει να γνωρίζεται όμως ότι δεν μπορείτε να αλλάξετε την επωνυμία κάποιου οργανισμού και να την κάνετε ίδια με κάποιου άλλου οργανισμού. Σε μια τέτοια περίπτωση η εφαρμογή

θα βγάλει το ίδιο μήνυμα με αυτό που φαίνεται παραπάνω και δεν θα σας επιτρέψει να κάνετε την αλλαγή αυτή.

Αν έχετε κάνει κάποιες αλλαγές και δεν πατήσατε κάποιο από τα πλήκτρα αποθήκευσης ενώ προσπαθήσατε να φύγετε από το παράθυρο αυτό θα δείτε να σας εμφανίζεται ένα μήνυμα. Το μήνυμα αυτό σας ενημερώνει ότι υπάρχουν αλλαγές που δεν τις έχετε αποθηκεύσει και σας ρωτάει αν θέλετε να τις αποθηκεύσει αυτόματα η εφαρμογή πριν να κλείσει το παράθυρο.

## Διαχείριση Τμημάτων

Οι διάφορες επαφές με τις οποίες η εταιρεία ανταλλάσσει έγγραφα εκτός από τον οργανισμό ανήκουν και σε κάποιο τμήμα. Τα διάφορα τμήματα στα οποία μπορεί μια επαφή να ανήκει μπορείτε να τα διαχειριστείτε αν από το κυρίως μενού της εφαρμογής επιλέξετε *Διαχείριση* και στη συνέχεια *Διαχείριση Τμημάτων*.

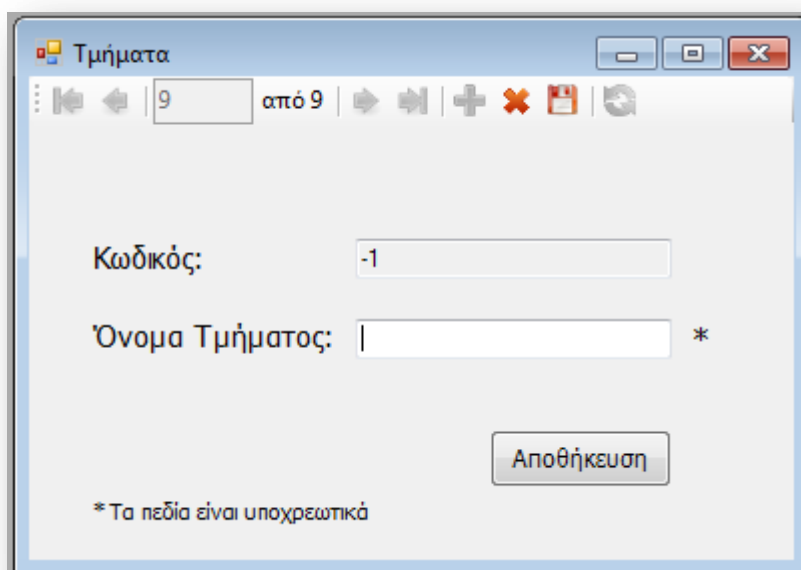


Ανοίγοντας το παράθυρο αυτό μπορείτε να δείτε όλα τα τμήματα που έχετε καταχωρημένα στη εφαρμογή. Επειδή τα τμήματα δεν έχουν πολλά στοιχεία δεν θεωρήθηκε αναγκαίο να υπάρχει αναζήτηση στη φόρμα αυτή. Υπάρχει όμως η μπάρα εργαλείων όπως υπάρχει και σε όλες τις φόρμες διαχείρισης. Τα σημαντικά εργαλεία που υπάρχουν στη μπάρα είναι η δυνατότητα προσθήκης νέου τμήματος, η διαγραφή κάποιου από τα ήδη καταχωρημένα και η τροποποίηση ενός τμήματος.

Για την προσθήκη ενός νέου τμήματος ανοίγει ένα νέο παράθυρο στο οποίο μπορείτε να καταχωρήσετε το τμήμα που θέλετε. Η επεξεργασία συγκεκριμένου τμήματος γίνεται αν επιλέξετε από τον πίνακα το τμήμα που θέλετε και στη συνέχεια πατήσετε από τη μπάρα εργαλείων το πλήκτρο *Επεξεργασία*. Όπως και στη προσθήκη νέου, έτσι και στην τροποποίηση ανοίγει όμοιο παράθυρο στο οποίο μπορείτε να κάνετε τις αλλαγές που θέλετε.

### Προσθήκη - Επεξεργασία Τμήματος

Αν από το παράθυρο της διαχείρισης των τμημάτων επιλέξετε *Προσθήκη Νέου* ή *Επεξεργασία* θα ανοίξει το παράθυρο που βλέπετε παρακάτω.



Τμήματα

9 από 9

Κωδικός: -1

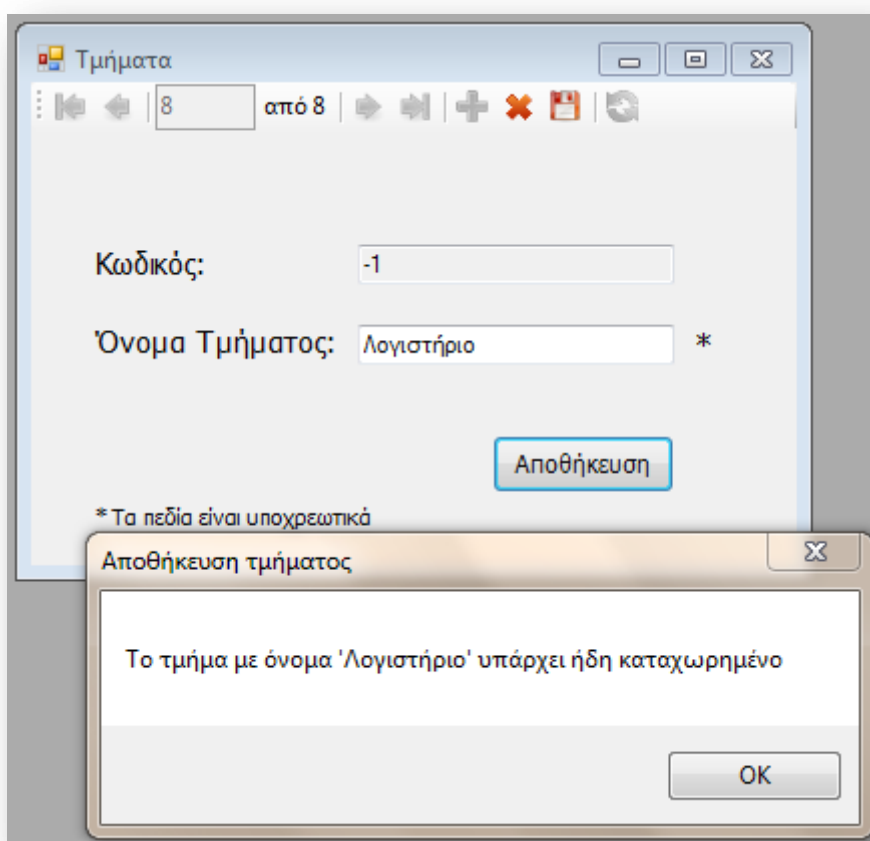
Όνομα Τμήματος: \*

Αποθήκευση

\* Τα πεδία είναι υποχρεωτικά

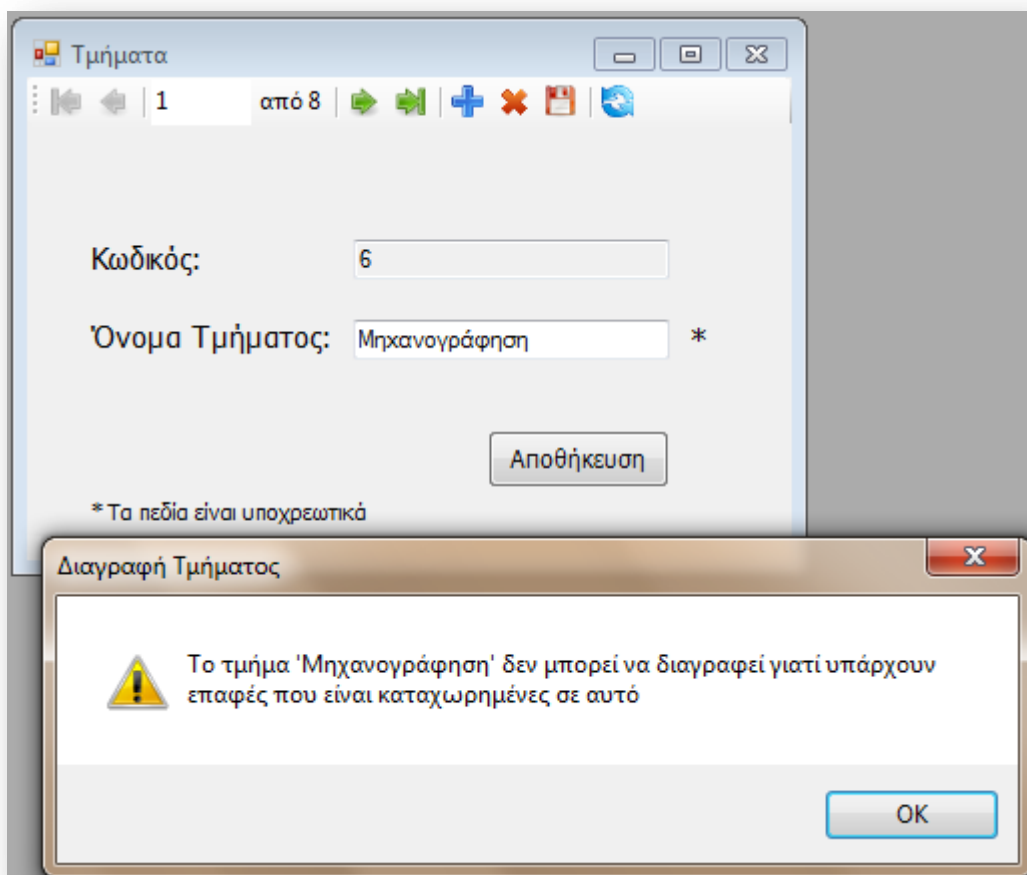


Στο παράθυρο αυτό δημιουργείτε νέο τμήμα ή επεξεργάζεστε κάποιο που ήδη υπάρχει. Και στις δύο περιπτώσεις θα πρέπει να γνωρίζεται ότι υπάρχουν στοιχεία που πρέπει να καταχωρούνται υποχρεωτικά. Τα στοιχεία αυτά αναφέρονται με αστεράκι. Αφού συμπληρώσετε τα στοιχεία που χρειάζονται πρέπει να πατήσετε το πλήκτρο *Αποθήκευση* που βρίσκεται στο κάτω μέρος της οθόνης ή αυτό που βρίσκεται στη μπάρα εργαλείων. Θα πρέπει εδώ να γνωρίζεται ότι δεν μπορείτε να καταχωρήσετε τμήμα με όνομα που ανήκει σε κάποιο άλλο ήδη καταχωρημένο. Σε μια τέτοια περίπτωση θα δείτε στη οθόνη σας να σας εμφανίζεται το ακόλουθο μήνυμα.



Αν για κάποιο λόγο κάνετε αλλαγές ή δημιουργήσετε ένα τμήμα, δεν πατήσετε αποθήκευση και προσπαθήσετε να κλείσετε το παράθυρο, η εφαρμογή θα σας ενημερώσει ότι υπάρχουν αλλαγές που δεν είναι καταχωρημένες και θα σας ρωτήσει αν θέλετε να τις καταχωρήσει. Αν εσείς θέλετε να τις αποθηκεύσετε τότε η εφαρμογή θα αποθηκεύσει τις αλλαγές αυτόματα και θα κλείσει το παράθυρο.

Από το παράθυρο αυτό έχετε επιπλέον τη δυνατότητα να διαγράψετε κάποιο τμήμα. Η εφαρμογή για να επιτρέψει τη διαγραφή κάποιου τμήματος θα πρέπει να μην υπάρχουν επαφές που να ανήκουν σε αυτό. Σε αντίθετη περίπτωση η αίτηση διαγραφής απορρίπτεται και ενημερώνεστε με κατάλληλο μήνυμα. Ένα τέτοιο παράδειγμα φαίνεται στην παρακάτω εικόνα.



## Διαχείριση Επαφών

Η εταιρεία συνεργάζεται με κάποιους οργανισμούς αλλά στη ουσία η επικοινωνία γίνεται μεταξύ των υπαλλήλων της κάθε εταιρείας. Οι διάφοροι υπάλληλοι της κάθε εταιρείας ή οργανισμού με τον οποίο συνεργάζεστε ορίζονται σαν επαφές και διαχειρίζονται μέσα από το παρακάτω παράθυρο. Στην κατηγορία επαφές εντάσσονται και οι υπάλληλοι της ίδιας της εταιρείας ώστε να μπορείτε μέσω της εφαρμογής να διαχειρίζεστε και έγγραφα που διακινούνται εσωτερικά.

Επώνυμο	Όνομα	Email	Δεύθυνση	Πόλη	Ταχ. Κωδικός	Κινητό Τηλ.	Σταθερό Τηλ.	Οργανισμός	Τμήμα
Ιωαννίδης	Σάββας	savioan@t.teithe...	Βενιζέλου 17	Εάνθη	54227			Doppler S.A	Μηχανογράφηση
Κακούλης	Νίκος	kaknik@t.teithe.gr	Άγγελου Σικελια...	Δράμα	66100			Hellas Online	Λογιστήριο
Τρικοπούλου	Χριστίνα							Hellas Online	Γραμματεία
Τερλέγκας	Βασίλης	tbill@doppler.gr		Κιλκίς				Doppler S.A	Μηχανογράφηση
Παπαδόπουλος	Γεώργιος	papgeorge@forth...		Θεσσαλονίκη			2310713246	Forthnet	Πωλήσεις
Κερίρης	Στέφανος	kerstef@hellason...		Θεσσαλονίκη		6994517280		Hellas Online	Γραμματεία
Τραγάκης	Στέφανος	trast@vodafone...	Τρόσσας 9	Αθήνα				Vodafone	Πωλήσεις
Πούλιος	Μάρκος	roulmar@kleema...		Κιλκίς				KLEEMANN	Παραγωγή
Μπίτασας	Μανώλης	mpiman@kleema...		Κιλκίς	61100	6942317289		KLEEMANN	Λογιστήριο

Αναζήτηση

Επώνυμο:  Πόλη:  Οργανισμός:   
 Όνομα:  Δεύθυνση:  Τμήμα:   
 Email:  Ταχ. Κωδικός:

Στο πάνω μέρος του παραθύρου μπορείτε να βρείτε μια μπάρα με εργαλεία που κάνουν το παράθυρο πιο λειτουργικό. Στην μπάρα υπάρχουν εργαλεία μετακίνησης ανάμεσα στις επαφές, εργαλεία διαχείρισης των επαφών και ένα εργαλείο για ανανέωση.

Από τα εργαλεία διαχείρισης μπορείτε να δημιουργήσετε μια νέα επαφή. Πατώντας το πλήκτρο *Προσθήκη Νέου* ανοίγει ένα νέο παράθυρο στο οποίο μπορείτε να καταχωρήσετε τα στοιχεία της νέας σας επαφής και να την αποθηκεύσετε στην εφαρμογή. Επιπλέον μπορείτε να επεξεργαστείτε μια ή περισσότερες επαφές. Για την επεξεργασία θα πρέπει να επιλέξετε τις επαφές που θέλετε και να πατήσετε *Επεξεργασία* από τα εργαλεία. Αμέσως θα δείτε να σας ανοίγει ένα νέο παράθυρο μέσα από το οποίο μπορείτε να κάνετε τις αλλαγές που θέλετε.

Εκτός από τη δημιουργία νέα επαφής μπορείτε μέσα από τη φόρμα αυτή να διαγράψετε και μια επαφή που δεν θέλετε. Προϋπόθεση για να γίνει μια επιτυχής διαγραφή είναι η επαφή να μην σχετίζεται με κάποιο έγγραφο. Αν η επαφή σχετίζεται με ένα ή περισσότερα έγγραφα τότε η εφαρμογή δεν θα επιτρέψει τη διαγραφή και θα εμφανίσει ένα μήνυμα που θα σας ενημερώνει.

The screenshot shows the 'Επαφές' (Contacts) application window. At the top, it indicates '3 από 10' contacts. A table lists various contacts with columns for name, email, address, city, postal code, mobile phone, fixed phone, organization, and department. A dialog box titled 'Διαγραφή Επαφής' (Delete Contact) is overlaid on the table, displaying a warning icon and the message: 'Η επαφή 'Τρικοπούλου Χριστίνα' δεν μπορεί να διαγραφεί γιατί χρησιμοποιείται σε κάποιο έγγραφο' (The contact 'Christina Trikoπούλου' cannot be deleted because it is used in some document). An 'OK' button is visible at the bottom of the dialog.

Επώνυμο	Όνομα	Email	Διεύθυνση	Πόλη	Ταχ. Κωδικός	Κινητό Τηλ.	Σταθερό Τηλ.	Οργανισμός	Τμήμα
Ιωαννίδης	Σάββας	savioan@t.tethe...	Βενιζέλου 17	Εξάνθη	54227			Doppler S.A	Μηχανογράφηση
Κακουλίδης	Νίκος	kakolik@t.tethe.gr	Άγγελου Σικελι...	Δρόμα	66100			Hellas Online	Λογιστήριο
Τρικοπούλου	Χριστίνα	trikchi@hellasonl...		Αθήνα	44178			Hellas Online	Γραμματεία
Τερλέγκας	Βασίλης	tbill@doppler.gr		Κιλκίς				Doppler S.A	Μηχανογράφηση
Παπαδόπουλος	Γεώργιος	papageorge@forth...		Θεσσαλονίκη			2310713246	Forthnet	Πωλήσεις
Κερκίρης	Στέφανος	kerstef@hellasonl...		Θεσσαλονίκη		6994517280		Hellas Online	Γραμματεία
Τραγάσης	Στέφανος	tragst@vodafone...	Τράσας 9	Αθήνα				Vodafone	Πωλήσεις
Πούλος	Μάρκος	roulmar@kleema...						KLEEMANN	Παραγωγή
Μπίτσας	Μανώλης	mpiman@kleema...						KLEEMANN	Λογιστήριο
Τουλούσης	Κωνσταντίνος	toukwn@hellaso...						Hellas Online	Ανθρώπινο Δυνα...

Αναζήτηση

Επώνυμο:  Πόλη:  Οργανισμός: Doppler S.A, Fortlnet, Hellas Online, KLEEMANN, Vodafone, ΙΚΕΑ, Πλαίσιο Τμήμα: Ανθρώπινο Δυναμικό, Γραμματεία, Δημόσιες Σχέσεις, Λογιστήριο, Μάρκετινγκ, Μηχανογράφηση, Παραγωγή, Πωλήσεις

Όνομα:  Διεύθυνση:  Καθαρισμός

Email:  Ταχ. Κωδικός:  Αναζήτηση

Αν έχετε επιλέξει περισσότερες από μια έγγραφες για διαγραφή η διαδικασία δεν θα διακοπεί, αν υπάρχει πρόβλημα με την διαγραφή κάποιων από τις επαφές. Η εφαρμογή θα βγάλει μήνυμα και θα ενημερώνει για το ποιες επαφές δεν μπορούν να διαγραφούν και θα συνεχίσει κανονικά τη διαδικασία μέχρι το τέλος.

The screenshot shows the 'Επαφές' (Contacts) application window after a search. It indicates '1 από 2' results. The table shows two contacts: 'Τρικοπούλου Χριστίνα' and 'Τουλούσης Κωνσταντίνος'. The search filters are updated: 'Επώνυμο' is 'Τ\*', 'Οργανισμός' is 'Hellas Online', and 'Τμήμα' is 'Ανθρώπινο Δυναμικό'.

Επώνυμο	Όνομα	Email	Διεύθυνση	Πόλη	Ταχ. Κωδικός	Κινητό Τηλ.	Σταθερό Τηλ.	Οργανισμός	Τμήμα
Τρικοπούλου	Χριστίνα	trikchi@hellasonl...		Αθήνα	44178			Hellas Online	Γραμματεία
Τουλούσης	Κωνσταντίνος	toukwn@hellaso...		Αθήνα	56190			Hellas Online	Ανθρώπινο Δυνα...

Αναζήτηση

Επώνυμο: Τ\* Πόλη:  Οργανισμός: Doppler S.A, Fortlnet, Hellas Online, KLEEMANN, Vodafone, ΙΚΕΑ Τμήμα: Ανθρώπινο Δυναμικό, Γραμματεία, Λογιστήριο, Μάρκετινγκ, Μηχανογράφηση, Παραγωγή, Πωλήσεις

Όνομα:  Διεύθυνση:  Καθαρισμός

Email:  Ταχ. Κωδικός:  Αναζήτηση

Επειδή υπάρχει μεγάλη πιθανότητα η επαφές με τις οποίες συνεργάζεστε να είναι πάρα πολλές στο παράθυρο υπάρχει και ένα πλαίσιο αναζήτησης. Με τη βοήθεια της αναζήτησης θα μπορείτε πιο εύκολα να βρείτε την ή τις επαφές που θέλετε κάθε φορά. Τα πεδία που προσφέρονται για αναζήτηση είναι όλα τα στοιχεία που μπορεί να έχει μια επαφή. Ένα παράδειγμα αναζήτησης φαίνεται στη παραπάνω εικόνα όπου ψάχνουμε τις επαφές που το επώνυμο τους αρχίζει από Τ και ανήκουν στον οργανισμό Hellas Online. Θα παρατηρήσετε ότι στη αναζήτηση με βάση το επώνυμο χρησιμοποιήθηκε ο χαρακτήρας «\*». Ο χαρακτήρας αυτός αντιπροσωπεύει το οτιδήποτε και μπορείτε να τον χρησιμοποιήσετε όσες φορές και σε όποιο σημείο της αναζήτησης θέλετε.

### Δημιουργία - Επεξεργασία Επαφής

Μια νέα επαφή μπορείτε να δημιουργήσετε αν μέσα από το παράθυρο διαχείρισης επαφών πατήσετε το πλήκτρο *Προσθήκη Νέου*. Το πλήκτρο αυτό ανοίγει ένα νέο παράθυρο. Το παράθυρο αυτό είναι το ίδιο με αυτό που ανοίγει όταν θέλετε να επεξεργαστείτε κάποια επαφή.

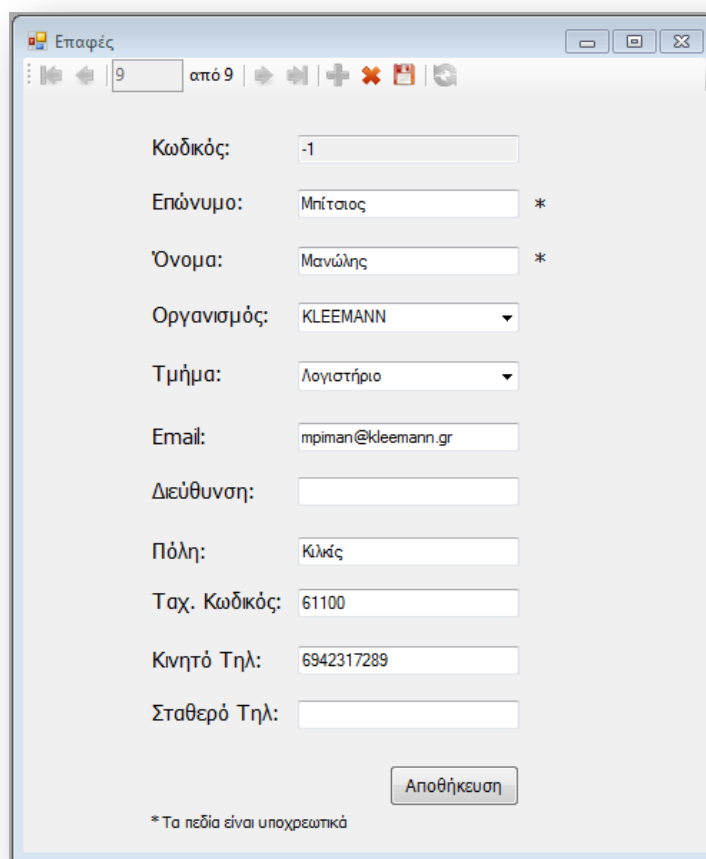
Όπως θα δείτε στο παράθυρο αυτό υπάρχουν υποχρεωτικά πεδία τα οποία έχουν ένα αστεράκι δίπλα από το κουτί καταχώρησης τους. Δεν μπορείτε να δημιουργήσετε κάποια επαφή αν δεν καταχωρήσετε τουλάχιστον τα πεδία αυτά. Κάθε επαφή εκτός από τα προσωπικά της στοιχεία ανήκει σε κάποιον οργανισμό και σε κάποιο τμήμα τα οποία μπορείτε να ορίσετε μέσα από το παράθυρο αυτό.

Εκτός από τη δημιουργία και τη επεξεργασία επαφής, προσφέρονται και άλλες λειτουργίες οι οποίες βρίσκονται στην μπάρα στο πάνω μέρος του παραθύρου. Με τη βοήθεια αυτών των εργαλείων μπορείτε να μετακινηθείτε ανάμεσα στις επαφές που έχετε καταχωρημένες, να δημιουργήσετε μια καινούργια επαφή, να διαγράψετε κάποια που ίσως δεν θέλετε και να αποθηκεύσετε τις αλλαγές που έχετε κάνει.

Η εφαρμογή δεν ελέγχει στην αποθήκευση αν μια επαφή είναι ήδη καταχωρημένη, όπως κάνει με τους οργανισμούς και τα τμήματα. Επιτρέπει την καταχώρηση δύο επαφών με το ίδιο όνομα και επώνυμο.

Παρόλα αυτά η εφαρμογή είναι αυστηρή στη διαγραφή μιας επαφής. Για να επιτύχει η διαγραφή κάποια επαφής θα πρέπει αυτή η επαφή να μην σχετίζεται με κάποιο έγγραφο. Σε αντίθετη περίπτωση η διαγραφή αποτυγχάνει.

Παράδειγμα του παράθυρου δημιουργίας – επεξεργασίας επαφής φαίνεται στη εικόνα που ακολουθεί.



The screenshot shows a web application window titled 'Επαφές'. The interface includes a navigation bar with a search box containing '9' and the text 'από 9'. Below the navigation bar is a form with the following fields:

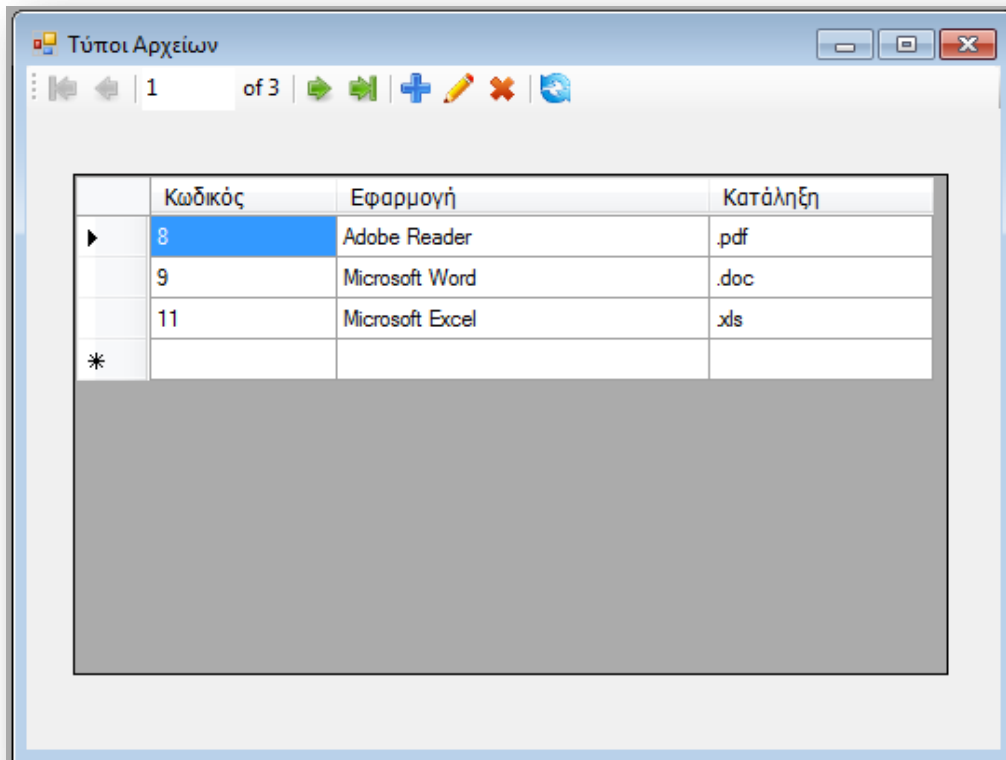
- Κωδικός: -1
- Επώνυμο: Μπίτσιος \*
- Όνομα: Μανώλης \*
- Οργανισμός: KLEEMANN (dropdown menu)
- Τμήμα: Λογιστήριο (dropdown menu)
- Email: mpiman@kleemann.gr
- Διεύθυνση: (empty text box)
- Πόλη: Κιλκίς
- Ταχ. Κωδικός: 61100
- Κινητό Τηλ: 6942317289
- Σταθερό Τηλ: (empty text box)

At the bottom of the form is a button labeled 'Αποθήκευση'. Below the button, there is a note: '\* Τα πεδία είναι υποχρεωτικά'.

## Διαχείριση Τύπων Αρχείου

Κάθε έγγραφο εισερχόμενο ή εξερχόμενο καταχωρείται στην εφαρμογή σε ηλεκτρονική μορφή. Ο τύπος του κάθε αρχείου που δημιουργείται για το κάθε έγγραφο είναι μια πληροφορία που θεωρήθηκε καλό να καταχωρείται στη εφαρμογή. Οι διάφοροι τύποι αρχείου διαχειρίζονται μέσα από το παράθυρο που

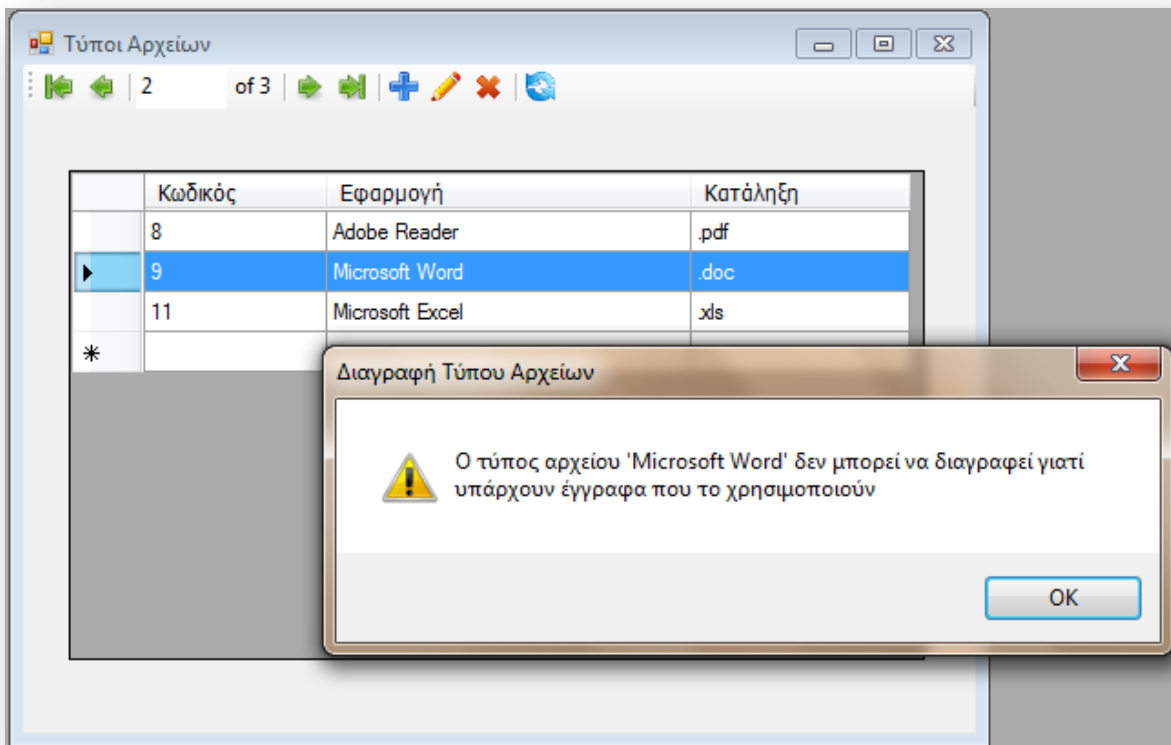
φαίνεται παρακάτω. Για να ανοίξετε το παράθυρο αυτό πρέπει από το κεντρικό μενού να επιλέξετε *Διαχείριση* και στη συνέχεια *Διαχείριση Τύπων Αρχείου*.



Στο παράθυρο αυτό υπάρχει μια μπάρα εργαλείων στο πάνω μέρος ώστε να μπορείτε να διαχειρίζεστε τους τύπους αρχείου. Αν θέλετε να δημιουργήσετε ένα νέο τύπο αρχείου θα πρέπει να επιλέξετε *Προσθήκη Νέου* από την μπάρα, αμέσως ένα νέο παράθυρο θα ανοίξει στο οποίο μπορείτε να καταχωρήσετε τα στοιχεία που θέλετε.

Μέσα από το παράθυρο αυτό έχετε τη δυνατότητα να ζητήσετε να επεξεργαστείτε έναν ή περισσότερους τύπους αρχείων, επιλέγοντας τους και πατώντας το πλήκτρο *Επεξεργασία*. Στη περίπτωση αυτή θα ανοίξει ένα νέο παράθυρο όπως αυτό που ανοίγει στη δημιουργία νέου τύπου αρχείου.

Η εφαρμογή σας επιτρέπει να διαγράψετε ένα τύπο αρχείου όταν αυτός δεν σχετίζεται με κάποιο έγγραφο. Κάθε προσπάθεια διαγραφής ενός τύπου που αναφέρετε σε κάποιο έγγραφο θα έχει σας αποτέλεσμα την εμφάνιση ενός μηνύματος ενημέρωσης και θα απορρίπτετε.



## Δημιουργία - Επεξεργασία Τύπου Αρχείων

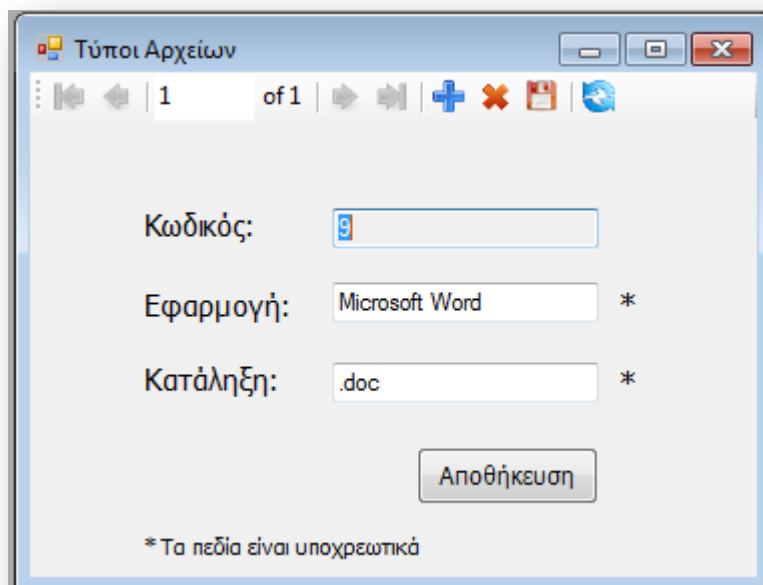
Για να μπορέσετε να ανοίξετε το παράθυρο για τη δημιουργία νέου τύπου αρχείων ή την επεξεργασία ενός ήδη καταχωρημένου θα πρέπει να πατήσετε την αντίστοιχη επιλογή από τη μπάρα εργαλείων που υπάρχει στο παράθυρο για τη διαχείριση των τύπων αρχείου.

Μέσα από το παράθυρο αυτό μπορείτε να χρησιμοποιήσετε τις λειτουργίες της μετακίνησης ανάμεσα στις εγγραφές, της δημιουργίας νέου τύπου αρχείων, τη διαγραφή του, την αποθήκευση των αλλαγών και την ανανέωση των εγγραφών.

Στη διαγραφή να θυμάστε ότι δεν μπορείτε να διαγράψετε ένα αρχείο που σχετίζεται με κάποιο εισερχόμενο ή εξερχόμενο έγγραφο.

Ένα παράδειγμα καταχώρησης τύπου αρχείου φαίνεται στην εικόνα που ακολουθεί.





## Διαμόρφωση Εφαρμογής

Τα εισερχόμενα και τα εξερχόμενα έγγραφα που καταχωρούνται στη εφαρμογή αποθηκεύονται σε κάποιο χώρο, σε κάποια διεύθυνση ενός υπολογιστή. Η διεύθυνση στην οποία αποθηκεύονται τα έγγραφα σας ορίζεται από εσάς σύμφωνα με τις ανάγκες σας. Προσπαθήστε η διεύθυνση που θα επιλέξετε να είναι σε κάποιον υπολογιστή που μπορούν να έχουν πρόσβαση όλοι οι χρήστες της εφαρμογής.

Για να μπορέσετε να διαμορφώσετε την εφαρμογή σύμφωνα με τις ανάγκες σας θα πρέπει από το μενού να πατήσετε εργαλεία και στη συνέχεια *Διαμόρφωση Ρυθμίσεων*. Το παράθυρο που θα ανοίξει θα σας ζητήσει να ορίσετε κάποια στοιχεία. Το σημαντικότερο είναι η διεύθυνσή στη οποία επιθυμείτε να αποθηκεύει η εφαρμογή τα έγγραφα σας. Επιπλέον πρέπει να δώσετε το όνομα που θέλετε στον γενικό φάκελο των εγγράφων σας καθώς και στον φάκελο που θα καταχωρούνται τα εισερχόμενα καθώς και τα εξερχόμενα έγγραφα.

Τέλος για να μπορείτε να δημιουργήσετε κάποιο έγγραφο μέσα από την εφαρμογή θα πρέπει να καταχωρήσετε κάποια πρότυπα. Τα πρότυπα που θα καταχωρήσετε η εφαρμογή θα τα αποθηκεύσει στη διεύθυνση που θα ορίσετε

εσείς. Δεν χρειάζεται όμως να δώσετε όνομα στο φάκελο που θα είναι τα πρότυπα καθώς η εφαρμογή δίνει αυτόματα ένα όνομα στον φάκελο των προτύπων.

Ένα παράδειγμα του παράθυρου των ρυθμίσεων εμφανίζεται παρακάτω.

Διαμορφώση Εφαρμογής

Διεύθυνση Αποθήκευσης Εγγράφων: D:\Πτυχειακή Έυρεση

Όνομασία Γενικού Φακέλου Εγγράφων: Protocol

Όνομασία Φακέλου Εισερχομένων: Inbox

Όνομασία Φακέλου Εξερχομένων: Outbox

Διεύθυνση Αποθήκευσης Προτύπων: D:\Πτυχειακή\Templates Έυρεση

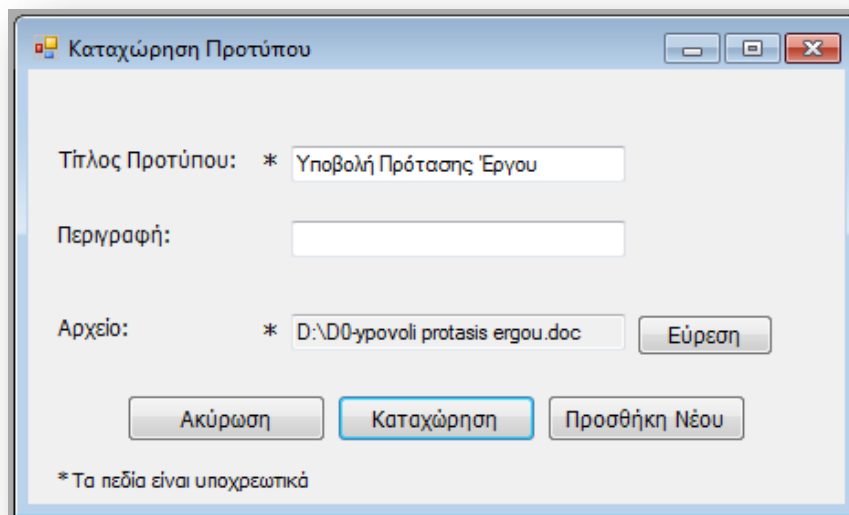
Αποθήκευση

## Καταχώρηση Προτύπου

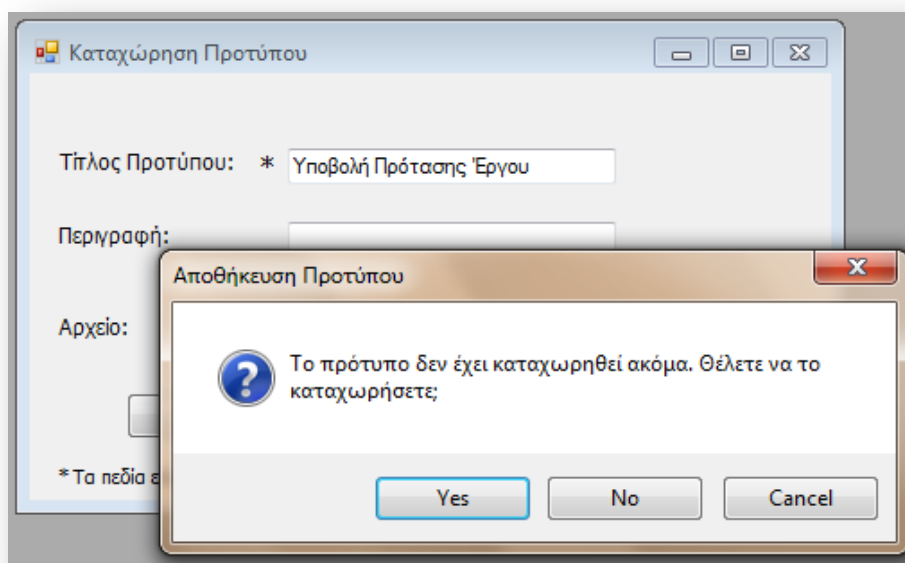
Για να μπορέσετε μέσα από την εφαρμογή να δημιουργήσετε κάποιο εξερχόμενο έγγραφο θα πρέπει να έχετε προηγουμένως καταχωρήσει κάποιο πρότυπο. Σύμφωνα με τη λογική που αναπτύχθηκε η εφαρμογή, τα εξερχόμενα αρχεία βασίζονται πάνω σε πρότυπα τα οποία τροποποιούνται ώστε να δημιουργηθεί το έγγραφο. Το παράθυρο για την καταχώρηση προτύπου μπορείτε να το ανοίξετε από το κυρίως μενού και συγκεκριμένα από τα εργαλεία.

Η καταχώρηση προτύπου είναι μια απλή λειτουργία. Το μόνο που έχετε να κάνετε είναι να έχετε το πρότυπο σε ηλεκτρονική μορφή και με τη βοήθεια του

παραθύρου που φαίνεται παρακάτω να καταχωρήσετε το πρότυπο. Στο παράθυρο αυτό θα πρέπει να επιλέξετε το πρότυπο που θέλετε να καταχωρήσετε, να του δώσετε ένα τίτλο και αν θέλετε και μια περιγραφή. Αφού το κάνετε αυτό πρέπει να πατήσετε το πλήκτρο *Καταχώρηση* ώστε το πρότυπο να μεταφερθεί στη διεύθυνση αποθήκευσης προτύπων με όνομα τον τίτλο που του έχετε δώσει.

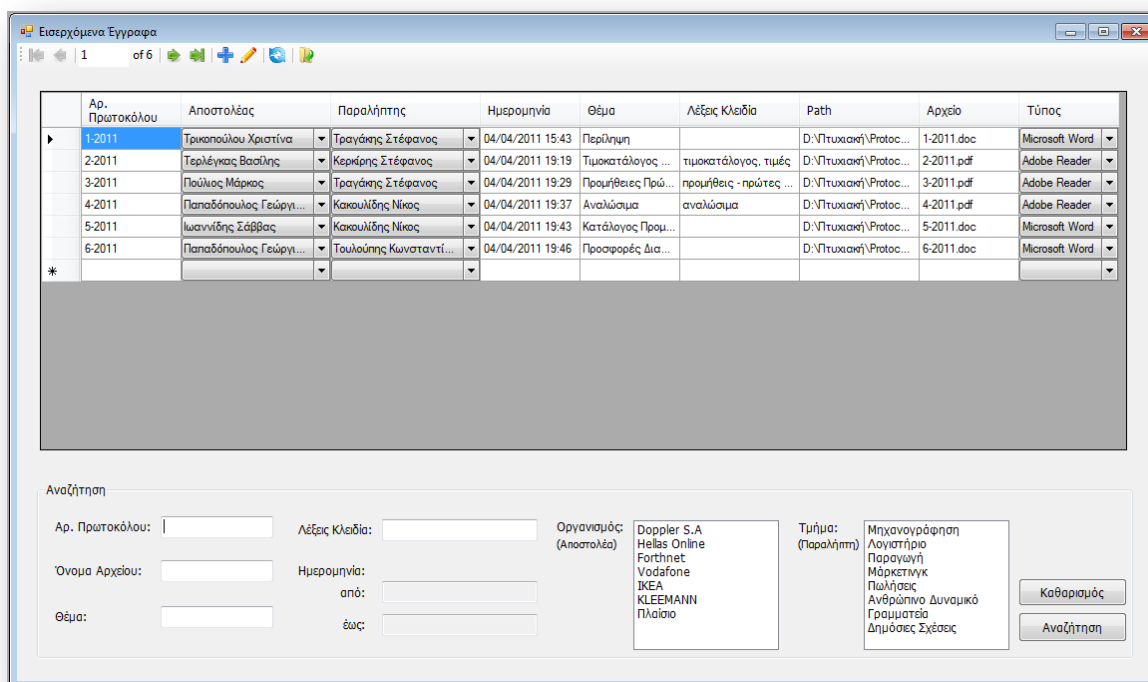


Αν τελικά δεν θέλετε να καταχωρήσετε το πρότυπο που έχετε επιλέξει μπορείτε να πατήσετε το πλήκτρο *Ακύρωση* ώστε να διακοπεί η λειτουργία αν όντως αυτό επιθυμείτε. Τέτοιο παράδειγμα φαίνεται στη εικόνα που ακολουθεί.



## Διαχείριση Εισερχόμενων Εγγράφων

Αν επιθυμείτε να δείτε τα εισερχόμενα έγγραφα σας ή να αναζητήσετε κάποιο από αυτά θα πρέπει να ανοίξετε το παράθυρο *Διαχείριση Εισερχόμενων Εγγράφων*. Το παράθυρο μπορείτε να το ανοίξετε είτε από τις συντομεύσεις, είτε από το μενού *Έγγραφα* και στη συνέχεια *Διαχείριση Εισερχομένων*.

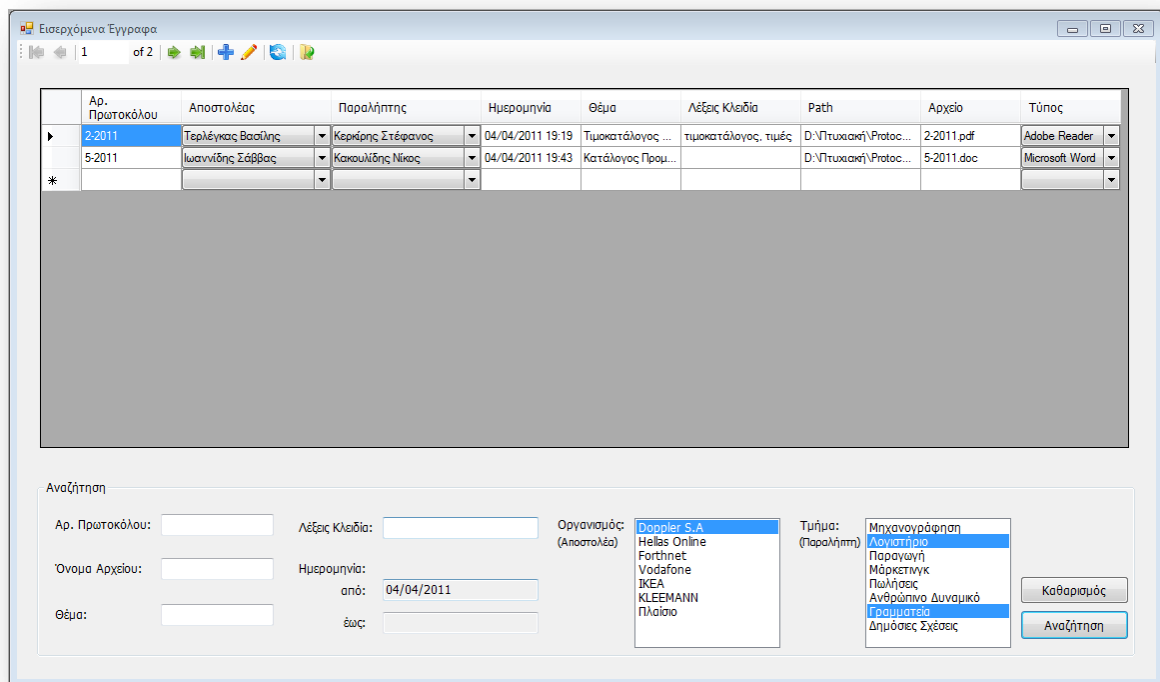


Το παράθυρο που ανοίγει έχει τη μορφή που φαίνεται στη εικόνα. Στο πάνω μέρος υπάρχει μια μπάρα εργαλείων. Στο κέντρο εμφανίζονται όλα τα εισερχόμενα έγγραφα και στο κάτω μέρος υπάρχει ένα πλαίσιο στο οποίο μπορείτε να κάνετε αναζήτηση.

Τα σημαντικότερα εργαλεία που υπάρχουν στη μπάρα είναι η *Προσθήκη Νέου*, κατά την οποία ανοίγει η φόρμα για την παραλαβή εισερχόμενου εγγράφου, η *Επεξεργασία* του εγγράφου, κατά την οποία η εφαρμογή επιτρέπει την αλλαγή συγκεκριμένων στοιχείων του εγγράφου και τέλος η *Εμφάνιση Εγγράφου*, η οποία ανοίγει το επιλεγμένο έγγραφο.

Στο πλαίσιο της αναζήτησης μπορείτε να κάνετε αναζήτηση εγγράφων με βάση κάποια πεδία. Στην αναζήτηση υπάρχει δυνατότητα να αναζητήσετε έγγραφα που

σχετίζονται με κάποιον οργανισμό ή/και κάποιο τμήμα. Επειδή τα εισερχόμενα έγγραφα γνωρίζετε ότι στάλθηκαν στη δική σας εταιρεία όταν κάνετε αναζήτηση με βάση το τμήμα, ελέγχεται το τμήμα του παραλήπτη, ενώ όταν γίνεται με βάση τον οργανισμό, ελέγχεται ο οργανισμός του αποστολέα. Στη εικόνα παρακάτω φαίνεται ένα παράδειγμα αναζήτησης.



## Παραλαβή Εγγράφου

Όταν ένα έγγραφο εισέλθει στη εταιρεία τότε αυτόματα χαρακτηρίζεται ως εισερχόμενο και πρέπει να καταχωρηθεί στην εφαρμογή. Η καταχώρηση αυτή ορίζεται σαν παραλαβή εγγράφου και το παράθυρο της μπορείτε να το ανοίξετε από τρία σημεία. Από τις συντομεύσεις, από το κυρίως μενού *Έγγραφα – Παραλαβή Εγγράφου* και μέσα από το παράθυρο της διαχείρισης των εισερχομένων.

Μόλις επιλέξετε παραλαβή εγγράφου θα δείτε στο παράθυρο που θα ανοίξει να εμφανίζεται αυτόματα ο αριθμός πρωτοκόλλου του εγγράφου. Εσείς θα πρέπει να

συμπληρώσετε όλα τα πεδία που έχουν οριστεί ως υποχρεωτικά ώστε η εφαρμογή να σας αφήσει να καταχωρήσετε το έγγραφο.

Παραλαβή Εγγράφου

1 of 1

Id Protocol: -1

Αρ. Πρωτοκόλου: 1-2011

Τύπος Εγγράφου:  Εισερχόμενο  Εξερχόμενο

Αποστολέας: Ιωαννίδης Σάββας Αναζήτηση

Παραλήπτης: Τερλέγκας Βασίλης Αναζήτηση

Ημερομηνία: 03 April 2011

Θέμα:

Λέξεις Κλειδιά:

Αρχείο: Έρευνα

Τύπος Αρχείου:

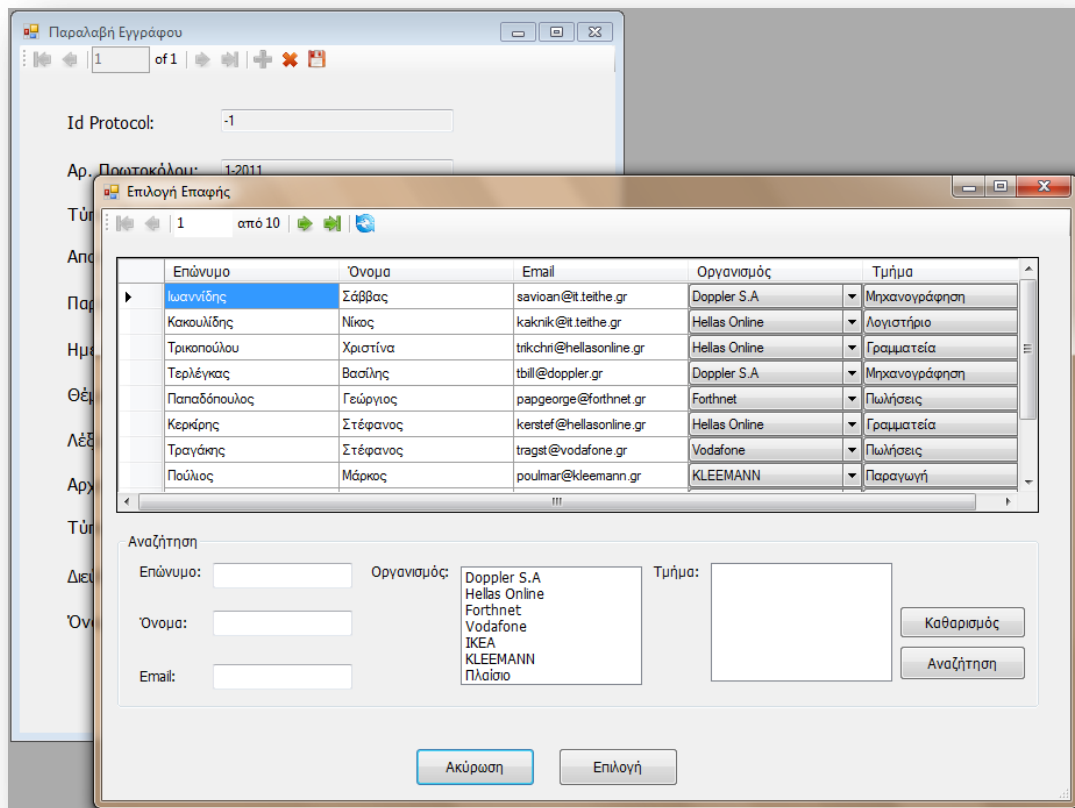
Διεύθυνση:

Όνομα Αρχείου:

Ακύρωση Αποθήκευση Εγγράφου

Στα στοιχεία που πρέπει να καταχωρηθούν για το έγγραφο είναι ο αποστολέας και ο παραλήπτης του. Επειδή οι επαφές που μπορεί να υπάρχουν είναι πάρα πολλές δημιουργήθηκε ένα παράθυρο στο οποίο σας βοηθά να αναζητήσετε την επαφή που είναι παραλήπτης και την επαφή που είναι αποστολέας. Για να ανοίξει το παράθυρο αυτό αρκεί να πατήσετε δίπλα από τον παραλήπτη και τον αποστολέα το πλήκτρο *Αναζήτηση*. Στο παράθυρο που θα ανοίξει θα επιλέξετε την επαφή που θέλετε και θα πατήσετε *Επιλογή*. Προσέχετε όμως να μην επιλέξετε ίδια επαφή σαν αποστολέα και σαν παραλήπτη. Κάτι τέτοιο η εφαρμογή

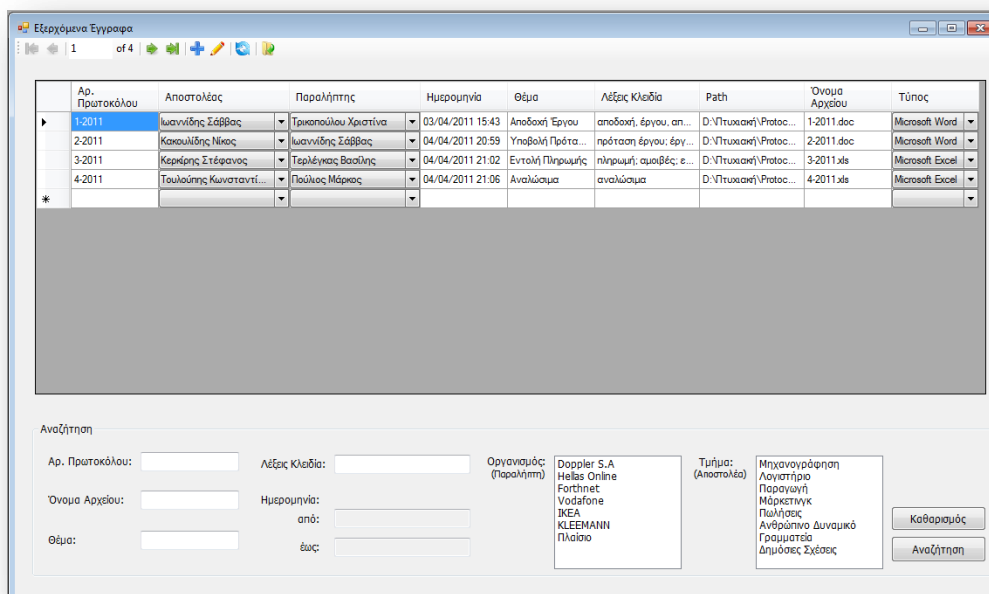
το εμποδίζει και θα σας υποχρεώσει να επιλέξετε διαφορετικές επαφές. Παράδειγμα αναζήτησης επαφής φαίνεται στη παρακάτω εικόνα.



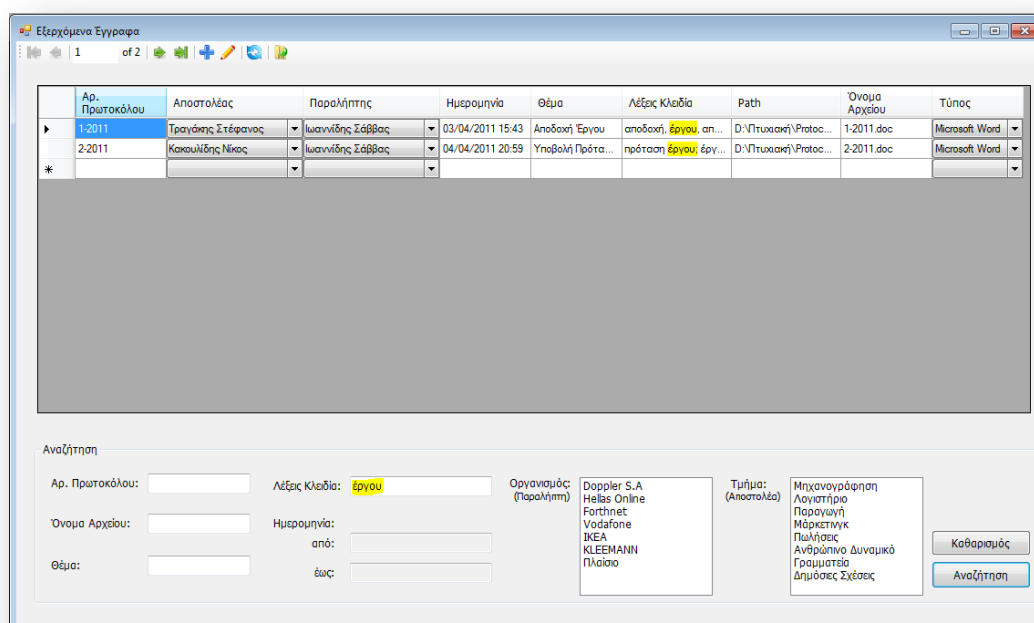
Αν για κάποιο λόγο επιθυμείτε να ακυρώσετε την παραλαβή εγγράφου τότε μπορείτε να πατήσετε το πλήκτρο Ακύρωση από το κάτω μέρος του παραθύρου ή από τη μπάρα εργαλείων. Αμέσως θα δείτε ένα μήνυμα που θα σας ρωτάει αν θέλετε όντως να ακυρώσετε τη διαδικασία. Ανάλογα με την απάντησή σας η εφαρμογή εκτελεί τις αντίστοιχες λειτουργίες.

## Διαχείριση Εξερχόμενων Εγγράφων

Αν θέλετε να δείτε ή να ανακτήσετε κάποιο εξερχόμενο έγγραφο τότε θα πρέπει να επιλέξετε από το κυρίως μενού ή τις συντομεύσεις τη *Διαχείριση Εξερχομένων*. Το παράθυρο που θα ανοίξει έχει την ίδια μορφή με το παράθυρο της *Διαχείρισης Εισερχομένων* και φαίνεται στη εικόνα που ακολουθεί.



Μέσα από το παράθυρο αυτό μπορείτε να κάνετε αναζήτηση με βάση συγκεκριμένα πεδία ώστε να βρείτε εξερχόμενα έγγραφα που αντιστοιχούν στην αναζήτησή σας. Ένα εξερχόμενο έγγραφο γνωρίζετε ότι στέλνεται από τη δική σας εταιρεία έτσι αν κάνετε αναζήτηση με βάση το τμήμα, αυτό που θα ελέγξει η εφαρμογή είναι το τμήμα του αποστολέα. Αν κάνετε αναζήτηση με βάση τον οργανισμό τότε η εφαρμογή ελέγχει τον οργανισμό στον οποίο ανήκει ο παραλήπτης του εγγράφου.





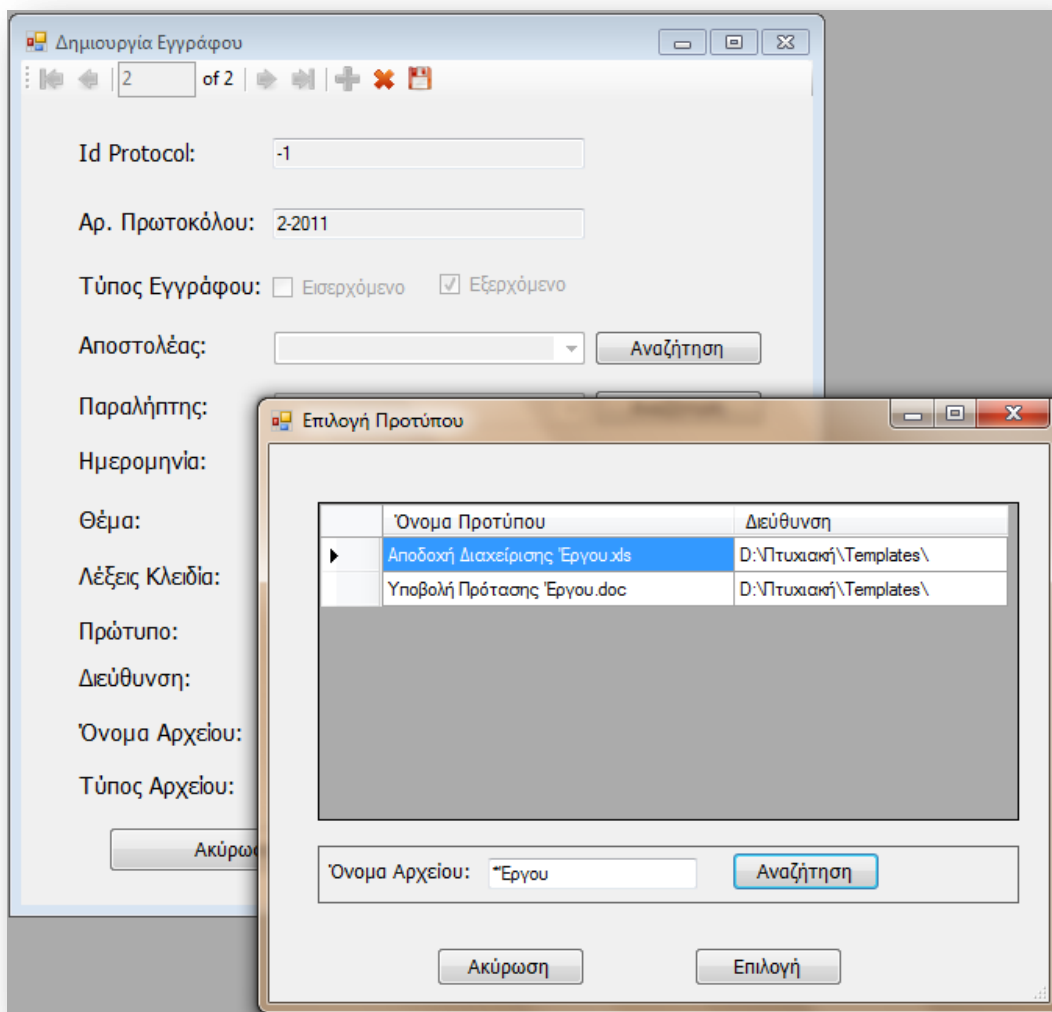
Επιπλέον μέσω από το παράθυρο αυτό μπορείτε να δημιουργήσετε ένα νέο έγγραφο, να επεξεργαστείτε συγκεκριμένα πεδία ενός καταχωρημένου εγγράφου και να ανοίξετε ένα έγγραφο που είναι ήδη καταχωρημένο στην εφαρμογή. Όλα αυτά μπορούν να γίνουν με τη βοήθεια των εργαλείων που υπάρχουν στη μπάρα στο πάνω μέρος του παραθύρου.

## Δημιουργία Εγγράφου

Για να δημιουργήσετε ένα νέο έγγραφο υπάρχουν τρεις τρόποι. Να το δημιουργήσετε επιλέγοντας το κατάλληλο κουμπί από τις συντομεύσεις, πηγαίνοντας από το μενού της εφαρμογής στα *Έγγραφα* και στη συνέχεια στη *Δημιουργία Εγγράφου* και τέλος μέσα από το παράθυρο της *Διαχείρισης Εξερχομένων*.

Όταν ανοίξει το παράθυρο για τη δημιουργία του εγγράφου, το έγγραφο θα πάρει αυτόματα έναν αριθμό πρωτοκόλλου. Εσείς θα πρέπει να επιλέξετε αποστολέα και παραλήπτη καθώς και να συμπληρώσετε όλα τα απαιτούμενα στοιχεία ώστε να μπορέσετε να κάνετε την καταχώρηση.

Κάθε έγγραφο που θέλετε να δημιουργήσετε βασίζεται σε κάποιο πρότυπο γι' αυτό καλό θα είναι να περάσετε τα πρότυπα σας πριν δημιουργήσετε κάποιο έγγραφο. Κατά τη δημιουργία του εγγράφου θα πρέπει να επιλέξετε το πρότυπο με βάση το οποίο θα δημιουργηθεί το έγγραφο σας. Η επιλογή του προτύπου γίνεται με τη βοήθεια ενός άλλου παραθύρου το οποίο εμφανίζει τα πρότυπα που έχετε καταχωρήσει. Μέσα στο παράθυρο αυτό μπορείτε να κάνετε αναζήτηση με βάση τον τίτλο που έχετε δώσει στο πρότυπο. Παράδειγμα αναζήτησης προτύπου υπάρχει στη εικόνα που φαίνεται στη συνέχεια. Όταν επιλέξετε ένα πρότυπο, το πρότυπο αυτό ανοίγει ώστε να κάνετε τις αλλαγές που θέλετε και να το σώσετε. Αρκεί να πατήσετε ένα Save και οι αλλαγές που έχετε κάνει θα καταχωρηθούν στο αρχείο που δημιουργήθηκε.



Αφού ολοκληρώσετε τη εισαγωγή των στοιχείων που χρειάζεται για το έγγραφο, μπορείτε να το καταχωρήσετε πατώντας το κουμπί *Αποθήκευση Εγγράφου* από το κάτω μέρος του παραθύρου ή από τη μπάρα εργαλείων.

Πριν αποθηκευτεί το έγγραφο έχετε τη δυνατότητα να ακυρώσετε τη διαδικασία αν το επιθυμείτε.