



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



Πτυχιακή Εργασία

**Ανάπτυξη εφαρμογής για τη συστηματική ταξινόμηση των
ιχθύων με χρήση γραμμωτού κώδικα**



Εκπόνηση πτυχιακής εργασίας :

Ιωάννης Δήμου

Αρ. Μητρώου : 05/2885

Επιβλέποντες καθηγητές :

Βασίλης Κώστογλου

Γιώργος Μίνος

ΘΕΣΣΑΛΟΝΙΚΗ

Ιούνιος 2013

ΠΡΟΛΟΓΟΣ

Διαβάζοντας κάποιος των τίτλο της συγκεκριμένης πτυχιακής εργασίας, θα διαπιστώσει ότι επιχειρείται το ‘πάντρεμα’ της επιστήμης της βιολογίας με την τεχνολογία και πιο συγκεκριμένα με την τεχνολογία γραμμωτού κώδικα. Στην ουσία, η συγκεκριμένη εφαρμογή κωδικοποιεί ολόκληρη την συστηματική ταξινόμηση των ιχθύων της Μεσογείου και της Μαύρης θάλασσας σε ένα πλήθος δεκαπενταψήφιων αριθμών. Αναλύοντας τον κάθε ένα από αυτούς, ο χρήστης μπορεί να αποκτήσει μία πλήρη άποψη για όλα τα επίπεδα της συστηματικής του ταξινόμησης. Τρόπος με τον οποίο επιτυγχάνεται αυτό θα αναλυθεί παρακάτω.

Ο λόγος που με οδήγησε να ασχοληθώ με την συγκεκριμένη πτυχιακή εργασία, σαν φοιτητής του Τμήματος Πληροφορικής του Α.Τ.Ε.Ι. Θεσσαλονίκης που είμαι, είναι αφενός ότι το μεγαλύτερο της μέρος είχε να κάνει με τον προγραμματισμό για την ανάπτυξη της απαιτούμενης εφαρμογής, ο οποίος είναι ένας τομέας που με ενδιαφέρει πάρα πολύ και αφετέρου, ότι έχοντας ως χόμπι το ψάρεμα και γενικότερα την ενασχόλησή μου με το υδάτινο περιβάλλον, το θεώρησα μία άριστη ευκαιρία να επεκτείνω τις γνώσεις μου σχετικά με αυτό. Αυτός ο συνδυασμός των δύο παραπάνω παραγόντων βοήθησε στην δημιουργία της συγκεκριμένης χρήσιμης αλλά και επεκτάσιμης εφαρμογής.

Όμως αυτή η εφαρμογή δεν θα μπορούσε να υλοποιηθεί χωρίς την πολύτιμη βοήθεια κάποιον συγκεκριμένων ανθρώπων, οι οποίοι με στήριξαν, με συμβούλεψαν, με βοήθησαν, μου πρόσφεραν την γνώση τους και γενικότερα ήταν δίπλα μου σε όλη την διάρκεια του σχεδιασμού και της δημιουργία της. Αρχικά, θα ήθελα να ευχαριστήσω πάρα πολύ, τόσο τον κ. Βασίλη Κώστογλου, Αναπληρωτή Καθηγητή του Τμήματος Πληροφορικής του Α.Τ.Ε.Ι. Θεσσαλονίκης, όσο και τον Δρ. κ. Γιώργο Μίνο, Επίκουρο Καθηγητή του Τμήματος Τεχνολογίας Αλιείας και Υδατοκαλλιεργειών του Παραρτήματος Ν. Μουδανιών του Α.Τ.Ε.Ι. Θεσσαλονίκης, για την ανάθεση του θέματος, την άριστη καθοδήγηση και οργάνωση, που μου παρείχαν, καθ’ όλη τη διάρκεια εκπόνησης της εργασίας αλλά και για τις ενδιαφέρουσες ιδέες και προτάσεις τους. Ακόμα θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε σε όλη τη διάρκεια των σπουδών μου όλα αυτά τα χρόνια. Τέλος θέλω να ευχαριστήσω τον τέως συμφοιτητή μου και μελλοντικό συνάδελφο Αντώνη Κοροσίδα, την Παναγιώτα Μάτσα και τον Σταύρο Καλλίνωση για τις πολύτιμες συμβουλές τους.

ΠΕΡΙΛΗΨΗ

Το κεντρικό θέμα αυτής της πτυχιακής εργασίας είναι η εφαρμογή που σχεδιάστηκε και υλοποιήθηκε κατά τη διάρκεια εκπόνησης της. Πρόκειται για μία εφαρμογή για ηλεκτρονικούς υπολογιστές, η οποία απευθύνεται κυρίως σε ιχθυολόγους και γενικότερα στην ευρύτερη επιστημονική κοινότητα καθώς περιέχει συσσωρευμένη και οργανωμένη πληροφορία σχετική με το αντικείμενο τους, χωρίς όμως να σημαίνει ότι δημιουργήθηκε αποκλειστικά για αυτούς. Στη συγκεκριμένη εφαρμογή μπορεί ο καθένας να περιηγηθεί και να μάθει πληροφορίες για όλους τους ιχθύες, οι οποίοι υπάρχουν στην περιοχή της Μεσογείου και της Μαύρης θάλασσας, να διαβάσει ενδιαφέρουσες περιγραφές, αλλά και να αποκτήσει άποψη για την μορφολογία τους και την γεωγραφική εξάπλωση τους, με την βοήθεια των σχετικών εικόνων.

Η εφαρμογή προσφέρει, σε όποιον χρήστη αποφασίσει να περιηγηθεί σε αυτή δυνατότητες αναζήτησης καθώς και τη δυνατότητα προσθήκης νέων εγγραφών με αυτόματη δημιουργία και αποθήκευση των αντίστοιχων barcode. Για την αποθήκευση μίας νέας εγγραφής απαιτείται μόνο η γνώση του ακριβώς προηγούμενου επιπέδου της συστηματικής ταξινόμησης, το οποίο επίπεδο είναι η οικογένεια. Αυτό οφείλεται στον σχεδιασμό της βάσης και στην καινοτομία την οποία κρύβει. Μέσα σε σχεσιακή βάση δεδομένων δημιουργήθηκε μία πλήρης δένδροειδή μορφή, πράγμα που απο μόνο του αποτελεί καινοτομία. Με αυτόν τον τρόπο επιτεύχθηκε η μεγάλη μείωση των αρχικών πινάκων από 8 σε 2 και η ενγέννη βελτιστοποίηση της εφαρμογής.

Ο γραμμωτός κώδικας που δημιουργείται δεν αντιπροσωπεύει και δεν συγκρίνεται σε καμία περίπτωση με τον πραγματικό γραμμωτό κώδικα που υπάρχει στο εμπόριο, καθώς εκεί υπάρχουν και οι αντίστοιχες τεχνικές ελέγχου για τον ασπρόμαυρο κώδικα αλλά και τα προεπιλεγμένα πεδία που υπάρχουν με σαφή προσδιορισμό. Παρόλα αυτά όμως, είναι βασισμένος στην ενγέννη λογική της κωδικοποίησης σε ψηφία μιας συγκεκριμένης πληροφορίας, τα οποία ψηφία με την βοήθεια κατάλληλων μεθόδων μετατρέπονται σε ασπρόμαυρο κώδικα.

Τέλος, η εφαρμογή προσφέρει μεγάλο εύρος αναβαθμίσεων κυρίως με τη δημιουργία αντίστοιχων βάσεων για όλα τα υπόλοιπα FAO, αλλά και τροποποίηση του ήδη υπάρχων barcode σε QRcode. Εκτός όμως από τα παραπάνω ανοίγει τον δρόμο για μελλοντική εμπορική εφαρμογή, η οποία θα μπορούσε να κρατήσει, να βασιστεί και να εξελίξει πολλά της μέρη σε επίπεδο αλγορίθμου και σχεδίασης.

Στις παρακάτω σελίδες αυτής της πτυχιακής εργασίας παρουσιάζεται τόσο η ανάλυση της παραπάνω εφαρμογής, όσο και ενότητες για τη μεθοδολογία ανάπτυξης της, για τις κλειδες προσδιορισμού και τις υπάρχουσες κλειδες προσδιορισμού ιχθύων, αλλά και μια συμπερασματική συζήτηση.

ΕΥΡΕΤΗΡΙΟ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ.....	6
2	ΣΥΣΤΗΜΑΤΙΚΗ ΙΧΘΥΩΝ	7
2.1	Αρχές Συστηματικής – Ονοματολογία	7
2.2	Ταξινόμηση Ιχθύων	8
3	ΓΡΑΜΜΩΤΟΙ ΚΩΔΙΚΕΣ.....	10
3.1	Σχετικά με τους Γραμμωτούς Κώδικες	11
3.2	Είδη Γραμμωτών Κωδίκων	12
3.2.1	Συμβολογίες EAN-13, EAN-8, UPC-A και UPC-E	12
3.2.2	Συμβολογία ITF - Interleaved Two of Five (ITF-14)	13
3.2.3	Συμβολογία EAN-128	14
3.2.4	Συμβολογία PDF 417.....	15
3.3	Λειτουργία Τεχνολογίας Barcode	16
3.4	Barcode Scanners	17
3.4.1	Ενσύρματοι και ασύρματοι laser barcode scanners χειρός	17
3.4.2	Omni direction barcode scanners	18
3.4.3	Βιομηχανικά (Industrial) scanners	18
3.5	Πλεονεκτήματα - Μειονεκτήματα	19
3.5.1	Πλεονεκτήματα γραμμωτών κωδίκων	19
3.5.2	Μειονεκτήματα γραμμωτών κωδίκων	20
4	ΕΡΓΑΛΕΙΑ - ΜΕΘΟΔΟΛΟΓΙΑ.....	21
4.1	Γλώσσα Προγραμματισμού	21
4.2	Βάση Δεδομένων	22
4.2.1	Σχετικά με την MySQL	22
4.2.2	Ανάλυση της βάσης δεδομένων	22
4.3	Πηγαίος κώδικας – Μεθοδολογία	25
4.3.1	Φόρμες Περιγραφής.....	26
4.3.2	Φόρμες Επιλογών	26
4.3.3	Φόρμες Βασικών Λειτουργιών.....	28
4.3.4	Φόρμες Γενικών Λειτουργιών	38
4.4	Εργαλεία Ανάπτυξης	40
4.5	Πηγές δεδομένων	42
5	ΠΕΡΙΓΡΑΦΗ - ΑΝΑΛΥΣΗ	43
6	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	57
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	59
	ΠΑΡΑΡΤΗΜΑ.....	61
	ΠΑΡΑΡΤΗΜΑ Α	61

ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΦΟΡΜΑΣ MENUFORM.VB.....	61
ΠΑΡΑΡΤΗΜΑ Β	62
ΚΩΔΙΚΑΣ BUTTON1_CLICK	62
ΠΑΡΑΡΤΗΜΑ Γ.....	64
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ PICTUREBOX1_CLICK, PICTUREBOX2_CLICK.....	64
ΚΑΙ BTNSAVESPECIES_CLICK	64
ΠΑΡΑΡΤΗΜΑ Δ	65
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ OPENFILEDIALOG1_FILEOK, OPENFILEDIALOG2_FILEOK ΚΑΙ ADDFORM_LOAD	65
ΠΑΡΑΡΤΗΜΑ Ε	66
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΤΗΣ ΜΕΘΟΔΟΥ BTNSEARCH_CLICK	66
ΠΑΡΑΡΤΗΜΑ ΣΤ	67
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ RADIOBUTTON_1PAINT ΚΑΙ.....	67
BTNSAVESPECIES_CLICK.....	67
ΠΑΡΑΡΤΗΜΑ Ζ.....	68
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΟΥ BARCODESEACH.VB	68
ΠΑΡΑΡΤΗΜΑ Η	72
ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΟΥ BTNSEACH_CLICK.....	72

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 - Συμβολογίες EAN-13, EAN-8, UPC-A και UPC-E	13
Εικόνα 2 - Συμβολογία ITF - Interleaved Two of Five (ITF-14).....	14
Εικόνα 3 - Συμβολογία EAN – 128	15
Εικόνα 4 – PDF.....	16
Εικόνα 5 - Ασύρματοι laser barcode scanners χειρός	18
Εικόνα 6 - Omni direction barcode scanners.....	18
Εικόνα 7 - Βιομηχανικό scanner που χρησιμοποιείται στον τομέα της αυτοβιομηχανίας.....	19
Εικόνα 8 - Πίνακες species dbtree1	23
Εικόνα 9 - Πίνακας επιπέδων συστηματικής.....	23
Εικόνα 10 - Περιεχόμενα του πίνακα dbtree1	24
Εικόνα 11 - Περιεχόμενα του πίνακα dbtree1	25
Εικόνα 12 - Γραφικό περιβάλλον της φόρμας επιλογών Menuform.vb.....	27
Εικόνα 13 - Περιβάλλον της φόρμας επιλογών Menuform2.vb	28
Εικόνα 14– Παράθυρο εισαγωγής στην εφαρμογή	43
Εικόνα 15 – Πεδίο επιλογών FAO.....	44
Εικόνα 16 – Ενεργοποίηση στο κουμπί ‘Είσοδος’ μετά την επιλογή του FAO	44
Εικόνα 17 - Φόρμα εισαγωγής στοιχείων	45
Εικόνα 18 – Επιλογές περιήγησης για τον απλό χρήστη.....	45
Εικόνα 19 – Επιλογές περιήγησης για τον διαχειριστή	46
Εικόνα 20 – Εισαγωγή στην εφαρμογή από γραμμή επιλογών.....	46
Εικόνα 21 – Περιβάλλον εφαρμογής για ‘Αναζήτηση με barcode’	47
Εικόνα 22 – Επιλογή είδους με βάση το barcode και Αναζήτηση.....	47
Εικόνα 23 – Παράδειγμα εφαρμογής με barcode 111010300006022 για αναζήτηση με βάση το barcode	49
Εικόνα 24 – Περιβάλλον εφαρμογής Εκτύπωση Στοιχείων.....	50
Εικόνα 25 - Περιβάλλον εφαρμογής Εκτύπωση Καρτέλας Ιχθύος.....	50
Εικόνα 26 - Περιβάλλον εφαρμογής για ‘Αναζήτηση με βάση το όνομα’	51
Εικόνα 27 – Προτάσεις επιλογών για το κριτήριο Greek Common Name (κοινό ελληνικό όνομα)	52
Εικόνα 28 - Παράδειγμα εφαρμογής με κριτήριο Greek Common Name (κοινό ελληνικό όνομα) και όνομα αλεπόσκυλος για αναζήτηση με βάση το όνομα	52
Εικόνα 29 - Παράδειγμα εφαρμογής με κριτήριο Scientific Name (επιστημονικό όνομα).....	53
Εικόνα 30 – Περιβάλλον εφαρμογής Προσθήκη-Επεξεργασία.....	53

Εικόνα 31 – Περιβάλλον εφαρμογής για την προσθήκη νέου είδους	54
Εικόνα 32 – Δημιουργία barcode μόλις δοθεί το όνομα της οικογένειας που ανήκει (Family Name)	55
Εικόνα 33 – Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους	56
Εικόνα 34 – Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους σύμφωνα με το barcode	56

1 ΕΙΣΑΓΩΓΗ

Δύο ήταν τα βασικά εργαλεία που χρησιμοποιήθηκαν για να αναπτυχθεί η εφαρμογή της Συστηματικής Ταξινόμησης Ιχθύων μέσω γραμμωτού κώδικα. Το ένα αναφορικά με τον προγραμματισμό της εφαρμογής και το άλλο με την διαχείριση της βάσης δεδομένων. Επιπλέον για την παραγωγή της εικόνας του γραμμωτού κώδικα χρησιμοποιήθηκε μία δωρεάν βιβλιοθήκη, με κλάσεις γραμμένες σε γλώσσα προγραμματισμού vb.net, η οποία ενσωματώθηκε στην υπάρχουσα εφαρμογή.

Το εργαλείο προγραμματισμού της εφαρμογής που χρησιμοποιήθηκε είναι το Microsoft Visual Studio Ultimate, ενώ η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι η “ vd.net “, η οποία είναι μια απεικονιστική, οδηγούμενη από συμβάντα αντικειμενοστραφής γλώσσα, στην οποία τα προγράμματα δημιουργούνται χρησιμοποιώντας ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment - IDE).

Για την διαχείριση της βάσης δεδομένων της εφαρμογής χρησιμοποιήθηκε το Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS) MySQL που σήμερα αποτελεί ιδιοκτησία της εταιρίας Oracle Corporation και διαθέτει τον πηγαίο της κώδικα ελεύθερο μέσω της GNU General Public License. Προτιμήθηκε η MySQL καθώς είναι ευρέως διαδεδομένη, με πολλές εφαρμογές σε εφαρμογές ελεύθερου λογισμικού (Joomla, WordPress) και είναι δημοφιλείς βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες όπου χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως Flickr, YouTube, Wikipedia, Google, Facebook, Twitter (Βικιπαίδεια 2013).

2 ΣΥΣΤΗΜΑΤΙΚΗ ΙΧΘΥΩΝ

2.1 Αρχές Συστηματικής – Ονοματολογία

Η συστηματική αποτελεί τη μελέτη των εξελικτικών σχέσεων μεταξύ των οργανισμών, ενώ, σύμφωνα με τον Σουηδό Βοτανολόγο Κάρολο Λινναίο (Carolus Linnaeus), η ταξινομία παράγει ένα τυποποιημένο σύστημα για την ονομασία και την ταξινόμηση των οργανισμών (ειδών) η οποία αντικατοπτρίζει τις εξελικτικές τους σχέσεις.

Η Ταξινομία ή Ταξινομική είναι μέρος της Συστηματικής που έχει δύο ευδιάκριτους στόχους. Ο πρώτος είναι να κατατάξει τους οργανισμούς σε είδη και ο δεύτερος να προσδιορίσει τις εξελικτικές τους σχέσεις και τα πρότυπα των εξελικτικών αλλαγών. Αντίστοιχα, οι συστηματικοί ζωολόγοι, οι οποίοι ασχολούνται με οτιδήποτε είναι γνωστό για τα ζώα και θα βοηθήσει στην καλύτερη κατανόηση των εξελικτικών σχέσεων των ζώων, έχουν σαν στόχους, α) την ανακάλυψη όλων των ειδών των ζώων, β) την ανοικοδόμηση των εξελικτικών τους σχέσεων και γ) την ταξινόμησή τους βάσει των εξελικτικών τους σχέσεων.

Η μελέτη της Ταξινομίας προϋπάρχει της Εξελικτικής με αποτέλεσμα πολλές ταξινομικές πρακτικές να αποτελούν υπολείμματα μιας προ-εξελικτικής άποψης του κόσμου. Για το λόγο αυτό η αναπροσαρμογή του ταξινομικού συστήματος ώστε να συμπεριλάβει την εξέλιξη δημιούργησε πολλά προβλήματα και αντιθέσεις.

Ο Κάρολος Λινναίος στα μέσα του 18ου αιώνα δημοσίευσε τη 10η έκδοση του Systema Naturae (1758), το οποίο δημιούργησε ένα σύστημα από κανόνες επιστημονικής ονοματολογίας, που εμπλουτίστηκαν και απέκτησαν παγκοσμιότητα από τις αρχές του αιώνα μας με τη θέσπιση του Διεθνούς Κώδικα Επιστημονικής Ονοματολογίας των Ζώων, ο οποίος είναι ένα σύστημα επικοινωνίας ανάμεσα στους επιστήμονες όλου του κόσμου, με σκοπό την εξασφάλιση της σταθερότητας, της παγκοσμιότητας και της μοναδικότητας των ονομάτων των ζώων.

Σε αυτό του το σύγγραμμα, ο Λινναίος τα επίπεδα της ταξινομικής ιεραρχίας τα ονομάζει **ταξινομικές βαθμίδες**, οι οποίες είναι το **Βασίλειο** (Kingdom), το **Φύλο** (Phylum), το **Υποφύλο** (Subphylum), η **Υπερομοταξία** (Superclass), η **Ομοταξία** (Class), η **Υφομοταξία** (Subclass), η **Υπέρταξη** (Superorder), η **Τάξη** (Order), η **Υπόταξη** (Suborder), η **Υπεροικογένεια** (Superfamily), η **Οικογένεια** (Family), η **Υποοικογένεια** (Subfamily), η **Ομοιογένεια** (Tribe), το **Γένος** (Genus), το **Υπογένος** (Subgenus), το **Είδος** (Species) και το **Υποείδος** (Subspecies).

Με αυτό του Ιεραρχικό σύστημα ταξινόμησης, ο Λινναίος στοχεύει στη διευθέτηση των οργανισμών σε μια ανιούσα σειρά ομάδων συνεχώς αυξανόμενης περιεκτικότητας, ενώ παρράλλαλη καθιερώνει την διώνυμη ονοματολογία για τα φυτά και τα ζώα, που είναι γνωστή ως «Διώνυμη Ονοματολογία

του Λινναίου». Σύμφωνα με αυτή, το όνομα του κάθε οργανισμού αποτελείται από δύο όρους. Ο πρώτος όρος που αρχίζει πάντοτε με κεφαλαίο γράμμα υποδηλώνει το γένος και ο δεύτερος αρχίζει με πεζό και υποδηλώνει το είδος.

Όπως για τους περισσότερους διάσημους συστηματικούς ερευνητές εκείνης της εποχής, έτσι και για τον Λινναίο πηγή έμπνευσης για την ονομασία των ιχθύων ήταν η ελληνική γλώσσα. Έτσι, είτε υιοθετούσαν αυτούσια αρχαία ελληνικά, είτε δημιουργούσαν καινούργια από την ελληνική γλώσσα, με παραγωγή ή σύνθεση, και τους έδιναν λατινική μορφή, γιατί γλώσσα της επιστήμης στην εποχή τους ήταν η λατινική.

Στην επιστήμη της βιολογίας ο όρος είδος χαρακτηρίζει μία από τις βασικές μονάδες βιοποικιλότητας, δηλαδή ένα σύνολο οργανισμών, ζώων, φυτών ή μικροοργανισμών που μοιάζουν τόσο ώστε να μπορούν αυθόρμητα να προσδιορίζονται και να καταχωρούνται με το ίδιο όνομα. Το είδος αποτελεί την πρώτη βαθμίδα ταξινόμησης των εμβίων όντων. Ένα είδος ή και περισσότερα προσδιορίζουν ένα συγκεκριμένο ταξινομικό γένος (Βικιπαίδεια 2013).

2.2 Ταξινόμηση Ιχθύων

Η συστηματική ταξινόμηση είναι η θεωρία και η πρακτική που χρησιμοποιούν οι βιολόγοι για την κατάταξη των εμβίων όντων. Η χρησιμότητά της έγκειται στην ανάγκη διαχωρισμού των ζωντανών οργανισμών που συναντάμε σε τεράστια ποικιλία στο φυσικό κόσμο: από πελώριες φάλαινες έως και μικροσκοπικά βακτήρια. Η εφαρμογή αυτή της ταξινόμησης προσφέρει μια ονοματολογία η οποία χρησιμοποιείται διεθνώς. Αποτελεί μέρος της ευρύτερης επιστήμης της Συστηματικής.(Wikipedia).

Ο πρώτος που συγκέντρωσε όλες τις σχετικές γνώσεις της εποχής του και πλούτισε αυτές με τις παρατηρήσεις του συγκεντρώνοντας όλες κατά επιστημονικά σύνολα, ήταν ο Αριστοτέλης, ο οποίος διέκρινε τα ψάρια από τα θηλαστικά από την απουσία θηλών των πρώτων σε σύγκριση προς τα δεύτερα. Μετά τον Αριστοτέλη, στη λίτσα των ερευνητών που ασχολούνται με τη συστηματική, προσθέτονται οι Γάλλοι Rondelet και Belon και ο Ιταλός Salviani(16^ο αιώνα). Στα μέσα του 18^{ου} αιώνα, ο Σουηδός βοτανολόγος Κάρολος Λινναίος (1707-1778) ταξινόμησε κατά σύστημα και τα τρία συστήματα της φύσης, δηλαδή Φυτά(Regnum vegetabile), Ζώα (Regnum animale) και Ορυκτά (Regnum lapideum) εισάγοντας την διπλή ονομασία.

Τα κριτήρια της κατάταξης των οργανισμών είναι μορφολογικά. Οι απαρτιζόμενες κύριες ομάδες υποδιαιρούνται σε άλλες μικρότερες και ακόμη μικρότερες κ.ο.κ. Οι βασικές βαθμίδες της ταξινομικής ιεραρχίας είναι οι εξής: το Βασίλειο, η Συνομοταξία, η Ομοταξία, η Τάξη, η Οικογένεια, το Γένος. και το Είδος. Ως βάση της ιεράρχησης λαμβάνεται το είδος, τα όμοια μεταξύ τους είδη

συναποτελούν γένος, όμοια γένη τις οικογένειες και αυτές τις τάξεις, ομοταξίες, κ.λπ. Έτσι, με το έργο του Λινναίου αλλά και αλλά και έργα άλλων αξιόλογων ερευνητών του 18^{ου} αιώνα, όπως οι Artedi, Bloch, Lacepede, Cuvier, Valenciennes Agassiz, Regan, τα προβλήματα της κατάταξης αρχίζουν να επιλύονται.

Μέχρι τον 18^ο αιώνα οι ερευνητές οργάνωναν τα συστήματα ταξινόμησης των ιχθύων βασιζόμενοι στις σχέσεις συγγένειας που είχαν οι ιχθύες μεταξύ τους. Όμως, τα νέα συστήματα ταξινόμησης βασίζονται στις εξελεκτικές συσχετίσεις των ψαριών με γνώμονα τους κοινούς χαρακτήρες κατασκευής τους. Παρόλα αυτά, οι αλλαγές και οι βελτιώσεις δεν σταματούν, αφού η γνώση των περισσότερων ψαριών είναι μακριά από το να είναι ολοκληρωμένη, αλλά και στο γεγονός ότι οι συστηματικές βαθμίδες, που βρίσκονται πάνω από το επίπεδο του είδους, τείνουν να είναι κάπως αυθαίρετες.

Τελικά, αυτές οι αλλαγές οδηγούν σε νέα συστήματα, τα οποία μπορεί να έχουν τελικά λίγη ομοιότητα με τα παλαιότερα στα οποία βασίζονται. Τα περισσότερα πρόσφατα σχήματα ταξινόμησης βασίζονται κατά μεγάλο μέρος στο σχήμα ταξινόμησης κατά Berg (1940). Το πιο πρόσφατο σύστημα ταξινόμησης των ιχθύων είναι το εξής (**Nelson 2006**), στο οποίο βασίστηκε και η μεθοδολογία της συγκεκριμένης εφαρμογής. Συμφωνα με αυτά οι ιχθείς χωρίζονται σε τρεις υπερκλάσεις τα **Πετρομυζινόμορφα**, τα **Μυξινόμορφα** και τα **Γναθόστομα**.

❖ **Υπερκλάση: Πετρομυζινόμορφα (Petromyzontomorphi).**

Κλάση (Ομοταξία) : Πετρομυζόντιδα (Petromyzontida)

❖ **Υπερκλάση: Μυξινόμορφα (Myxiniomorphi).**

Κλάση (Ομοταξία) : Μυξίνοι (Myxini)

❖ **Υπερκλάση: Γναθόστομα (Gnathostomata).**

Κλάση (Ομοταξία) : Χονδριχθύες (Chondrichthyes)

Κλάση (Ομοταξία) : Ακτινοπτερύγιοι (Actinopterygii)

Κλάση (Ομοταξία): Σαρκοπτερύγιοι (Sarcopterygii)

Οι πιο πρωτόγονες είναι τα **Πετρομυζινόμορφα** και τα **Μυξινόμορφα**, τα οποία είναι ιχθύες χωρίς σιαγόνες (άγανθα), που έφτασαν σε στάδιο ακμής πριν από 35 μέχρι 500 εκατομ. χρόνια και αντιπροσωπεύονται στην σημερινή πανίδα των ψαριών με 50 περίπου είδη των οικογενειών Myxiniidae και Petromyzontidae. Μοναδικός εκπρόσωπος στη χώρα μας είναι μόνο ένα είδος, το *Petromyzon marinus* L., 1758.

Τα **Γναθόστομα** είναι η υπέρκλαση όλων των ιχθύων που έχουν γνάθο και χωρίζεται σε τρεις κλάσεις, οι οποίες είναι οι Χονδριχθύες (Chondrichthyes), οι Ακτινοπτερύγιοι (Actinopterygii) και οι Σαρκοπτερύγιοι (Sarcopterygii).

Οι **Χονδριχθύες** (Chondrichthyes) χωρίζονται σε δύο ευδιάκριτες υποκλάσεις, τους Ελασμοβράγχιους (Elasmobranchii) και στους Ολοκέφαλους (Holocerhali). Οι πρώτοι διακρίνονται σε δύο άλλες ευδιάκριτες κατηγορίες (Υπερτάξεις), οι οποίες είναι οι Πλευροτριματικοί (Γαλεόμορφοι) (καρχαρίες, σκυλάκια) και οι Υποτριματικοί (ράγιες, βάτοι) που περιλαμβάνουν τις χίμαιρες.

Οι **Ακτινοπτερύγιοι** (Actinopterygii),) χωρίζονται σε δύο ευδιάκριτες υποκλάσεις, τους Χονδροστέους (στούριόνια) και τους Νεοπτερύγιους (οι περισσότεροι αρτίγονοι ιχθύες).

Οι **Σαρκοπτερύγιοι** (Sarcopterygii) χωρίζονται σε δύο ευδιάκριτες υποκλάσεις, τους Κροσσοπτερύγιους (Coelacanthiformes), στους οποίους περιλαμβάνονται οι κοιλάκανθοι και στους Δίπνευστους (Dipnotetrapodomorpha) που περιλαμβάνονται οι δίπνευστοι Χονδριχθύες. (Μίνος, 2010)

3 ΓΡΑΜΜΩΤΟΙ ΚΩΔΙΚΕΣ

Αυτή η ενότητα του κειμένου, περιέχει πληροφορίες που αφορούν τους γραμμωτούς κώδικες. Υπάρχουν αναφορές τόσο για την εξέλιξή τους, όσο και για την ακριβή λειτουργία τους στο εμπόριο.

Ακόμα, αναλύονται τα βασικότερα είδη κώδικα που υπάρχουν στην αγορά, αλλά και στα μέσα, τύπου scanners, τα οποία απαιτούνται για την εν γένη λειτουργία τους. Τέλος γίνεται εκτενής αναφορά σε όλα τα πλεονεκτήματα και οφέλη που έχουν, αλλά όπως συμβαίνει πάντα και στα λίγα πλην όμως υπαρκτά μειονεκτήματα, τα οποία έχουν.

3.1 Σχετικά με τους Γραμμωτούς Κώδικες

Όπως αναφέρθηκε και στην εισαγωγή ο γραμμωτός ή γραμμικός ή ραβδωτός κώδικας ή ραβδοκώδικας, είναι μία από τις πολλές εφαρμογές που ανήκουν στην κατηγορία των AIDC (Automatic Identification and Data Capture ή κοινώς Αυτόματη Αναγνώριση Στοιχείων Κτήση Δεδομένων), με τα οποία επιτυγχάνεται η εύκολη και γρήγορη λήψη και αποθήκευση της πληροφορίας την στιγμή που αυτή δημιουργείται.

Ο γραμμωτός κώδικας (barcode) εμφανίστηκε την δεκαετία του '60 στις Ηνωμένες Πολιτείες Αμερικής (Η.Π.Α.) και εφαρμόστηκε για να καλύψει τις ανάγκες των Super Market με στόχο τη γρήγορη και εύκολη διακίνηση των αγαθών από τα ταμεία. Ως γνωστόν, στην προ barcode εποχή, οι ταμίες ήταν αναγκασμένοι να πληκτρολογούν στην ταμειακή μηχανή την τιμή του κάθε προϊόντος που επρόκειτο να αγοραστεί από τον πελάτη, με αποτέλεσμα τη δημιουργία ουρών, σφαλμάτων και τη δαπάνη πολύτιμου χρόνου.

Το πρώτο σύστημα barcode εκείνης της όχι και τόσο μακρινής εποχής περιοριζόταν στο "χτύπημα" των προϊόντων και δεν επέτρεπε λειτουργίες όπως η αυτόματη ενημέρωση των αποθεμάτων κάθε φορά που ένα προϊόν αγοραζόταν. Στη συνέχεια, η σχετική τεχνολογία αναβαθμίστηκε και ξέφυγε από τα στενά όρια της ταμειακής καταγραφής των εξερχόμενων προϊόντων, καθώς επεκτάθηκε στην καταγραφή και των εισερχόμενων αγαθών, στην απογραφή, στον έλεγχο των αποθεμάτων κ.λ.π.. Η Ελλάδα εισήλθε στον κόσμο του barcode στα τέλη της δεκαετίας του '80, γεγονός που συνδυάστηκε με τον πολλαπλασιασμό των πολυκαταστημάτων και σουπερ μάρκετ.

Η βασική ιδέα που διέπει τη συγκεκριμένη τεχνολογία θα μπορούσε να συνοψιστεί στο εξής: Τοποθετείται μια σειρά πληροφοριών σε ένα προϊόν με τρόπο πού ειδικά μηχανήματα μπορούν να τη "διαβάσουν" αυτόματα και να τη μεταβιβάσουν σε κεντρικό υπολογιστικό σύστημα, ώστε το προϊόν να είναι αναγνωρίσιμο. Η μεταφορά των δεδομένων από τα ειδικά μηχανήματα ανάγνωσης (scanners, ανιχνευτές κ.λ.π.) σε υπολογιστή, μπορεί να γίνει είτε ενσύρματα είτε ασύρματα.

Ο δεύτερος τρόπος, που κερδίζει συνεχώς έδαφος και αποτελεί ό,τι πιο εξελιγμένο, υποστηρίζεται από ασύρματα δίκτυα τοπικής εμβέλειας (Wireless Local Area Networks - WLAN) μέσα στα οποία μπορεί να γίνει, χωρίς καλώδια, μετάδοση δεδομένων από ένα φορητό τερματικό σε έναν υπολογιστή λ.χ., και από εκεί οπουδήποτε, ακόμα και στο Internet.

Σήμερα οι barcodes είναι παντού, υποκαθιστούν τη χειρόγραφη εισαγωγή ή πληκτρολόγηση της πληροφορίας σε κάποιο σύστημα και ταυτόχρονα "απογειώνουν" τα δεδομένα σε ψηφιακές λεωφόρους υψηλής ταχύτητας και αξιόπιστης διαχείρισης.

Οι Γραμμωτοί κώδικες αποτελούν κλάδο του γενικότερου τομέα τεχνολογιών Αυτόματης Συλλογής Δεδομένων. Είναι τυποποιημένα σύμβολα, τα οποία μπορούν να χαρακτηριστούν σαν ένα είδος αλφαβήτου. Αυτά απεικονίζουν με συνδιασμούς από γραμμές διαφορετικού πλάτους και είδους (σκοτεινές και φωτεινές) μία συγκεκριμένη πληροφορία. Αυτή η πληροφορία διαβάζεται από μηχανήματα ηλεκτρονικής αναγνώρισης.

Παράλληλα, απάντηση στην ανάγκη της χρήσης ενός ακριβούς και κωδικοποιημένου τρόπου αναγνώρισης αντί της χρήσης αλφαριθμητικών χαρακτήρων για τα ονόματα και τις διευθύνσεις, παρέχουν οι EAN – Κωδικοί Αριθμοί Θέσης, που είναι ένας αριθμητικός κωδικός, ο οποίος προσδιορίζει μια οποιαδήποτε νομική, λειτουργική ή φυσική οντότητα σε μία επιχείρηση ή Οργανισμό. Οι αριθμοί αυτοί χρησιμοποιούνται ευρέως και σε καθημερινή βάση από χιλιάδες εταιρείες που ασχολούνται σε τομείς, όπως χημικά, ηλεκτρονικά, τραπεζικές λειτουργίες, προϊόντα φροντίδας της υγείας, λιανική πώληση και ιδιοκατασκευές.

Η παραγωγή προτύπων διαφανειών, που απαιτείται για την εκτύπωση ενός συμβόλου Γραμμωτού Κώδικα (barcode) είναι εργασία εξειδικευμένη κατά την οποία πρέπει να τηρούνται οι Διεθνείς Προδιαγραφές του συστήματος EAN. Η εκτύπωση του Κωδικού ενός προϊόντος πρέπει ν'αντιμετωπίζεται σαν αναπόσπαστο μέρος του σχεδιασμού της συσκευασίας ή της ετικέτας και όχι εκ των υστέρων.

3.2 Είδη Γραμμωτών Κωδίκων

Η κυριότερη χρήση των γραμμωτών κωδικών είναι η ταυτοποίηση των προϊόντων. Όλα τα προϊόντα έχουν ένα σύμβολο, με ασπρόμαυρες μπάρες και διάφορα σύμβολα ή αριθμούς, το οποίο καταδεικνύει την μοναδικότητά του. Τα σύμβολα αυτά όμως διαφέρουν από προϊόν σε προϊόν και διακρίνονται σε διαφορετικούς τύπους ή καλύτερα διαφορετικές συμβολογίες barcode.

Οι συμβολογίες είναι οι ακόλουθες:

3.2.1 Συμβολογίες EAN-13, EAN-8, UPC-A και UPC-E

Όπως παρατηρείται στην παρακάτω εικόνα (Εικόνα 1 - Συμβολογίες EAN-13, EAN-8, UPC-A και UPC-E) είναι αριθμητικές συμβολογίες, δηλαδή μπορούν να απεικονίσουν μόνο τα ψηφία 0-9. Είναι σταθερού μήκους και το περιεχόμενο ενός συμβόλου EAN-13 είναι πάντα 13 ψηφία, ενώ στα σύμβολα EAN-8, UPC-A και UPC-E το περιεχόμενο είναι 8, 12 και 8 ψηφία αντίστοιχα. Είναι

μονοδιάστατες-γραμμικές συμβολογίες, δηλαδή η "κωδικοποίηση" των χαρακτήρων (ψηφίων) είναι συνάρτηση μόνο του πλάτους των μπαρών, είτε φωτεινών είτε σκοτεινών, και όχι του ύψους των. Κάθε χαρακτήρας-ψηφίο αντιστοιχεί σε 7 στοιχειώδεις ή μοναδιαίες μπάρες.



Εικόνα 1 - Συμβολογίες EAN-13, EAN-8, UPC-A και UPC-E

Ως στοιχειώδη (μοναδιαία) μπάρα ή module, ορίζεται μία μπάρα με το ελάχιστο προδιαγεγραμμένο πλάτος, την οποία μπάρα όμως χρησιμοποιούμε ως "μονάδα μέτρησης" και συμβολίζεται συνήθως με το X. Κάθε χαρακτήρας (ψηφίο) απεικονίζεται με 4 μπάρες, εκ των οποίων 2 "σκοτεινές" και 2 "φωτεινές". Η κάθε μία από αυτές αποτελείται από 1 έως και 4 modules.

Οι συμβολογίες αυτές διαβάζονται κάτω από οποιαδήποτε διεύθυνση (omnidirectionally). Αυτό τις καθιστά ιδανικές για τη σήμανση των καταναλωτικών μονάδων των προϊόντων σε συνδυασμό με τους scanners των ταμείων των Σημείων Λιανικής Πώλησης (Supermarkets κλπ).

Βάσει των διαστασιολογικών προδιαγραφών των συγκεκριμένων συμβόλων, ορίζεται ένα μέγεθος ως "ονομαστικό" ή nominal. Το τελικό μέγεθος ενός συμβόλου μπορεί να κυμαίνεται από 80% έως και 200% του ονομαστικού μεγέθους ανάλογα βέβαια και με την ποιότητα εκτύπωσης. Σε πρωτογενείς συσκευασίες προϊόντων, με πολύ μικρό διαθέσιμο χώρο για την εκτύπωση του γραμμωτού κώδικα, χρησιμοποιείται ο EAN-8 ο οποίος έχει λιγότερα ψηφία.

Βάση των προτύπων του EAN τα ψηφία κωδικοποιούν τις εξής πληροφορίες:

520	0123	45678	3
Κωδικός Χώρας	Κωδικός Εταιρίας	Κωδικός Εταιρίας	Κωδικός Προϊόντος

Στην Ελλάδα την ευθύνη διαχείρισης των κωδικών EAN-13 και EAN-8 έχει το Ελληνικό Κέντρο Σήμανσης Προϊόντων (ΕΛΚΕΣΗΠ).

3.2.2 Συμβολογία ITF - Interleaved Two of Five (ITF-14)

Ο γραμμωτός κώδικας INTERLEAVED 2 of 5 ή ITF 14, τοποθετείται στην δευτερογενή συσκευασία των προϊόντων (κιβώτια, πολυσυσκευασίες κτλ). Μπορεί να κωδικοποιήσει μόνο αριθμούς. Περιέχει την ίδια πληροφορία με τον EAN-13 της πρωτογενούς συσκευασίας και έχει την δυνατότητα να κωδικοποιήσει μεταβλητό πλήθος ψηφίων αλλά πάντα άρτιο. Όπως βλέπουμε και στην παρακάτω

εικόνα (Εικόνα 2 - Συμβολογία ITF - Interleaved Two of Five (ITF-14)), ο γραμμωτός κώδικας ITF είναι μία αριθμητική συμβολογία σταθερού μήκους 14 ψηφίων. Είναι μονοδιάστατη-γραμμική συμβολογία.



Εικόνα 2 - Συμβολογία ITF - Interleaved Two of Five (ITF-14)

Υπάρχουν μόνο δύο πλάτη μπαρών, "φαρδιές" και "στενές". Κάθε ζεύγος ψηφίων απεικονίζεται με 5 "σκοτεινές" και 5 "φωτεινές" μπάρες. Η κάθε ομάδα των 5 μπαρών - "σκοτεινών" ή "φωτεινών" - αποτελείται από 2 "φαρδιές" και 3 "στενές" μπάρες και διαβάζεται αμφίδρομα (bidirectionally). Αυτό είναι κι ένας από τους λόγους για τους οποίους δεν χρησιμοποιείται για τη σήμανση των καταναλωτικών μονάδων των προϊόντων σε συνδυασμό με τους scanners των ταμείων των Σημείων Λιανικής Πώλησης (Supermarkets κλπ). Χρησιμοποιείται όμως για τη σήμανση των χαρτοκιβωτίων σε συνδυασμό με τους φορητούς scanners που κατά κανόνα χρησιμοποιούνται στις αποθήκες. Το τελικό μέγεθος ενός συμβόλου μπορεί να κυμαίνεται από 25% έως και 100% του ονομαστικού μεγέθους ανάλογα βέβαια και με την ποιότητα εκτύπωσης.

3.2.3 Συμβολογία EAN-128

Ο γραμμωτός κώδικας EAN-128 τοποθετείται στις μονάδες μεταφοράς προϊόντων (παλέτες, κιβώτια κτλ.). Μπορεί να κωδικοποιήσει αριθμούς και γράμματα. Χρησιμοποιεί δείκτες εφαρμογής (Application Identifiers) για τον προσδιορισμό της πληροφορίας που κωδικοποιεί.

Χρησιμοποιείται σε εφαρμογές διαχείρισης αποθηκών και διακίνησης. Οι πληροφορίες που ενσωματώνει είναι πολύ μεγάλες (είδος, ημερομηνίες παραγωγής και λήξης, lot number, ποσότητας, κ.τ.λ.).

Όπως φαίνεται και στην παρακάτω εικόνα (Εικόνα 3 - Συμβολογία EAN – 128) είναι αλφαριθμητική συμβολογία και μπορεί να απεικονίσει όλους τους χαρακτήρες του κώδικα ASCII 128. Αποτελεί μια "διάλεκτο" της ιδιαίτερα διαδεδομένης συμβολογίας CODE 128 και διαφέρει από αυτήν στο ότι περιέχει έναν παραπάνω ειδικό χαρακτήρα που ονομάζεται Function Code 1 (FNC1).



Εικόνα 3 - Συμβολολογία EAN – 128

Ο γραμμωτός κώδικας EAN-128 τοποθετείται στις μονάδες μεταφοράς προϊόντων (παλέτες, κιβώτια κτλ.). Μπορεί να κωδικοποιήσει αριθμούς και γράμματα. Χρησιμοποιεί δείκτες εφαρμογής (Application Identifiers) για τον προσδιορισμό της πληροφορίας που κωδικοποιεί. Χρησιμοποιείται σε εφαρμογές διαχείρισης αποθηκών και διακίνησης. Οι πληροφορίες που ενσωματώνει είναι πολύ μεγάλες (είδος, ημερομηνίες παραγωγής και λήξης, lot number, ποσότητας, κ.τ.λ.).

Όπως φαίνεται και στην παραπάνω εικόνα (Εικόνα 3 - Συμβολολογία EAN – 128), είναι αλφαριθμητική συμβολολογία και μπορεί να απεικονίσει όλους τους χαρακτήρες του κώδικα ASCII 128. Αποτελεί μια "διάλεκτο" της ιδιαίτερα διαδεδομένης συμβολολογίας CODE 128 και διαφέρει από αυτήν στο ότι περιέχει έναν παραπάνω ειδικό χαρακτήρα που ονομάζεται Function Code 1 (FNC1). Είναι μεταβλητού μήκους και μπορεί να απεικονίσει μέχρι 48 αλφαριθμητικούς χαρακτήρες ή 96 μόνο αριθμητικούς εκμεταλλευόμενα το χαρακτηριστικό "υψηλής πυκνότητας" το οποίο έχει. Είναι μονοδιάστατη-γραμμική συμβολολογία.

Κάθε χαρακτήρας-ψηφίο αντιστοιχεί σε 13 "στοιχειώδεις/μοναδιαίες μπάρες - modules". Κάθε χαρακτήρας-ψηφίο απεικονίζεται με 6 μπάρες - 3 "σκοτεινές" και 3 "φωτεινές" - η κάθε μία εκ των οποίων αποτελείται από 1 έως και 4 modules και διαβάζεται αμφίδρομα (bidirectionally).

Αυτός είναι κι ένας από τους λόγους για τους οποίους δεν χρησιμοποιείται για τη σήμανση των καταναλωτικών μονάδων των προϊόντων σε συνδυασμό με τους scanners των ταμείων των Σημείων Λιανικής Πώλησης (Supermarkets κλπ). Χρησιμοποιείται όμως για τη σήμανση των μονάδων Logistics σε συνδυασμό με τους φορητούς scanners που κατά κανόνα χρησιμοποιούνται στις αποθήκες. Το τελικό μέγεθος ενός συμβόλου μπορεί να κυμαίνεται από 25% έως και 100% του ονομαστικού μεγέθους ανάλογα βέβαια και με την ποιότητα εκτύπωσης.

3.2.4 Συμβολολογία PDF 417

Όπως φαίνεται και παρακάτω (Εικόνα 4 – PDF 417), ο κώδικας PDF 417 είναι μια δισδιάστατη συμβολολογία με δυνατότητα κωδικοποίησης μεγάλου αριθμού δεδομένων. Είναι συμβολολογία μεταβλητού ύψους και πλάτους και μπορεί να κωδικοποιήσει πληροφορία σε 3 – 90 γραμμές

οριζόντια και 1 – 30 κωδικούς σε κάθε γραμμή. Χρησιμοποιείται κυρίως σε εφαρμογές διακίνησης προϊόντων όπου χρειάζονται μεγάλη κωδικοποιημένη πληροφορία.



Εικόνα 4 – PDF 417

3.3 Λειτουργία Τεχνολογίας Barcode

Τι είναι όμως και πώς λειτουργεί ο γραμμωτός κώδικας;

Ο γραμμωτός κώδικας δεν είναι τίποτα άλλο, παρά η εναλλαγή λευκών και μαύρων γραμμών για την παράσταση αριθμών, ώστε να είναι εύκολα αναγνώσιμοι από τις ηλεκτρονικές μηχανές – υπολογιστές. Το πιο γνωστό σύστημα γραμμωτού κώδικα είναι το E.A.N. 13 (European Article Numbering). Το E.A.N – 13, σημαίνει «Ευρωπαϊκή Αρίθμηση Αντικειμένων» και αποτελείται από 13 ψηφία. Το πρώτο είναι ξεχωριστό ενώ τα υπόλοιπα 12 είναι χωρισμένα στα δύο, ένα αριστερά και ένα δεξιά.

Το πρώτο ψηφίο, γράφεται αριστερά πρώτος από όλους και με αριθμητική μορφή. Τα επτά πρώτα ψηφία, υποδηλώνουν την χώρα προέλευσης και τον κατασκευαστή. Τα πέντε επόμενα, δείχνουν τον τύπο του προϊόντος και το τελευταίο ψηφίο, είναι το ψηφίο ελέγχου με το οποίο ο υπολογιστής ελέγχει αν διάβασε σωστά τον κώδικα. Υπάρχουν και οι βοηθητικοί χαρακτήρες, είναι οι πιο μακριοί χαρακτήρες και βρίσκονται ένας στην αρχή, ένας στην μέση και ένας στο τέλος. Ο μεσαίος χωρίζει τα 12 στοιχεία σε δύο υποπεδία των 6 στοιχείων. Κάθε ψηφίο αποτελείται από μια αλληλουχία άσπρων και μαύρων γραμμών. Η αλληλουχία μαύρου (1) – άσπρου (0) πρέπει να γίνεται τέσσερις φορές. Επίσης κάθε ψηφίο αποτελείται από επτά μονάδες (modules) πλάτους 0.33 χιλιοστών η κάθε μία. Ήλαδη κάθε ψηφίο, έχει πλάτος 2.31 χιλ. = $7 / 0.33$. Κάθε μέρος μαύρου ή άσπρου μπορεί να αποτελείται από παραπάνω από ένα modules, δηλαδή, 1 ψηφίο = $7 \times 0.33 \text{modules} = 4$ εναλλαγές άσπρου μαύρου = 2.31 χιλ. Είναι δυνατό να υπάρχει μεγέθυνση και σμίκρυνση της παράστασης αλλά σε περίπτωση σμίκρυνσης δεν επιτρέπεται το module να είναι κάτω από 0.23 χιλ. Κάθε ψηφίο του κώδικα μπορεί να παραστεί με 3 τρόπους. Οι πρώτοι δύο χρησιμοποιούνται για την κωδικοποίηση στο αριστερό πεδίο ενώ ο τρίτος στο δεξί.

Κατ' αρχάς, στην πιο συνήθη του μορφή συνίσταται από μια διαδοχή μαύρων και λευκών λωρίδων (bars) τυπωμένων πάνω σε κάποιο προϊόν / συσκευασία. Ανάμεσα στις λωρίδες παρεμβάλλονται κενά διαστήματα ποικίλου μεγέθους. Η σχεδίαση, εκτύπωση και απεικόνιση των λωρίδων ακολουθούν συγκεκριμένους κανόνες και κάθε αλληλουχία λωρίδων αντιστοιχεί αμφιμονοσήμαντα

σε κάποιον αριθμό. Αυτό στην πράξη σημαίνει ότι οι αριθμοί μπορούν να συμβολίζονται από διαφορετική κάθε φορά αλληλουχία λωρίδων (άλλοτε μία μαύρη λεπτή λωρίδα μαζί με μία λευκή παχιά λωρίδα μπορούν να αντιστοιχούν στον αριθμό 6, άλλοτε συμβαίνει ο ίδιος συνδυασμός να αντιστοιχεί στον αριθμό 8 κ.ο.κ.).

Ο πιο κοινός τύπος γραμμωτού κώδικα είναι ο EAN (European Article Numbering), ο οποίος αποτελείται από αριθμοσειρά 13 ψηφίων και απαντάται στα περισσότερα καταναλωτικά είδη. Ωστόσο, υπάρχουν και άλλοι τύποι barcode με περισσότερα ή και λιγότερα ψηφία από τα 13 του προτύπου EAN, τα οποία βρίσκουν εφαρμογή σε εξειδικευμένους επιχειρηματικούς κλάδους, όπως λ.χ. στη σιδηροδρομική βιομηχανία. Η ανάγνωση του barcode, συντελείται ως εξής: ο κωδικός EAN μεταφράζεται μέσα σε κλάσματα δευτερολέπτου από κάποιο scanner σε γλώσσα H/Y (δηλ. 0, 1 - δυαδικό σύστημα). Η ανάγνωση του barcode συνίσταται στην αποκωδικοποίηση της ανάκλασης μιας δέσμης ακτίνων laser που "πέφτει" πάνω στην ετικέτα barcode.

Ο βαθμός ανάκλασης είναι μεταβλητός (άρα ξεχωριστός για κάθε προϊόν) γιατί η δέσμη laser συναντά διαδοχικά τις λευκές και μαύρες λωρίδες. Τα scanners αποκωδικοποιούν τη μεταβλητή ανάκλαση (σήμα) και τη μετατρέπουν σε αριθμούς ή γράμματα, τα οποία ταυτίζονται ως προς το περιεχόμενο με τους χαρακτήρες που κωδικοποιήθηκαν με μορφή barcode.

Έτσι, όταν ένα προϊόν περνά από το ταμείο λαμβάνει χώρα η εξής διαδικασία: αναγνώριση, ανάγνωση, μετάφραση και αποθήκευση, που σημαίνεται από το χαρακτηριστικό ήχο ("μπιπ") που ακούγεται κάθε φορά που η ταμίας χρησιμοποιεί το scanner. Τη συγκεκριμένη στιγμή ανιχνεύεται το είδος από τη βάση δεδομένων και διατίθενται για εκτύπωση τα επιμέρους προσδιοριστικά στοιχεία του, όπως η περιγραφή του προϊόντος, η τιμή πώλησής του, η ποσότητα κ.λπ., στοιχεία που καταγράφονται στην ταμειακή μηχανή. Συγχρόνως, αφαιρείται το προϊόν από τη βάση δεδομένων της αποθήκης και των αποθεμάτων.

3.4 Barcode Scanners

Η αναγνώριση ενός προϊόντος, μέσω της αναγνώρισης του barcode είναι ο ταχύτερος, ασφαλέστερος και οικονομικότερος τρόπος για την ανάκληση των στοιχείων του προϊόντος ή της καταχώρησης των πληροφοριών που το συνοδεύουν. Ανάλογα, όμως, με τις συνθήκες κάτω από τις οποίες θα σκαναριστεί το barcode, θα πρέπει να επιλεγεί και ο κατάλληλος barcode scanner.

Έτσι προκύπτουν οι ακόλουθες κατηγορίες :

3.4.1 Ενσύρματοι και ασύρματοι laser barcode scanners χειρός

Οι scanners χειρός είναι η οικονομικότερη λύση, αλλά προϋποθέτει την ύπαρξη εργαζομένου ο οποίος θα τους χρησιμοποιεί. Διαβάζουν συνήθως από μικρή απόσταση το barcode (max 1 – 2

μέτρα). Τέτοιου είδους scanners χρησιμοποιούνται ευρέως είτε σε σημεία λιανικής πώλησης είτε σε δειγματοληπτικούς ελέγχους σχετικά με την παραγωγή, ποιοτικούς ελέγχους κ.α. .

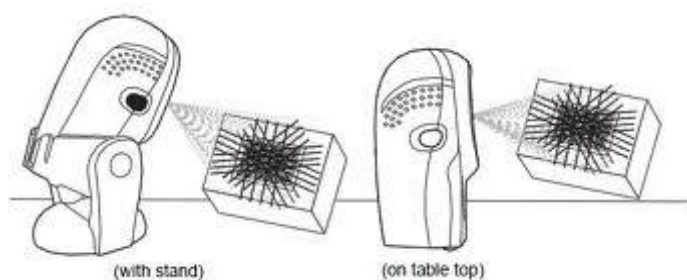
Εκτός όμως από τα παραπάνω βρίσκουν χρήση και σε παραλαβή προϊόντων σε καταστήματα λιανικής πώλησης και σε χώρους στάθμευσης. Στην κατηγορία αυτή ανήκουν τα scanners της παρακάτω εικόνας (Εικόνα 5 - Ασύρματοι laser barcode scanners χειρός).



Εικόνα 5 - Ασύρματοι laser barcode scanners χειρός

3.4.2 Omni direction barcode scanners

Οι omni direction scanners εκπέμπουν δέσμη πολλαπλών laser ακτίνων και επιπλέον μπορούν να είναι σταθερά τοποθετημένοι σε κάποιο σημείο. Τα ανωτέρω επιτρέπουν στον χρήστη την ελεύθερη χρήση και των δύο χεριών του, άρα μεγαλύτερη ταχύτητα στην περάτωση της εργασίας. Συνήθως χρησιμοποιούνται σε ταμεία πολυκαταστημάτων και super market, σε χώρους στάθμευσης και για μεθόδους ιχνηλάτισης. Ακόμα μπορούν να χρησιμοποιηθούν κατά την διάρκεια της παραγωγής για αυτόματη ανάγνωση όταν οι ταχύτητες είναι μικρές και οι συσκευασίες, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 6 - Omni direction barcode scanners).



Εικόνα 6 - Omni direction barcode scanners

3.4.3 Βιομηχανικά (Industrial) scanners

Η κατηγορία των βιομηχανικών scanners προσφέρει την δυνατότητα της αυτόματης και απόλυτα ασφαλούς ανάγνωσης των barcode, σε οποιοδήποτε σημείο της συσκευασίας κι αν βρίσκεται, με οποιαδήποτε ταχύτητα κι αν κινείται το προϊόν.

Οι βιομηχανικοί scanners εκπέμπουν χιλιάδες ακτίνες laser ανά δευτερόλεπτο, διαβάζουν από μεγάλη απόσταση και αποτελούν το πλέον σύγχρονο μέσο αυτόματης ανάγνωσης των barcode, χωρίς την ύπαρξη εργαζομένου στο σημείο του scanning.

Επομένως χρησιμοποιούνται είτε για αυτόματη ανάγνωση και καταγραφή κατά τη διάρκεια της παραγωγής, είτε για συνδυασμό με βιομηχανικούς αυτοματισμούς με στόχο την ταυτόχρονη μέτρηση των διαστάσεων ή την ζύγιση των προϊόντων, είτε για τον συνδυασμό με βιομηχανικούς αυτοματισμούς για την αυτόματη οδήγηση των προϊόντων, σε συγκεκριμένες θέσεις. Ένα παράδειγμα τέτοιου scanner φαίνεται στην παρακάτω εικόνα (Εικόνα 7 - Βιομηχανικό scanner που χρησιμοποιείται στον τομέα της αυτοβιομηχανίας).



Εικόνα 7 - Βιομηχανικό scanner που χρησιμοποιείται στον τομέα της αυτοβιομηχανίας

3.5 Πλεονεκτήματα - Μειονεκτήματα

Όλα τα πράγματα σε όλους τους τομείς έχουν πλεονεκτήματα και μειονεκτήματα. Οι γραμμωτοί κώδικες δεν μπορούσαν να αποτελούν εξέρεση.

3.5.1 Πλεονεκτήματα γραμμωτών κωδίκων

Τα πλεονεκτήματα που έχει ο γραμμωτός κώδικας, είναι αυτά που τον οδήγησαν σε ευρεία χρήση παγκοσμίως. Αυτά είναι :

- Οι γραμμωτοί κώδικες γνωρίζουν μεγάλη αποδοχή, επειδή αποτελούν το απλούστερο, ακριβέστερο και οικονομικότερο μέσο αναγνώρισης αντικειμένων μέσω κατάλληλων συσκευών ανάγνωσης (αναγνώστες ή SCANNERS).
- Η συλλογή δεδομένων με ανάγνωση γραμμωτών κωδικών είναι γρηγορότερη από την πληκτρολόγηση ή καταγραφή των δεδομένων σε χαρτί.
- Με την αντικατάσταση χειροκίνητων συστημάτων με συστήματα barcodes μπορεί να επιτευχθεί αυξημένη παραγωγικότητα και μικρότερο εργατικό κόστος.
- Το βασικότερο πλεονέκτημα τους σε σχέση με άλλες αντίστοιχες τεχνολογίες είναι η χαμηλή πιθανότητα σφάλματος στην εισαγωγή και ανάκτηση δεδομένων. Με την ενσωμάτωση

συστημάτων αυτοελέγχου (π.χ. ψηφία ελέγχου) υπάρχει προστασία από περίπτωση λάθους ανάγνωσης.

- Ο γραμμωτός κώδικας έχει γρήγορη ανάγνωση και η πολύ καλή αποδοτικότητα.
- Η πληροφορία μπορεί να μεταφερθεί στον κεντρικό υπολογιστή και είναι άμεσα διαθέσιμη σε οποιονδήποτε τη χρειάζεται.
- Επιτυγχάνεται η έγκαιρη πληροφόρηση για αποδοτικότερη διαχείριση πόρων, αποθεμάτων, πρώτων υλών κτλ όπως και η ακρίβεια στην ακρίβεια των δεδομένων.

3.5.2 Μειονεκτήματα γραμμωτών κωδίκων

Όπως είπαμε παρά τα πολλά πλεονεκτήματα που παρέχουν, έχουν και αυτοί κάποια μειονεκτήματα. Παρόλο που με την τεχνολογία του barcode, ο κόσμος είναι πλέον εξοικειωμένος, ο γραμμωτός κώδικας έχει τέσσερα βασικά μειονεκτήματα. Αυτά είναι τα ακόλουθα:

- Θα πρέπει να είναι ευδιάκριτο και εύκολα αναγνώσιμο.
- Το barcode δεν μπορεί να γραφεί ή να παραμορφωθεί.
- Δεν μπορούν να γίνουν αλλαγές στα δεδομένα ενός barcode από την στιγμή που έχει τυπωθεί.
- Καταλαμβάνει χώρο στο αντικείμενο που έχει τυπωθεί.

4 ΕΡΓΑΛΕΙΑ - ΜΕΘΟΔΟΛΟΓΙΑ

Σε αυτή την ενότητα, υπάρχει μια εκτενή αναφορά σε όλα τα μέσα που χρησιμοποιήθηκαν για την περαίωση της συγκεκριμένης εφαρμογής. Ειδικότερα, αναλύονται τα εργαλεία που χρησιμοποιήθηκαν, η γλώσσα προγραμματισμού και το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που επιλέχθηκαν, καθώς και οι λόγοι για τους οποίους προτιμήθηκαν.

Επιπλέον, θα αναλυθεί ένα μεγάλο μέρος του πηγαίου κώδικα της εφαρμογής, με την βοήθεια επεξηγηματικών εικόνων και την αναλυτική περιγραφή αυτών. Με αυτόν τον τρόπο, θα γίνει πιο εύκολη η κατανόηση της μεθοδολογία που ακολουθήθηκε και θα βοηθήσει τον αναγνώστη να δημιουργήσει μία πλήρη εικόνα και άποψη για την εφαρμογή.

Τέλος, γίνεται ειδική αναφορά στις πηγές δεδομένων, το οποίο είναι από τα σπουδαιότερα μέρη αυτής της εφαρμογής, εξαιτίας του τεράστιου πλήθους των δεδομένων που έπρεπε να συλλεχθούν και να επεξεργαστούν κατάλληλα, ώστε να επιτευχθεί ένας από τους στόχους της εφαρμογής, ο οποίος ήταν η παροχή ολοκληρωμένης και σωστά δομημένης πληροφορίας με γνώμονα τις αρχές της συστηματικής ταξινόμησης των ιχθύων.

4.1 Γλώσσα Προγραμματισμού

Η γλώσσα προγραμματισμού που επιλέχθηκε για την δημιουργία της εφαρμογής, στα πλαίσια της πτυχιακής εργασίας, είναι η vb.net, η οποία είναι μια απεικονιστική, οδηγούμενη από συμβάντα αντικειμενοστραφής γλώσσα, στην οποία τα προγράμματα δημιουργούνται χρησιμοποιώντας ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment – IDE).

Με το IDE, ο προγραμματιστής μπορεί να σχεδιάσει την διεπιφάνεια, να γράψει τον κώδικα, να εκτελέσει και να διορθώσει το πρόγραμμα σε αρκετά σύντομο χρόνο. Η διαδικασία της γρήγορης ανάπτυξης μιας εφαρμογής αναφέρεται ως Rapid Application Development (RAD). Η Visual Basic είναι η πιο διαδεδομένη γλώσσα RAD που χρησιμοποιείται σήμερα. Επιπλέον, η vb.net είναι συμβατή με το .NET και χρησιμοποιεί το περιβάλλον Visual Studio.NET ως IDE, έχοντας ως βασικές λειτουργίες την οπτική σχεδίαση, την πλήρη βοήθεια, την αποσφαλμάτωση, την μεταγλώττιση, την διαχείριση έργου και τη διανομή της εφαρμογής. Είναι λοιπόν μία γλώσσα που μπορεί να εφαρμοστεί σε οποιοδήποτε μηχάνημα με μόνη προϋπόθεση την ύπαρξη/εγκατάσταση του NET. Γίνεται κατανοητό ότι η γλώσσα προγραμματισμού vb.NET, είναι μία γλώσσα που μπορεί να καλύψει τις απαιτήσεις οποιουδήποτε προγραμματιστή παρέχοντας εύκολη υλοποίηση χωρίς να στερείται σε αποδοτικότητα, ενώ οι πολλές βιβλιοθήκες που υπάρχουν στο internet βοηθούν στην δημιουργία εφαρμογών υψηλών απαιτήσεων.

4.2 Βάση Δεδομένων

Σε αυτή την υποενότητα, θα αναλυθεί το Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS) MySQL, όπως και οι λόγοι για τους οποίους επιλέχθηκε για την δημιουργία της βάσης δεδομένων που χρησιμοποιεί η εφαρμογή. Επιπλέον θα γίνει εκτενή αναφορά στη βάση δεδομένων fish_database, η οποία θα αφορά τους πίνακες και τα πεδία της, καθώς και στις καινοτομίες που αναπτύχθηκαν μέσα σε αυτή.

4.2.1 Σχετικά με την MySQL

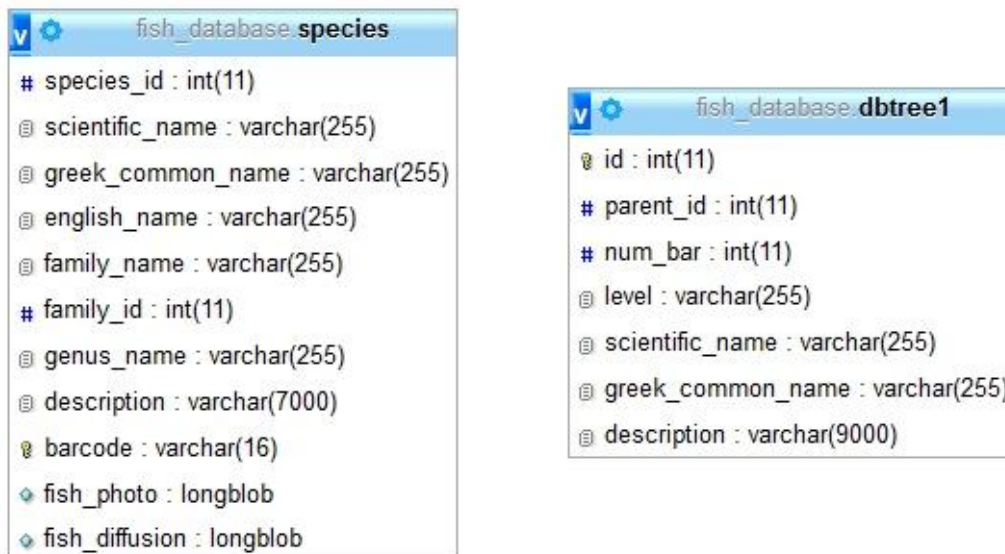
Για την εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, ήταν απαραίτητη και η ύπαρξη μίας βάσης δεδομένων με την οποία θα συνδέεται η εφαρμογή, έτσι ώστε να αντλεί τα απαραίτητα δεδομένα και να τα παρουσιάζει σαν πληροφορίες στον τελικό χρήστη.

Η βάση δεδομένων αποτελεί ένα σημαντικό κομμάτι της συγκεκριμένης εφαρμογής, διότι ο όγκος των δεδομένων είναι αρκετά μεγάλος. Μέσα στην βάση δεδομένων έχουν αποθηκευτεί πληροφορίες για κάθε επίπεδο της συστηματικής κατάταξης των ιχθύων, καθώς και ένας μεγάλος αριθμός εικόνων για κάθε είδος ιχθύος. Χρησιμοποιήθηκε, λοιπόν, το Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS) MySQL (MySQL, Wikipedia). Η MySQL αποτελεί σήμερα ιδιοκτησία της εταιρίας Oracle Corporation και διαθέτει τον πηγαίο της κώδικα ελεύθερο, μέσω του GNU General Public License. Επίσης, είναι ευρέως διαδεδομένη, με πολλές εφαρμογές κυρίως σε Project ελεύθερου λογισμικού, όπως το Joomla και το WordPress, αλλά και σε πολύ γνωστά προϊόντα του Παγκόσμιου Ιστού, όπως η Wikipedia, το Google, το Facebook και πολλά άλλα.

Αποτελεί έτσι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις, ενώ έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι, η οποία στα αγγλικά γράφεται My. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

4.2.2 Ανάλυση της βάσης δεδομένων

Πριν την ανάλυση της βάσης δεδομένων που χρειάστηκε να δημιουργηθεί για την εφαρμογή, κρίζει απαραίτητο να δοθεί προσοχή στους δύο πίνακες της βάσης μας όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 8 - Πίνακες species dbtree1).



Εικόνα 8 - Πίνακες species dbtree1

Παρατηρώντας κάποιος την παραπάνω εικόνα, καταλήγει σε δύο απλά συμπεράσματα – προβληματισμούς. Αυτά είναι :

- Η πολύ απλή δομή της βάσης
- Η μη ορατή σύνδεση μεταξύ των πινάκων της

Στα παραπάνω γίνεται ιδιαίτερη αναφορά γιατί εκεί βασίζονται και οι προγραμματιστικές και μη καινοτομίες της εφαρμογής. Η πολύ απλή δομή που επιτυγχάνεται να έχει η βάση ήταν στόχος και σκοπός στο πλαίσιο της βελτιστοποίησης της στο τελικό αποτέλεσμα.

Με βάση την συστηματική ταξινόμηση που αναλύθηκε σε προηγούμενο κεφάλαιο (2. ΣΥΣΤΗΜΑΤΙΚΗ ΙΧΘΥΩΝ), η συστηματική των ιχθύων θα μπορούσε να αναλυθεί σε 8 επίπεδα (level), όπως παρουσιάζονται στον πίνακα της εικόνας 9 (Πίνακας επιπέδων συστηματικής).

Level1	Superclass
Level2	Class
Level3	Subclass
Level4	Superorder
Level5	Order
Level6	Suborder
Level7	Family
Level8	Species

Εικόνα 9 - Πίνακας επιπέδων συστηματικής

Σε αυτή τη βάση λοιπόν επιχειρήθηκε και επιτεύχθηκε η δημιουργία μίας πλήρους δενδροειδούς δομής μέσα σε μία σχεσιακή βάση δεδομένων, η οποία δεν υποστηρίζει αντικειμενοστρέφια.

Ο πίνακας dbtree1 αποτελείται από 7 'εσωτερικά επίπεδα', τα οποία είναι συνδεδεμένα μεταξύ τους. Αντί λοιπόν να υπάρχουν 8 διαφορετικοί πίνακες συνδεδεμένοι μεταξύ τους, δημιουργήθηκαν μόλις δύο, εκ των οποίων ο ένας (dbtree1) έχει ενσωματωμένη την πληροφορία των 7 πρώτων επιπέδων και είναι συνδεδεμένος προγραμματιστικά με τον πίνακα species. Ο species αποτελεί το 8ο επίπεδο αυτής της συστηματικής ταξινόμησης ενώ ο πίνακας dbtree1 αποτελείται από 7 'εσωτερικά επίπεδα', τα οποία είναι συνδεδεμένα μεταξύ τους.

Αξίζει ακόμα να σημειωθεί ότι η δενδροειδής δομή που φαίνεται να προκύπτει από τον πίνακα των επιπέδων δεν είναι πλήρης, δηλαδή παραδείγματος χάρη δεν είναι υποχρεωτικό μία τάξη (order) να έχει οπωσδήποτε υπέρταξη (superorder) ή υπόταξη (suborder). Για το λόγο αυτό δημιουργήθηκαν 35 επιπλέον εγγραφές, οι οποίες έχουν τον ρόλο 'κενών κόμβων' και μέσω αυτών κάνουμε το μη πλήρες δένδρο πλήρες, αρχής γενομένης από την εγγραφή 268 έως την εγγραφή 293, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 10- Περιεχόμενα του πίνακα dbtree1).

 Edit  Copy  Delete 266	121	145	family	Stomiidae	-
 Edit  Copy  Delete 267	267	146	family	Coelacanthidae	-
 Edit  Copy  Delete 268	22	0	suborder	-	-
 Edit  Copy  Delete 269	22	0	suborder	-	-
 Edit  Copy  Delete 270	22	0	suborder	-	-

Εικόνα 10 - Περιεχόμενα του πίνακα dbtree1

Για να γίνει πιο κατανοητή η εσωτερική λειτουργία του πίνακα dbtree1 θα πρέπει πρώτα να γίνει ανάλυση μερικών πεδίων του κώδικα έτσι και έναν παραλληλισμό με την δομή ενός δένδρου. Αρχικά πεδίο id, το οποίο είναι ένας μοναδικός αριθμός και αποτελεί το κύριο κλειδί του πίνακα. Στη δενδροειδή μορφή, οι 293 εγγραφές αυτού του πεδίου είναι όλοι οι κόμβοι των 7 πρώτων επιπέδων.

Ακολουθώς σαν parent_id ορίζεται το id του ακριβώς μεγαλύτερου επιπέδου. Δηλαδή στη δενδροειδή μορφή μας, αυτό το πεδίο έχει την πληροφορία του 'γονιού' κόμβου. Σαν ρίζα (root) ορίζεται το 0. Σε αυτή 'δείχνουν' οι 3 μοναδικές εγγραφές του 1^{ου} επιπέδου (level-superclass), όπως φαίνεται και στην εικόνα 11 (Περιεχόμενα του πίνακα dbtree1).



	id	parent_id	num_bar	level	scientific_name	greek_common_name
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	0	1	superclass	Gnathostomata	Γναθόστομα
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	0	2	superclass	Petromyzontomorphi	Πετρομυζινόμορφα
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	0	3	superclass	Myxinomorphi	Μυξινόμορφα

Εικόνα 11 - Περιεχόμενα του πίνακα dbtree1

Το επόμενο πεδίο του πίνακα είναι το 'num_bar'. Εδώ η βάση εισάγει μια ακόμη καινοτομία. Η εγγραφή αυτή είναι ο αύξοντα αριθμός του κάθε εσωτερικού επιπέδου level. Στη δένδροειδή μορφή μας είναι η τιμή που περιέχεται στον κάθε κόμβο. Από αυτό τον αριθμό προκύπτει αργότερα και η αλληλουχία των ψηφίων του barcode.

Το τελευταίο πεδίο που χρίζει ανάλυσης στον πίνακα dbtree1 είναι το πεδίο 'level'. Αυτό το επίπεδο χαρακτηρίζει τη θέση της κάθε εγγραφής και στη δένδροειδή μορφή, χαρακτηρίζει τον κάθε κόμβο με το όνομα του επιπέδου, στο οποίο βρίσκεται.

Τώρα όσο αναφορά την ένωση των δύο πινάκων η απάντηση βρίσκεται στον προγραμματισμό της εφαρμογής και επιτυγχάνεται με δύο εμφολευμένα Query. Περισσότερες πληροφορίες θα δοθούν στην παρακάτω υπόενοτητα 4.3, όπου θα αναλυθεί ο πηγαίος κώδικας και η μεθοδολογία.

4.3 Πηγαίος κώδικας – Μεθοδολογία

Σε αυτή την ενότητα υπάρχει εκτενή αναφορά στη μεθοδολογία που χρησιμοποιήθηκε και ανάλυση μεγάλου μέρους του πηγαίου κώδικα της εφαρμογής. Αρχικά, για την υλοποίηση της, χρειάστηκαν να δημιουργηθούν 15 διαφορετικές φόρμες. Για την ευκολότερη ανάλυση αυτών, επιχειρήθηκε και επιτεύχθηκε η κατηγοριοποίηση τους με γνώμονα την λειτουργία τους στην εφαρμογή. Έτσι προκύπτουν οι τέσσερις ακόλουθες κατηγορίες :

- Οι φόρμες περιγραφής, στις οποίες εντάσσονται οι φόρμες **AboutAdd.vb**, **AboutApp.vb**, **AboutBcode.vb**, **AboutBs.vb** και **AboutNs.vb**.
- Οι φόρμες επιλογών, στις οποίες εντάσσονται οι φόρμες **Menuform.vb** και **Menuform2.vb**.
- Οι φόρμες βασικών λειτουργιών, στις οποίες εντάσσονται οι φόρμες **AddForm.vb**, **NameSearch**, **BarcodeSearch.vb** και **EditForm.vb**.
- Οι φόρμες γενικών λειτουργιών, στις οποίες εντάσσονται οι φόρμες **Setting.vb**, **StartForm.vb**, **LoginForm.vb**, **PrintForm.vb**.

Στα υποκεφάλαια που ακολουθούν υπάρχει εκτενή αναφορά για κάθε μία από αυτές τις κατηγορίες, τόσο για τον τρόπο ομαδοποίησής τους, όσο και για το περιεχόμενο αυτών.

4.3.1 Φόρμες Περιγραφής

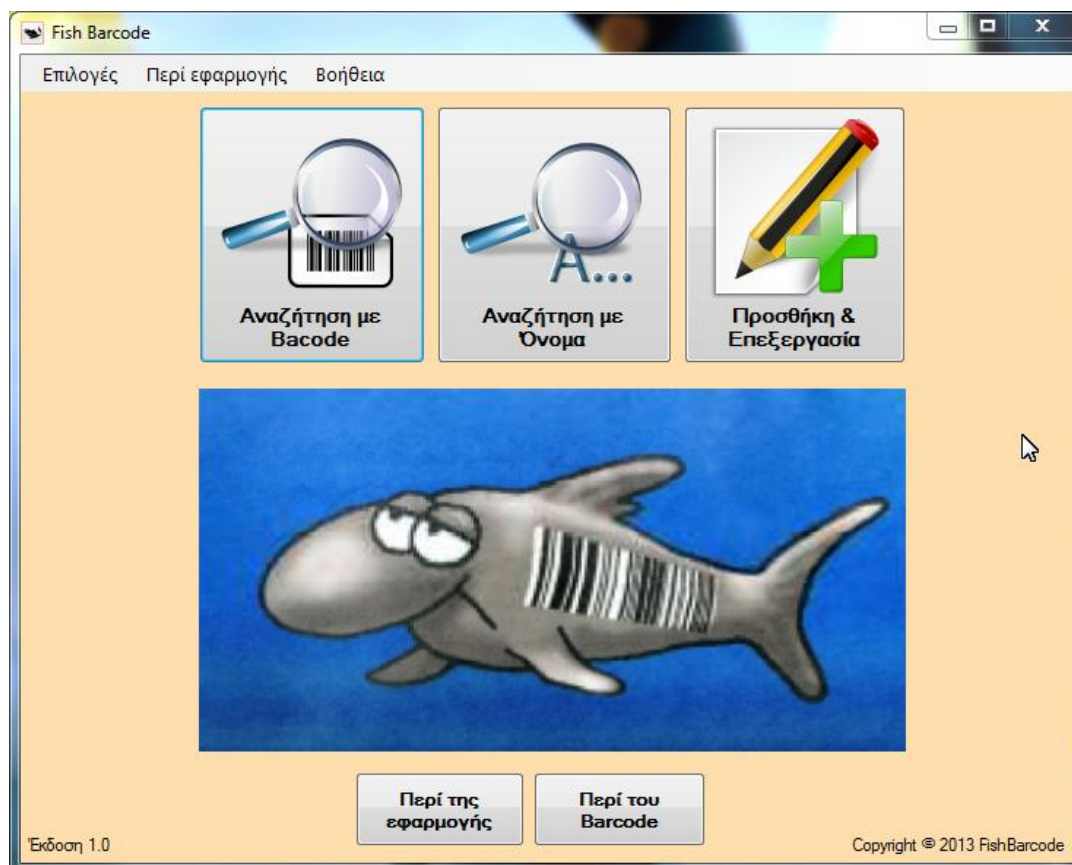
Αρχικά θα αναλυθούν οι φόρμες περιγραφής, οι οποίες όπως προκύπτει και από το όνομά τους χρησιμοποιούνται για να παρέχουν στο χρήστη χρήσιμες περιγραφές με ότι πρέπει να γνωρίζει ο χρήστης για την εφαρμογή. Οι φόρμες περιγραφής, στις οποίες εντάσσονται οι φόρμες **AboutAdd.vb**, **AboutApp.vb**, **AboutBcode.vb**, **AboutBs.vb** και **AboutNs.vb**.

Η πρώτη φόρμα αυτής της κατηγορίας είναι η AboutAdd.vb, η οποία περιεχει αναλυτικές πληροφορίες τόσο για την εν γέννη όσο και για. Ακολούθως έχουμε την φόρμα AboutBcode.vb, η οποία περιέχει όλες τις πληροφορίες που πρέπει να γνωρίζει ο εκάστοτε χρήστης για τον γραμμωτό κώδικα των ιχθύων.

4.3.2 Φόρμες Επιλογών

Οι φόρμες επιλογών είναι οι φόρμες των βασικών μενού της εφαρμογής και ονομάστηκαν έτσι, γιατί παρέχουν την επιλογή στο χρήστη ή στο διαχειριστή να επιλέξει ποία λειτουργία της εφαρμογής θέλει να χρησιμοποιήσει. Παρόλο που δεν έχουν κάποιον ιδιαίτερο προγραμματισμό μέσα τους, αποτελούν σημαντικό κομμάτι της εφαρμογής, είναι αρμόδιες για την περιήγηση σε αυτή.

Αρχικά, η **Menuform.vb**, αποτελείται από πέντε κουμπιά, ένα εσωτερικό μενού επιλογών (**Menustrip1**) και μία εικόνα, ενώ στο κάτω αριστερό μέρος της αναγράφεται η έκδοση της εφαρμογής και στο κάτω δεξιό το Copyright της, όπως φαίνεται και στην παρακάτω εικόνα (Εικόνα 12 - Γραφικό περιβάλλον της φόρμας επιλογών Menuform.vb).



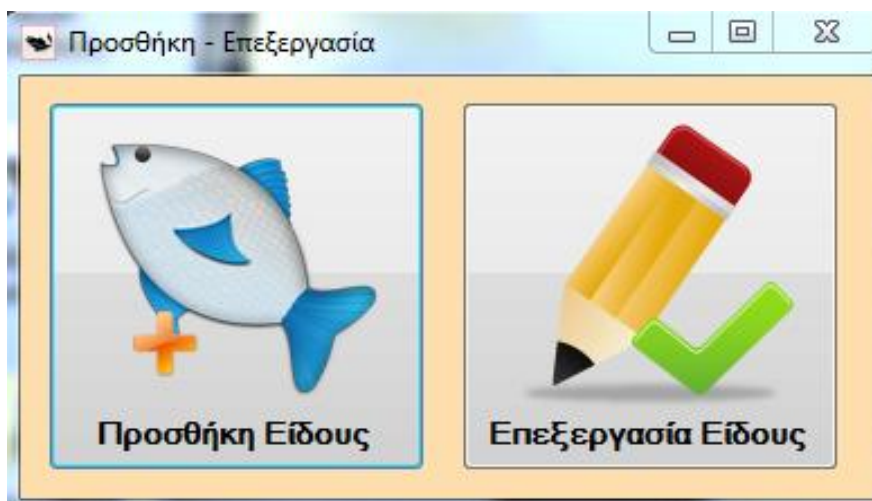
Εικόνα 12 - Γραφικό περιβάλλον της φόρμας επιλογών Menuform.vb

Το πρώτο κουμπί είναι το **‘Αναζήτηση με barcode’**, το οποίο μας οδηγεί στη φόρμα , **BarcodeSearch.vb**, η οποία είναι υπεύθυνη για την αναζήτηση πληροφοριών με βάση την αλληλουχία των ψηφίων του barcode.

Το δεύτερο κουμπί είναι το **‘Αναζήτηση με όνομα’**, το οποίο μας οδηγεί στη φόρμα , **NameSearch.vb**, η οποία είναι υπεύθυνη για την αναζήτηση πληροφοριών με βάση το όνομα που θα επιλέξουμε ως κριτήριο αναζήτησης.

Το τρίτο κουμπί είναι το **‘Προσθήκη & Επεξεργασία’**, το οποίο μας οδηγεί στη δεύτερη φόρμα επιλογών της εφαρμογής , τη **Menuform2.vb**, η οποία αποτελείται από μόνο δύο κουμπιά τα **‘Προσθήκη Είδους’** και **‘Επεξεργασία Είδους’**, όπως φαίνεται στην εικόνα 15 όπου απεικονίζεται το περιβάλλον την φόρμας επιλογών.

Το πρώτο κουμπί, δηλαδή το **‘Προσθήκη Είδους’**, ανοίγει τη φόρμα **AddForm.vb**, ενώ το δεύτερο, δηλαδή το **‘Επεξεργασία Είδους’**, ανοίγει τη φόρμα **EditForm.vb**.



Εικόνα 13 - Περιβάλλον της φόρμας επιλογών Menuform2.vb

Το τέταρτο κουμπί είναι το 'Περί της εφαρμογής', το οποίο μας οδηγεί στη φόρμα περιγραφής AboutApp.vb και το πέμπτο κουμπί, το 'Περί barcode', μας οδηγεί στην αντίστοιχη φόρμα περιγραφής AboutBs.vb.

Το εσωτερικό μενού επιλογών έχει τρία βασικά πεδία. Αρχικά, το πεδίο '**Επιλογές**', το οποίο δίνει την δυνατότητα διπλής εισόδου στις αντίστοιχες λειτουργίες της εφαρμογής που αναφέρθηκαν πιο πάνω. Το δεύτερο πεδίο είναι το '**Περί εφαρμογής**', το οποίο στο υπομενού που περιέχει δίνει την δυνατότητα στο χρήστη της εφαρμογής να γνωρίσει καλύτερα τις λειτουργίες της εφαρμογής, παρέχοντας πληροφορίες για την κάθε μία ξεχωριστά. Τα υποπεδία, αυτού του πεδίου, οδηγούν στις αντίστοιχες φόρμες περιγραφής, οι οποίες είναι κατά σειρά επιλογής η **AboutBs.vb**, η **AboutNs.vb** και η **AboutAdd.vb**

Η διαδικασία με την οποία οδηγούμαστε κάθε φορά σε κάποια άλλη φόρμα γίνεται με την υλοποίηση αντίστοιχων μεθόδων, οι οποίες περιέχουν τον κώδικα της φόρμας.

Η πρώτη μέθοδος που υλοποιείται σε αυτή τη φόρμα, είναι η **MainFormLoad()**, η οποία ξεκινάει την υλοποίηση της μόλις εισαχθούν σωστά τα στοιχεία εισόδου, που ζητούνται στη φόρμα εισόδου **LoginForm.vb**. Ο πηγαίος κώδικας που ακολουθείται δίνεται στο Παράρτημα Α (Πηγαίος Κώδικας Φόρμας Menuform.vb).

Εν κατακλείδι, οι φόρμες επιλογής είναι βασικό μέρος της εφαρμογής και λειτουργούν ως συνδετικός κρίκος με τα υπόλοιπα μέρη της εφαρμογής.

4.3.3 Φόρμες Βασικών Λειτουργιών

Σε αυτή την κατηγορία αυτή, εντάσσονται οι φόρμες, οι οποίες είναι υπεύθυνες για την υλοποίηση των βασικών λειτουργιών της εφαρμογής, οι οποίες έχουν ενσωματωμένο το μεγαλύτερο μέρος του

πηγαίου κώδικα. Όπως αναφέρθηκε και στον πρόλογο της υποενότητας 4.3, αυτές είναι οι φόρμες **AddForm.vb**, **NameSearch**, **BarcodeSearch.vb** και **EditForm.vb**.

4.3.3.1 AddForm.vb

Κρίνεται σκόπιμο, η πρώτη φόρμα που θα αναλυθεί, να είναι η **AddForm.vb**, η οποία είναι η φόρμα που αναλαμβάνει να προσθέσει ένα καινούριο είδος ιχθύ, στη βάση δεδομένων της εφαρμογής.

Αρχικά στη φόρμα εισάγουμε δύο βιβλιοθήκες την **MySql.Data.MySqlClien** και την **System.IO**. Η **MySql.Data.MySqlClien** είναι μία βιβλιοθήκη της MySql . Η **System.IO** είναι μια βιβλιοθήκη, η οποία έχει την αποκλειστική ευθύνη της διαχείρισης εισόδου και εξόδου των δεδομένων. Στη συνέχεια, έχουμε κλάση **AddForm**, η οποία αποτελείται από 7 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **Button1_Click**, οποία ενεργοποιείται όταν επιλεγθεί το κουμπί 'Δημιουργία barcode'.
2. Η μέθοδος **PictureBox1_Click**, η οποία ενεργοποιείται όταν επιλεγθεί το πλαίσιο, στο οποίο θα εισαχθεί η εικόνα της μορφολογίας του είδους.
3. Η μέθοδος **PictureBox2_Click**, η οποία ενεργοποιείται όταν επιλεγθεί το πλαίσιο, στο οποίο θα εισαχθεί η εικόνα της γεωγραφικής εξάπλωσης του είδους.
4. Η μέθοδος **ButtonSave Species_Click**, η οποία ενεργοποιείται όταν επιλεγθεί το κουμπί 'Αποθήκευση'.
5. Η μέθοδος **OpenFileDialog1_FileOk**, η οποία ενεργοποιείται όταν επιλεγθεί η εικόνα της μορφολογίας του είδους.
6. Η μέθοδος **OpenFileDialog2_FileOk**, η οποία ενεργοποιείται όταν επιλεγθεί η εικόνα της γεωγραφικής εξάπλωσης του είδους.
7. Η μέθοδος **AddForm_Load**, η οποία ενεργοποιείται με το άνοιγμα της συγκεκριμένης φόρμας.

Ακόμα, μέσα στην κλάση έχουν δηλωθεί τρεις μεταβλητές, οι οποίες είναι οι εξής:

1. Η μεταβλητή **maxidnum** ως ακέραιος, η οποία χρησιμοποιείται
2. Η μεταβλητή **spid** ως ακέραιος, η οποία χρησιμοποιείται
3. Η μεταβλητή **fn** ως μία νέα **AutoCompleteStringCollection**, δηλαδή μία συλλογή αλφαριθμητικών για αυτόματη καταχώρηση.

Ξεκινώντας με την ανάλυση της μεθόδου **Button1_Click**, η οποία δίνεται στο Παράρτημα Β (Κώδικας Button1_Click – 1^ο μέρος) και της οποίας ο κώδικας της οποίας θα δοθεί σε 3 κομμάτια, και με την υλοποίηση της με την μέθοδο **if**, μπορεί και ελέγχει αν συμπληρώθηκαν τα 2 υποχρεωτικά πεδία της φόρμας, τα οποία είναι το 'family name' και το **scientific_name**. Έπειτα,

δηλώνεται η μεταβλητή **m_DataBaseConnection** ως μία νέα **MySqlDataBaseConnection**, δηλαδή δημιουργώντας έτσι μία νέα σύνδεση με την MySql με τις εξής παραμέτρους:

1. Ως **server** ορίζεται ο **localhost**.
2. Ως **user_id** ορίζεται η ρίζα της βάσης **root**.
3. Ως **password** ορίζεται ο κωδικός **123**.
4. Ως **βάση**, από την οποία θα παρθούν τα δεδομένα, ορίζεται η **fish_database**.
5. Ως **charset**, δηλαδή το σύστημα το οποίο έχουν γραφτεί τα δεδομένα, ορίζεται το **utf8**.
6. Η τιμή **pooling** ορίζεται ως **false**.

Η μέθοδος Pooling που έχει χρησιμοποιηθεί και την οποία παρέχει το ADO.NET, το οποίο είναι μία βιβλιοθήκη της Microsoft, βοηθάει στην σύνδεση με την βάση δεδομένων, με σκοπό την αύξηση των επιδόσεων και της ταχύτητας σύνδεσης των διεργασιών με την βάση.

Στη συνέχεια, ορίζεται ένας adapter τύπου **MySqlDataAdapter**, τον **SQLDataAdapter**, ο οποίος αναλαμβάνει να κάνει την σύνδεση με την βάση δεδομένων και ένα **DataSet**, το **SQLDataSet**, στο οποίο θα αποθηκευτούν τα δεδομένα προσωρινά μέχρι την υλοποίησή τους. Έπειτα δημιουργείται η μεταβλητή strSQL, η οποία δηλώνεται τύπου String.

Μέσα στην μεταβλητή αυτή εισάγονται τα δεδομένα από 2 εμφωλευμένα Query. Το δεύτερο, κατά σειρά εμφάνισης Query, βρίσκει την οικογένεια του ιχθύος (family_name), που έχει ίδιο όνομα με αυτό που έχει πληκτρολογήσει ή επιλέξει ο χειριστής, ενώ το πρώτο κρατάει για κάθε ένα επίπεδο (από το επίπεδο family μέχρι και το επίπεδο superclass), το πεδίο num_bar και αποθηκεύει στην αντίστοιχη μεταβλητή (π.χ. το t1.num_bar, το οποίο είναι ο αύξοντα αριθμός του επιπέδου της υπερτάξης αποθηκεύεται στη μεταβλητή lev1bc). Με αυτό τον τρόπο συνδέονται εσωτερικά οι δύο πίνακες της βάσης.

Στη συνέχεια, δίνεται στον **SQLDataAdapter** μορφή, δηλώνοντας των ως ένα νέο **MySqlDataAdapter**, με παραμέτρους τη μεταβλητή strSQL, δηλαδή τι θα πρέπει να υλοποιήσει, και την **m_DatabaseConnection**, δηλαδή τη σύνδεση που θα πρέπει να χρησιμοποιήσει.

Τέλος, δηλώνεται το **SQLDataSet** ως νέο αντικείμενο και καλείται η μέθοδος **Fill**, του **SQLAdater**, η οποία αναλαμβάνει να γεμίσει τον adapter με τα δεδομένα του **DataSet**. Στη συνέχεια ακολουθεί το 2^ο μέρος του κώδικα που δίνεται στο Παράρτημα Β (Κώδικας Button1_Click – 2^ο μέρος) κατά σειρά κομμάτι κώδικα της μεθόδου.

Στη συνέχεια, αναλύεται η διαδικασία με την οποία δημιουργείται η αλληλουχία των ψηφίων, των 7 πρώτων επιπέδων της συστηματικής, δημιουργώντας αρχικά τις μεταβλητές bcode, thesuperclass, theclass, thesubclass, thesuperorder, theorder, thesuborder και thefamily, οι οποίες είναι τύπου String.

Στην πρώτη μεταβλητή (bcode), αποθηκεύεται αρχικά η αλληλουχία των 7 πρώτων επιπέδων. Οι υπόλοιπες χρησιμοποιούνται για την αποθήκευση των τροποποιημένων δεδομένων, που περιέχονται στις μεταβλητές lev_bc. Ακολούθως με τη μέθοδο **'For Each row As DataRow In SQLDataSet.Tables(0).Rows'**, τα δεδομένα των μεταβλητών 'lev_bc' μεταφέρονται στις μεταβλητές που δημιουργήθηκαν νωρίτερα.

Στην τελική του μορφή, το κομμάτι του κάθε επιπέδου της αλληλουχίας, έχει συγκεκριμένο μέγεθος χαρακτήρων. Τα τρία πρώτα επίπεδα αποτελούνται από 1 μόλις χαρακτήρα-ψηφίο. Για το λόγο αυτό, μεταφέρονται αυτούσια. Το τέταρτο και πέμπτο κατά σειρά επίπεδο (Superorder-Suborder) αποτελούνται από 2 χαρακτήρες-ψηφία. Για τον λόγο αυτό, γίνεται έλεγχος στο εύρος των ψηφίων και όπου χρειάζεται προστίθεται ένα μηδενικό μπροστά. Αντίστοιχη διαδικασία ακολουθείτε και για το έκτο και έβδομο επίπεδο, τα οποία αποτελούνται από τρεις χαρακτήρες-ψηφία, προσθέτοντας ένα ή δύο μηδενικά αντίστοιχα. Τέλος, μετά την κατάλληλη τροποποίηση που έγινε στους χαρακτήρες-ψηφία του καθενός επιπέδου ξεχωριστά, αποθηκεύονται στην μεταβλητή bcode που είχε δημιουργηθεί πιο πάνω.

Στη συνέχεια, εκτελείται ένα Query στην μεταβλητή strSQL, στο ζητείται να συλλέξει σε αυτή όλα τα famiy_id από τον πίνακα species, για τα οποία ισχύει ότι το family_name είναι ίδιο με αυτό που εισήχθη στο αντίστοιχο textbox. Δίνεται στον **SQLDataAdapter** μορφή, δηλώνοντας των ως ένα νέο **MySqlDataAdapter**, με παραμέτρους τη μεταβλητή strSQL, δηλαδή τι θα πρέπει να υλοποιήσει, και την **m_DatabaseConnection**, δηλαδή τη σύνδεση που θα πρέπει να χρησιμοποιήσει.

Τέλος, δηλώνεται το **SQLDataSet** ως νέο αντικείμενο και καλείται η μέθοδος **Fill**, του **SQLAdater**, η οποία αναλαμβάνει να γεμίσει τον adapter με τα δεδομένα του **DataSet**. Αυτή η διαδικασία είναι βασική και επαναλαμβάνεται πολλές φορές στον κώδικα, όχι μόνο της **AddForm**, αλλά και σε όλες τις φόρμες, στις οποίες εκτελούνται λειτουργίες. Έτσι ακολουθεί το 3^ο μέρος του κώδικα, ο οποίος δίνεται στο Παράρτημα Β (Κώδικας Button1_Click – 3^ο μέρος).

Αρχικά, μέσω της μεθόδου **For each Row**, διατηρείται στη μεταβλητή spid, το family_id. Η ανάλυση θα γίνει στη μέθοδο **btnSaveSpecies_Click**, η οποία αναλάβει να αποθηκεύσει όλα τα στοιχεία που χρειάζεται ο πίνακας species για να ολοκληρώσει μία εγγραφή. Στη συνέχεια, αναλύεται η διαδικασία με την οποία η αλληλουχία των ψηφίων, παίρνει την τελική της μορφή.

Εκτελείται ένα Query στην μεταβλητή strSQL, με το οποίο ζητείται να συλλέξει σε αυτή το μέγιστο species_id από τον πίνακα species. Έπειτα δημιουργείται μία βοηθητική μεταβλητή (maxid) τύπου String, στην οποία αποθηκεύουμε το μέγιστο species_id. Στη συνέχεια, μετατρέπεται η αλφαριθμητική τιμή της μεταβλητής σε αριθμητική και την αυξάνουμε κατά ένα. Ακολούθως,

προστίθεται τους χαρακτήρες-ψηφία, του 8ου και τελευταίου επιπέδου στην μεταβλητή `bcode`, η οποία περιέχει ήδη τους χαρακτήρες-ψηφία των πρώτων 7 επιπέδων.

Για τη δημιουργία της εικόνα του γραμμωτού κώδικα, χρησιμοποιείται η **μέθοδος `MakeBarcodeImage()`**, η οποία βρίσκεται στην **κλάση `Code128Rendering`**. Η κλάση `Code128Rendering` είναι μία βιβλιοθήκη της `vb.net`, η οποία υπάρχει ελεύθερη στο internet και αναλαμβάνει να μετατρέψει μία συμβολοσειρά σε εικόνα γραμμωτού κώδικα. Έπειτα, μετατρέπεται το περιεχόμενο του της εικόνας σε ορατό και δίνεται η δυνατότητα ενεργοποίησης στο κουμπί αποθήκευσης. Με αυτή την διαδικασία ολοκληρώνεται το πρώτο μέρος της μεθόδου `if` που υλοποιείται και στα τρία κομμάτια κώδικα που αναλύθηκαν.

Τέλος, στο δεύτερο μέρος της, δηλαδή την `else`, υπάρχει το αντίστοιχο μήνυμα σφάλματος σε περίπτωση λανθασμένης εισαγωγής των δύο βασικών στοιχείων. Η ανάλυση των μεθόδων **`PictureBox1_Click`**, **`PictureBox2_Click`**, οι οποίες ενεργοποιούνται όταν επιλεχθούν τα αντίστοιχα πλαίσια εικόνων της φόρμας, υλοποιώντας το άνοιγμα αυτών των πλαισίων και η μέθοδος **`btnSaveSpecies_Click`**, η οποία ενεργοποιείται όταν επιλεχθεί το κουμπί 'Αποθήκευση' της φόρμας και υλοποιεί την διαδικασία αποθήκευσης των στοιχείων στο πεδίο `species` της βάσης δίνονται αναλυτικά μέσω του κώδικα τους στο Παράρτημα Γ (Κώδικας Μεθόδων `PictureBox1_Click`, `PictureBox2_Clic` και `btnSaveSpecies_Click`).

Αρχικά, δηλώνεται η μεταβλητή **`m_DataBaseConnection`** ως μία νέα **`MySqlConnection`** ακολουθώντας την ίδια διαδικασία που αναλύθηκε στην προηγούμενη μέθοδο, ενώ στη συνέχεια, δηλώνονται 2 μεταβλητές για κάθε μία από τις εικόνες που έχουν επιλεχθεί. Η πρώτη, `memImage`, δηλώνεται ως ένα νέο **`System.IO.Memory`**. Έτσι, η εικόνα που θα αποθηκευτεί αυτή τη μεταβλητή θα μετατραπεί σε μία ροή δεδομένων.

Αντίστοιχα, η `byteImage` δηλώνεται ως `byte`. Έτσι, η εικόνα που θα αποθηκευτεί σε αυτή τη μεταβλητή θα πάρει δυαδική μορφή. Αυτό γίνεται, γιατί δεν δύναται να εισάχθει μία εικόνα στην βάση με τη μορφή που έχει. Για αυτό πρέπει να μετατραπεί σε κάποια άλλη μορφή και η μορφή που επιλέχθηκε είναι η δυαδική.

Στη συνέχεια, ελέγχεται αν είναι γεμάτα τα αντίστοιχα πλαίσια εικόνων και αν είναι, τροποποιείται η εικόνα σε ένα ρεύμα δεδομένων και έπειτα στο δυαδικό σύστημα. Στη συνέχεια, εκτελείται ένα `Query` στην μεταβλητή `strSQL`, με το οποίο ζητείται να εισαχθούν στον πίνακα `species` τα εξής:

- Η τιμή που περιέχεται στη μεταβλητή κλάσης `maxidnum`, η οποία περιέχει τον αύξοντα αριθμό του είδους, ο οποίος είναι και κύριο κλειδί.

- Η τιμή που περιέχεται στο πλαίσιο κειμένου του `scientific_name`, το οποίο περιέχει το επιστημονικό όνομα του είδους. Το αντίστοιχο ισχύει και για τα πλαίσια κειμένου, τα οποία περιέχουν το ελληνικό όνομα, το αγγλικό όνομα, την οικογένεια, το γένος και την περιγραφή.
- Η τιμή που περιέχεται στη μεταβλητή κλάσης `spid`, η οποία περιέχει τον αύξοντα αριθμό της οικογένειας από τον πίνακα `dbtee1`, δηλαδή το `id` του 'γονιού'.
- Οι παραμετρικές μεταβλητές `@data` και `@data1`. Όπου γίνεται αναφορά σε αυτές θα αντιστοιχίζεται η ανάλογη μεταβλητή `byImage`, δηλαδή με την αντίστοιχη δυαδική μορφή της εικόνας. Αυτό γίνεται, γιατί δεν μπορούμε να εισάγουμε μία εικόνα στην βάση με τη μορφή που έχει.

Οι επόμενες εντολές **`sqlCommand`** δίνουν τη δυνατότητα να υλοποιηθεί η παραμετροποίηση που αναφέρθηκε πιο πάνω. Η ιδιότητα **`AddWithValue`**, χρησιμοποιείται για να αντιστοιχίσει την παράμετρο με την μεταβλητή `byImage`. Τέλος, χρησιμοποιείται η μέθοδος **`ExecuteNonQuery`**, μιας και το Query που χρησιμοποιείται εισάγει δεδομένα και δεν υπάρχει κάποιο άλλο που να επιστρέφει δεδομένα.

Στη συνέχεια του Παραρτήματος (Παράρτημα Δ – Κώδικας Μεθόδων `OpenFileDialog1_FileOk`, `OpenFileDialog2_FileOk` και `AddForm_Load`) , δίνεται η ανάλυση του κώδικα των μεθόδων **`OpenFileDialog1_FileOk`**, **`OpenFileDialog2_FileOk`**, οι οποίες χειρίζονται την διαδικασία επιλογής εικόνας από τον χρήστη και η μέθοδος **`AddForm_Load`**, η οποία ενεργοποιείται με το που ανοίγει η φόρμα.

Αρχικά, δηλώνεται η μεταβλητή **`m_DataBaseConnection`** ως μία νέα **`MySqlConnection`**, δηλαδή δημιουργούμε μία νέα σύνδεση με την MySql οι παράμετροι και η χρησιμότητα της οποίας αναλύθηκε στην αρχή αυτής της υποενότητας. Στη συνέχεια, ορίζεται ένας adapter τύπου **`MySqlDataAdapter`**, τον **`SQLDataAdapter`**, ο οποίος αναλαμβάνει να κάνει την σύνδεση με την βάση δεδομένων και ένα **`DataSet`**, το **`SQLDataSet`**, στο οποίο θα αποθηκευτούν τα δεδομένα προσωρινά μέχρι την υλοποίησή τους.

Έπειτα δημιουργείται η μεταβλητή `strSQL`, η οποία δηλώνεται τύπου `String` και εκτελείται ένα Query στην μεταβλητή `strSQL`, με το οποίο ζητείται να συλλεχθούν όλα τα μοναδικά `family_name` από τον πίνακα `species`. Η μοναδικότητα επιτυγχάνεται με την εντολή `Distinct`.

Στην συνέχεια ζητείται να εισαχθούν, μέσα στην μεταβλητή κλάσης `fn` όλα τα `family_name`. Παράλληλα υλοποιείται μία μέθοδος ελέγχου, η οποία ελέγχει αν τα `family_name` που προϋπάρχουν, είναι σωστά. Αν βρει κάποια εισαγωγή που περιέχει κάποιο σφάλμα την παρακάμπτει, με σκοπό να μην δημιουργήσει πρόβλημα στην λειτουργία της εφαρμογής. Έπειτα η σύνδεση που είχε αρχικά δημιουργηθεί τερματίζεται.

Στη συνέχεια, οι υπόλοιπες γραμμές κώδικα που υπάρχουν αφορούν την μέθοδο `AutoComplete`, η οποία χρησιμοποιείται για την παραγωγή προτάσεων για την επιλογή του `family_name`. Οι ιδιότητες της ***AutoComplete***, οι οποίες χρησιμοποιούνται είναι οι ακόλουθες:

- Η ιδιότητα ***AutoCompleteMode***, η οποία χρησιμοποιείται για να ορίσει τον τρόπο με τον οποίο θα γίνει η εμφάνιση των υποψήφιων δεδομένων για εισαγωγή.
- Η ιδιότητα ***AutoCompleteSource***, η οποία χρησιμοποιείται για να ετοιμάσει το πλαίσιο κειμένου, ώστε να αποδεχτεί μια προσαρμοσμένη πηγή δεδομένων για την αυτόματη συμπλήρωση του.
- Η ιδιότητα ***AutoCompleteCustomSource***, η οποία χρησιμοποιείται για να ορίσει τον τρόπο εισαγωγής της λίστας των τιμών σου συλλέχθηκαν.
- Η ιδιότητα ***AutoCompleteSuggest***, η οποία χρησιμοποιείται για την αναγνώριση των χαρακτήρων που εισάγονται, σε επίπεδο πραγματικού χρόνου εκτέλεσης, με βάση τους οποίους δημιουργεί λίστες συμβουλών.

4.3.3.2 *EditForm*

Η δεύτερη φόρμα που θα αναλυθεί, να είναι η `EditForm.vb`, η οποία είναι η φόρμα που αναλαμβάνει να τροποποιήσει ένα ήδη υπάρχων είδος ιχθύος, στη βάση δεδομένων της εφαρμογής. Η κλάση της, `EditForm`, αποτελείται από 11 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος ***PictureBox1_Click***, η οποία ενεργοποιείται όταν επιλεγθεί το πλαίσιο, στο οποίο είτε θα εισαχθεί είτε θα τροποποιηθεί η εικόνα της μορφολογίας του είδους.
2. Η μέθοδος ***PictureBox2_Click***, η οποία ενεργοποιείται όταν επιλεγθεί το πλαίσιο, στο οποίο είτε θα εισαχθεί είτε θα τροποποιηθεί η εικόνα της γεωγραφικής εξάπλωσης του είδους. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου ***PictureBox1_Click*** και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.
3. Η μέθοδος ***btnSearch_Click***, η οποία αναλαμβάνει την εύρεση του κριτηρίου που θα δοθεί από το διαχειριστή με στόχο την τροποποίηση.
4. Η μέθοδος ***RadioButton1_Paint***, η οποία αναλαμβάνει να χρωματίσει το πρώτο `RadioButton` με κόκκινο χρώμα. Το default χρώμα ήταν το μπλε. Αυτό γίνεται στα πλαίσια της ευκολία του χρήστη που θέλουμε να παρέχει η εφαρμογή στον κάθε χρήστη. Το ίδιο ισχύει για όλες τις παρόμοιες μεθόδους, οι οποίες αναλαμβάνουν να χρωματίσουν κάποιο `RadioButton`.
5. Η μέθοδος ***RadioButton2_Paint***, η οποία αναλαμβάνει να χρωματίσει το δεύτερο `RadioButton` με κόκκινο χρώμα. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου ***RadioButton1_Paint*** και για το λόγο αυτό θα παραληφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.

6. Η μέθοδος **OpenFileDialog1_FileOk**, η οποία ενεργοποιείται όταν επιλεγθεί η εικόνα της μορφολογίας του είδους.
7. Η μέθοδος **OpenFileDialog2_FileOk**, η οποία ενεργοποιείται όταν επιλεγθεί η εικόνα της γεωγραφικής εξάπλωσης του είδους. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **OpenFileDialog1_FileOk** και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.
8. Η μέθοδος **btnSaveSpecies_Click**, η οποία ενεργοποιείται όταν επιλεγθεί το κουμπί **‘Αποθήκευση Αλλαγών’**.
9. Η μέθοδος **EditForm_Load**, η οποία ενεργοποιείται με το άνοιγμα της συγκεκριμένης φόρμας.
10. Η μέθοδος **RadioButton1_CheckedChanged**, η οποία αναλαμβάνει, όταν επιλεγθεί το πρώτο RadioButton, να δημιουργήσει τις ανάλογες προτάσεις επιλογών μέσω της μεθόδου **AutoComplete**.
11. Η μέθοδος **RadioButton2_CheckedChanged**, η οποία αναλαμβάνει, όταν επιλεγθεί το δεύτερο RadioButton, να δημιουργήσει τις ανάλογες προτάσεις επιλογών μέσω της μεθόδου **AutoComplete**. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **RadioButton1_CheckedChanged** και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.

Στο Παράρτημα Ε (Πηγαίος Κώδικας της Μεθόδου **btnSearch_Click**), δίνεται ο πηγαίος κώδικας της μεθόδου **btnsearch_click** σε δύο μέρη.

Αρχικά, στο πρώτο μέρος (Παράρτημα Ε - Πηγαίος Κώδικας της Μεθόδου **btnSearch_Click** - 1^ο μέρος) η κλάση εξοπλίζεται με δύο βιβλιοθήκες, οι οποίες είναι η **MySql.Data.MySqlClient** και η **System.IO**, η χρησιμότητα και λειτουργία των οποίων αναλύθηκε στη φόρμα **AddForm.vb**. Έπειτα, δηλώνονται σαν μεταβλητές κλάσης οι **sn**, την **bc**, οι οποίες ορίζονται ως συλλογές χαρακτήρων αυτόματης συμπλήρωσης και την **bcode**, η οποία ορίζεται ως αλφαριθμητική.

Αρχικά, η μέθοδος **btnSearch_Click**, ελέγχει αν έχει συμπληρωθεί το TextBox του κριτηρίου, ελέγχοντας αν το πλήθος των χαρακτήρων του είναι μεγαλύτερο από μηδέν. Έπειτα, ορίζεται η μεταβλητή **qstr** ως αλφαριθμητική, στην οποία δίνουμε ως default τιμή το **scientific_name**. Και έπειτα ελέγχεται ποιο από τα δύο κριτήρια έχει επιλεγθεί και δίνεται στην μεταβλητή **qstr** η αντίστοιχη τιμή. Έπειτα, δηλώνεται η μεταβλητή **m_DataBaseConnection** ως μία **νέα MySqlConnection**, δημιουργώντας μία νέα σύνδεση με την MySql και των παραμέτρων και η χρησιμότητα της οποίας αναλύθηκε στην ανάλυση της φόρμας **AddForm.vb**.

Στη συνέχεια, ορίζεται έναν adapter τύπου **MySqlDataAdapter**, τον **SQLDataAdapter**, ο οποίος αναλαμβάνει να κάνει την σύνδεση με την βάση δεδομένων και ένα **DataSet**, το **SQLDataSet**, στο οποίο θα αποθηκευτούν τα δεδομένα προσωρινά μέχρι την υλοποίησή τους και ακολούθως, εκτελείται ένα Query στην μεταβλητή strSQL, με το οποίο ζητείται να συλλέξει από τον πίνακα species τα στοιχεία της εγγραφής, η οποία ικανοποιεί την αντίστοιχη συνθήκη.

Στη συνέχεια, δίνεται στον **SQLDataAdapter** μορφή, δηλώνοντας των ως ένα νέο **MySqlDataAdapter**, με παραμέτρους τη μεταβλητή strSQL, δηλαδή τι θα πρέπει να υλοποιήσει, και την **m_DatabaseConnection**, δηλαδή τη σύνδεση που θα πρέπει να χρησιμοποιήσει.

Τέλος, δηλώνεται το **SQLDataSet** ως νέο αντικείμενο και καλείται η μέθοδος **Fill**, του **SQLAdater**, η οποία αναλαμβάνει να γεμίσει τον adapter με τα δεδομένα του **DataSet**.

Στο δεύτερο μέρος του κώδικα (Παράρτημα Ε - Πηγαίος Κώδικας της Μεθόδου btnSearch_Click - 2^ο μέρος), μέσω της μεθόδου **For Each Row**, δίνονται τιμές στα πλαίσια κειμένου της φόρμας και στη μεταβλητή bcode. Έτσι η μεταβλητή bcode παίρνει το περιεχόμενο της εγγραφής του πεδίου barcode, το πλαίσιο κειμένου του scientific name παίρνει το περιεχόμενο της εγγραφής του πεδίου scientific_name κ.ο.κ.

Στη συνέχεια, ελέγχεται αν το πεδίο εικόνας της μορφολογίας έχει κάποιο περιεχόμενο. Εφόσον υπάρχει, ακολουθείται η διαδικασία που ακολουθήθηκε και στη φόρμα **AddForm.vb**, δηλαδή μετατρέπεται αρχικά το περιεχόμενο σε μια ροή δεδομένων και στη συνέχεια σε δυαδική μορφή. Η ίδια διαδικασία ακολουθείτε και για το πεδίο εικόνας της γεωγραφικής κατανομής. Σε περίπτωση που δεν έχει επιλεγεί κάποιο κριτήριο εμφανίζεται στην οθόνη αντίστοιχο προειδοποιητικό μήνυμα. Τέλος, αφού έχει επιλεγεί κάποιο κριτήριο, ενεργοποιείται το κουμπί της αποθήκευσης. Το συγκεκριμένο κουμπί αρχικά είναι ανενεργό.

Στη συνέχεια στο Παράρτημα ΣΤ (Πηγαίος Κώδικας Μεθόδων RadioButton_1Paint btnSaveSpecies_Click) δίνεται ο πηγαίος κώδικας από τη μέθοδο **RadioButton_1Paint**, η οποία χρησιμοποιείται για να κάνουμε κόκκινο το RadioButton και τη μέθοδο **btnSaveSpecies_Click**.

4.3.3.3 **BarcodeSeach.vb**

Η τρίτη φόρμα που θα αναλυθεί, να είναι η **BarcodeSearch.vb**, η οποία είναι η φόρμα που αναλαμβάνει να αναζητήσει στη βάση δεδομένων το είδος με βάση το barcode που θα της δοθεί. Η κλάση της, **BarcodeSearch**, αποτελείται από 4 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **Button1_Click**, η οποία αναλαμβάνει να υλοποιήσει τη διαδικασία της αναζήτησης με βάση το Barcode, το οποίο θα δοθεί.

2. Η μέθοδος **Button2_Click**, η οποία αναλαμβάνει τη μορφοποίηση των στοιχείων κατά την επιλογή του κουμπιού **‘Εκτύπωση Στοιχείων’**.
3. Η μέθοδος **BarcodeSearch_Load**, η οποία ενεργοποιείτε με το άνοιγμα της συγκεκριμένης φόρμας.
4. Η μέθοδος **Button1_Click_1**, η οποία αναλαμβάνει τη μορφοποίηση των στοιχείων κατά την επιλογή του κουμπιού **‘Εκτύπωση Καρτέλας’**.

4.3.3.4 NameSearch

Η τρίτη φόρμα που θα αναλυθεί, να είναι η **NameSearch.vb**, η οποία είναι η φόρμα που αναλαμβάνει να αναζητήσει στη βάση δεδομένων το είδος με βάση το όνομα, το οποίο θα της δοθεί. Η κλάση της, **NameSearch**, αποτελείται από 11 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **btnSearch_Click**, η οποία αναλαμβάνει την εύρεση του κριτηρίου αναζήτησης που θα επιλεγεί.
2. Η μέθοδος **Button2_Click**, η οποία αναλαμβάνει τη μορφοποίηση των στοιχείων κατά την επιλογή του κουμπιού **‘Εκτύπωση Στοιχείων’**. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **Button2_Click**, της φόρμας **EditForm.vb**, και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.
3. Η μέθοδος **RadioButton1_Paint**, η οποία αναλαμβάνει να χρωματίσει το πρώτο **RadioButton** με κόκκινο χρώμα. Το default χρώμα ήταν το μπλε. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **RadioButton1_Paint**, της φόρμας **EditForm.vb** και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω. Το ίδιο ισχύει και για τις μεθόδους **RadioButton2_Paint** και **RadioButton3_Paint**.
4. Η μέθοδος **RadioButton2_Paint**, η οποία αναλαμβάνει να χρωματίσει το δεύτερο **RadioButton** με κόκκινο χρώμα
5. Η μέθοδος **RadioButton3_Paint**, η οποία αναλαμβάνει να χρωματίσει το τρίτο **RadioButton** με κόκκινο χρώμα
6. Η μέθοδος **NameSearch_Load**, η οποία ενεργοποιείτε με το άνοιγμα της συγκεκριμένης φόρμας.
7. Η μέθοδος **RadioButton1_CheckedChanged**, η οποία αναλαμβάνει, όταν επιλεγθεί το πρώτο **RadioButton**, να δημιουργήσει τις ανάλογες προτάσεις επιλογών μέσω της μεθόδου **AutoComplete**. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **RadioButton1_CheckedChanged**, της φόρμας **EditForm.vb**, και για το λόγο αυτό θα παραληφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω. Το ίδιο ισχύει και για τις μεθόδους **RadioButton2_CheckedChanged** και **RadioButton3_CheckedChanged**.

8. Η μέθοδος **RadioButton2_CheckedChanged**, η οποία αναλαμβάνει, όταν επιλεγθεί το δεύτερο RadioButton, να δημιουργήσει τις ανάλογες προτάσεις επιλογών μέσω της μεθόδου **AutoComplete**.

9. Η μέθοδος **RadioButton3_CheckedChanged**, η οποία αναλαμβάνει, όταν επιλεγθεί το τρίτο RadioButton, να δημιουργήσει τις ανάλογες προτάσεις επιλογών μέσω της μεθόδου **AutoComplete**.

10. Η μέθοδος **cardbtn_Click**, η οποία αναλαμβάνει τη μορφοποίηση των στοιχείων κατά την επιλογή του κουμπιού **‘Εκτύπωση Καρτέλας’**. Ο κώδικας αυτής της μεθόδου είναι παρόμοιος με τον κώδικα της μεθόδου **Button1_Click_1**, της φόρμας **EditForm.vb**, και για το λόγο αυτό θα παραλειφθεί στον πηγαίο κώδικα που θα δοθεί παρακάτω.

Απο την κλάση **NameSeach** δίνεται μόνο ο πηγαίος κώδικας από την μέθοδο **btnSeach_Click** στο Παράρτημα Η – (Πηγαίος Κώδικας Μεθόδου btnSeach_Click), σε 6 κομμάτια.

Αρχικά εξοπλίζεται η κλάση με δύο βιβλιοθήκες, οι οποίες είναι η **MySql.Data.MySqlClient** και η **System.IO**, η χρησιμότητα και λειτουργία των οποίων αναλύθηκε στη φόρμα AddForm.vb. Έπειτα, δηλώνονται σαν μεταβλητές κλάσης οι sn, gcn και en, οι οποίες ορίζονται ως συλλογές χαρακτήρων αυτόματης συμπλήρωσης. Στην αρχή της ακολουθείται η ίδια μεθοδολογία που ακολουθήθηκε στη φόρμα EditForm για την εύρεση κριτηρίου επιλογής και τη δημιουργία μιας νέας σύνδεσης με τη βάση δεδομένων.

4.3.4 Φόρμες Γενικών Λειτουργιών

Σε αυτή την κατηγορία αυτή, εντάσσονται οι φόρμες, οι οποίες είναι υπεύθυνες για την υλοποίηση γενικότερων λειτουργιών της εφαρμογής. Όπως αναφέρθηκε και στον πρόλογο της υποενότητας 4.3 αυτές είναι οι φόρμες **StartForm.vb**, **SettingsForm.vb**, **LoginForm.vb** και **PrintForm.vb**.

4.3.4.1 StartForm.vb

Η κλάση StartForm αυτής της φόρμας αποτελείται από 6 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **btnEnter_Click**, η οποία ενεργοποιείται με την επιλογή του κουμπιού **‘Αποθήκευση’**, το οποίο υπάρχει στη φόρμα.
2. Η μέθοδος **cbFao_SelectedValueChanged**, η οποία αναλαμβάνει να ελέγξει αν για την τιμή **FAO** που επιλέχθηκε υπάρχουν καταχωρίσεις.
3. Η μέθοδος **btnSettings_Click**, η οποία ενεργοποιείται με την επιλογή του κουμπιού **‘Ρυθμίσεις’** και ανοίγει την αντίστοιχη φόρμα **SettingsForm.vb**.

4. Η μέθοδος **StartForm_Load**, η οποία ενεργοποιείται με το άνοιγμα της συγκεκριμένης φόρμας και αναλαμβάνει να ενημερώσει όλες της φόρμες της εφαρμογής για την ρύθμιση φόντου, η οποία έχει επιλεγθεί.
5. Η μέθοδος **cbFao_MeasureItem**, η οποία αναλαμβάνει να μετρήσει τα το σύνολο των χαρακτήρων του κάθε Fao.
6. Η μέθοδος **cbFao_DrawItem**, η οποία με βάση την μέτρηση χαρακτήρων, που έχει κάνει η μέθοδος **cbFao_MeasureItem**, αναλαμβάνει να χρωματίσει με έντονο μαύρο χρώμα το **Fao 37**. Αυτό γίνεται στα πλαίσια της ευκολία του χρήστη. που θέλουμε να παρέχει η εφαρμογή στον κάθε χρήστη.

4.3.4.2 SettingsForm.vb

Η κλάση SettingsForm αυτής της φόρμας αποτελείται από 6 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **SettingsLoad**, η οποία ενεργοποιείται με το άνοιγμα της συγκεκριμένης φόρμας και παράλληλα κλείνει την φόρμα StartForm.vb.
2. Η μέθοδος **Settings_FormClosing**, η οποία ενεργοποιείται για να κλείσει τη φόρμα και παράλληλα ανοίγει την φόρμα **StartForm.vb**.
3. Η μέθοδος **lightblueRbn_CkeckedChanged**, η οποία αναλαμβάνει να δώσει σε όλες της φόρμες της εφαρμογής ως χρώμα φόντου το ανοιχτό μπλε.
4. Η μέθοδος **darkgreyRbn_CkeckedChanged**, η οποία αναλαμβάνει να δώσει σε όλες της φόρμες της εφαρμογής ως χρώμα φόντου το σκούρο γκριζο.
5. Η μέθοδος **lightorangeRbn_CkeckedChanged**, η οποία αναλαμβάνει να δώσει σε όλες της φόρμες της εφαρμογής ως χρώμα φόντου το ανοιγοχτό πορτοκαλί.
6. Η **μέθοδος defaultRbn_CkeckedChanged**, η οποία αναλαμβάνει να δώσει σε όλες της φόρμες της εφαρμογής ως χρώμα φόντου το γκρι, το οποίο είναι και το default.

4.3.4.3 LoginForm.vb

Η κλάση LoginForm αυτής της φόρμας αποτελείται από 3 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **Ok_Click**, η οποία ενεργοποιείται με την επιλογή του κουμπιού **'ok'**, το οποίο υπάρχει στη φόρμα και αναλαμβάνει να ελέγξει αν τα στοιχεία που πληκτρολογήθηκαν είναι σωστά.
2. Η μέθοδος **Cancel_Click**, η οποία ενεργοποιείται όταν επιλεγθεί το κουμπί **'Cancel'** και αναλαμβάνει να διακόψει την λειτουργία της φόρμας.

3. Η μέθοδος **LoginForm_Load**, η οποία ενεργοποιείται με το άνοιγμα της συγκεκριμένης φόρμας και παράλληλα κλείνει την φόρμα **StartForm.vb**.

4.3.4.4 PrintForm

Η κλάση LoginForm αυτής της φόρμας αποτελείται από 3 μεθόδους, οι οποίες είναι οι εξής:

1. Η μέθοδος **printBtn_Click**, η οποία ενεργοποιείται με το που ανοίξει η σελίδα.
2. Η μέθοδος **printDocument1_PrintPage**, η οποία ενεργοποιείται όταν επιλεγεί το κουμπί **‘Εκτύπωση’** της φόρμας και αναλαμβάνει τη διαδικασία της εκτύπωσης ανάλογα με τις επιλογές που θα δώσει ο χρήστης.
3. Η μέθοδος **pgSetup_BottonClick**, η οποία ενεργοποιείται όταν επιλεγεί το κουμπί **‘Διαμόρφωση σελίδας’** και αναλαμβάνει να διαμορφώσει τη σελίδα ανάλογα με τις επιλογές του χρήστη.
4. Η μέθοδος **printPreviewBtn_Click**, η οποία ενεργοποιείται όταν επιλεγεί το κουμπί **‘Προεπισκόπηση’** και αναλαμβάνει να υλοποιήσει την διαδικασία της προεπισκόπησης.

4.4 Εργαλεία Ανάπτυξης

Δύο ήταν τα βασικά εργαλεία που χρησιμοποιήθηκαν για να αναπτυχθεί η εφαρμογή της Συστηματικής Ταξινόμησης Ιχθύων μέσω γραμμωτού κώδικα. Το ένα έχει να κάνει με τον προγραμματισμό της εφαρμογής και το άλλο με την διαχείριση της βάσης δεδομένων της εφαρμογής.

Εκτός από αυτά, έγινε χρήση κι ενός προγράμματος επεξεργασίας εικόνας, στα πλαίσια της επεξεργασίας διάφορων εικόνων ιχθύων που λήφθηκαν από το Διαδίκτυο, ώστε να προσαρμοστούν στις απαραίτητες διαστάσεις και μέγεθος καθώς και ενός προγράμματος λήψης και επεξεργασίας κατευθείαν από την οθόνη του υπολογιστή.

Το εργαλείο προγραμματισμού που χρησιμοποιήθηκε είναι το Microsoft Visual Studio και ποιο συγκεκριμένα η έκδοση Ultimate. Το **Microsoft Visual Studio** είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft, το οποίο χρησιμοποιείται την δημιουργία πλήθους εφαρμογών, οι οποίες θα απευθύνονται σε όλες τις πλατφόρμες που υποστηρίζονται από τα Microsoft Windows, Windows Mobile, Windows CE, NET Framework, NET Compact Framework, το Microsoft Silverlight και γενικότερα σε ένα πλήθος άλλων περιβαλλόντων και ρυθμίσεων.

Το Visual Studio περιλαμβάνει προγράμματα επεξεργασίας, υποστήριξη και εντοπισμού σφαλμάτων σε επίπεδο πηγαίου κώδικα, ενσωματωμένα εργαλεία σχεδιαστής για τη δημιουργία

GUI εφαρμογών, σχεδίαση ιστοσελίδων καθώς και πλήθος άλλων εφαρμογών. Επιπλέον, υποστηρίζει πολλές διαφορετικές γλώσσες προγραμματισμού, με τη βοήθεια των γλωσσικών υπηρεσιών, οι οποίες επιτρέπουν στον επεξεργαστή κώδικα εντοπισμού σφαλμάτων σε οποιαδήποτε γλώσσα προγραμματισμού.

Το **XAMPP** είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets. Η ονομασία του είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

1. X (αναφέρεται στο "cross-platform" που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
2. Apache HTTP εξυπηρετητής
3. MySQL
4. PHP
5. Perl

Το XAMPP προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης και έχει δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl. Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

Αρχικά οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί.

Στην πράξη το XAMPP ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων. Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite. Η έκδοση του XAMPP 1.8.1 για Windows, η οποία χρησιμοποιήθηκε περιέχει :

- Apache 2.4.3
- MySQL 5.5.27
- PHP 5.4.7
- phpMyAdmin 3.5.2.2

- FileZilla FTP Server 0.9.41
- Tomcat 7.0.30 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.1.1 Portable
- XAMPP Control Panel 3.1.0

Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Το XAMPP με άλλα λόγια ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server.

4.5 Πηγές δεδομένων

Ένα κομμάτι της εφαρμογής αποτελούν και τα δεδομένα των ιχθύων αλλά και όλων των πληροφοριών που χρειάστηκε να συλλεχθούν. Λόγω του τεράστιου όγκου αυτών των δεδομένων για τη συγκεκριμένη εφαρμογή, το κομμάτι αυτό αποτελεί ένα σημαντικό μέρος της. Όπως αναφέρθηκε και στην ανάλυση της βάσης, τα επίπεδα που σχηματίζουν τη συστηματική κατάταξη για κάθε είδος ιχθύος είναι 8, αλλά και για τον λόγο του ότι τα είδη των ιχθύων είναι 540, η βάση δεδομένων της εφαρμογής χρειάστηκε να αποθηκεύσει έναν μεγάλο όγκο δεδομένων.

Επιπλέον, ένα μεγάλο μέρος αυτού του όγκου αφορά τις εικόνες της μορφολογίας και της γεωγραφικής εξάπλωσης για κάθε ένα από τα είδη αυτά. Για τον λόγο αυτό θεωρήθηκε σκόπιμο να γίνει ειδική αναφορά στις πηγές που χρησιμοποιήθηκαν για την συγκέντρωση όλων αυτών των δεδομένων, τα οποία μετατρέπει σε οργανωμένες πληροφορίες η εφαρμογή.

Τρεις είναι οι βασικότερες πηγές που συνετέλεσαν για την άντληση των πληροφοριών για την συστηματική κατάταξη των ιχθύων, αλλά και κάθε πληροφορίας γύρω από αυτά τα είδη ιχθύων:

1. **Κασπίρης, 2000**, ένα βιβλίο το οποίο περιέχει μία κλείδα προσδιορισμού ιχθύων, παρέχοντας ένα μεγάλο εύρος πληροφοριών ,
2. **Μίνος, 2010**, σημειώσεις οι οποίες περιέχουν επίσης πάρα πολλές πληροφορίες γύρω από τα είδη των ιχθύων και τη συστηματική τους κατάταξη και
3. **Εμμανουήλ Τόλης**, ο οποίος στα πλαίσια της Πτυχιακής του εργασίας «Σχεδιασμός και υλοποίηση ηλεκτρονικής κλείδας προσδιορισμού ιχθύων με δυνατότητες ενημέρωσης», είχε συγκεντρώσει ένα μεγάλο μέρος αυτών των πληροφοριών.

5 ΠΕΡΙΓΡΑΦΗ - ΑΝΑΛΥΣΗ

Στην παρούσα ενότητα θα αναλυθεί το περιβάλλον της εφαρμογής καθώς και οι δυνατότητες που παρέχει αυτή στο χρήστη, η οποία δημιουργήθηκε με βάση το FAO 37 που περιλαμβάνει την περιοχή της Μεσογείου και της Μαύρης θάλασσας.

Ο FAO «**Food and Agriculture Organization**» είναι μία εξειδικευμένη υπηρεσία του συστήματος των Η.Ε., η οποία έχει ως σκοπό τη βελτίωση του επιπέδου διατροφής των κατοίκων των πιο καθυστερημένων οικονομικά ζωνών, την αποτελεσματικότερη λειτουργία των συστημάτων παραγωγής και διανομής των γεωργικών πόρων και τη βελτίωση των συνθηκών ζωής των αγροτικών πληθυσμών.

Ανοίγοντας την εφαρμογή «Συστηματική ταξινόμηση των Ιχθύων με Χρήση Γραμμωτού Κώδικα», το πρώτο παράθυρο που εμφανίζεται πέρα από τις ονομαστικές πληροφορίες που παρέχει, περιλαμβάνει ακόμα δύο κουμπιά και ένα πεδίο επιλογών (Εικόνας 14).



Εικόνα 14– Παράθυρο εισαγωγής στην εφαρμογή

Το κουμπί **‘Είσοδος’** είναι ανενεργό και δεν επιτρέπει την είσοδο στην εφαρμογή, μιας και προϋποθέτει την επιλογή ενός αριθμού FAO για τον οποίο θα υπάρχει και η αντίστοιχη βάση πληροφοριών.

Το πεδίο επιλογών **‘Επιλογή FAO’**, περιέχει μία λίστα με όλα τα διαθέσιμα FAO για την περιοχή της Μεσογείου και της Μαύρης Θάλασσας. (Εικόνα 15)



Εικόνα 15 – Πεδίο επιλογών FAO

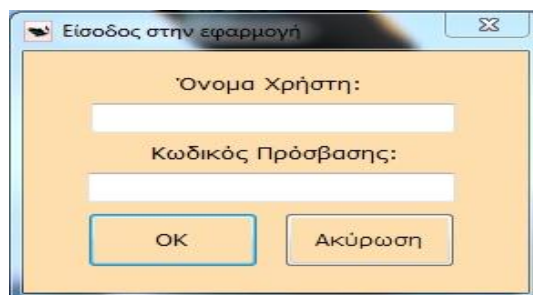
Επιλέγοντας το FAO 37, το κουμπί **‘είσοδος’** ενεργοποιείται ενώ το κουμπί **‘Ρυθμίσεις’** μας παρέχει την δυνατότητα να αλλάξουμε το χρώμα του φόντου της εφαρμογής. (Εικόνα 16)



Εικόνα 16 – Ενεργοποίηση στο κουμπί **‘Είσοδος’** μετά την επιλογή του FAO

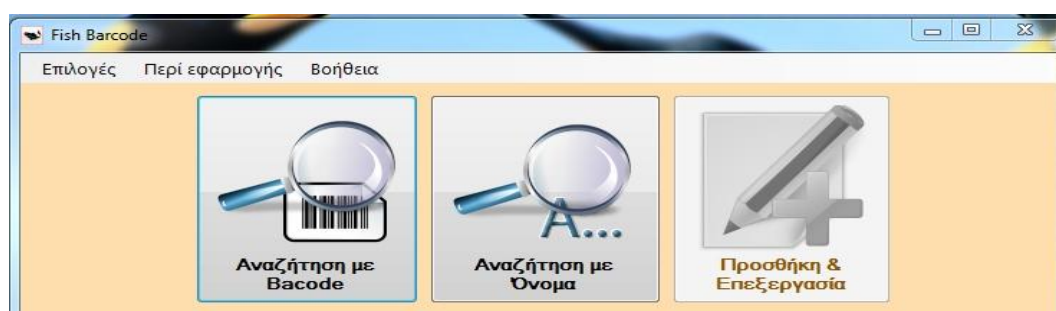
Επιλέγοντας πλέον το ενεργό κουμπί **‘Είσοδος’**, δίνεται η δυνατότητα εισόδου σε μια νέα φόρμα (Εικόνα 4), όπου απαραίτητη προϋπόθεση για το επόμενο βήμα είναι η εισαγωγή των στοιχείων ενός χρήστη, του ονόματος και του κωδικού του. (Εικόνα 17 - Φόρμα εισαγωγής στοιχείων)

Εδώ γίνεται η διάκριση για το είδος του χρήστη. Η εφαρμογή διακρίνει δύο χρήστες, τον διαχειριστή και τον απλό χρήστη. Για κάθε ένα από αυτούς υπάρχουν συγκεκριμένα ονόματα και κωδικοί.



Εικόνα 17 - Φόρμα εισαγωγής στοιχείων

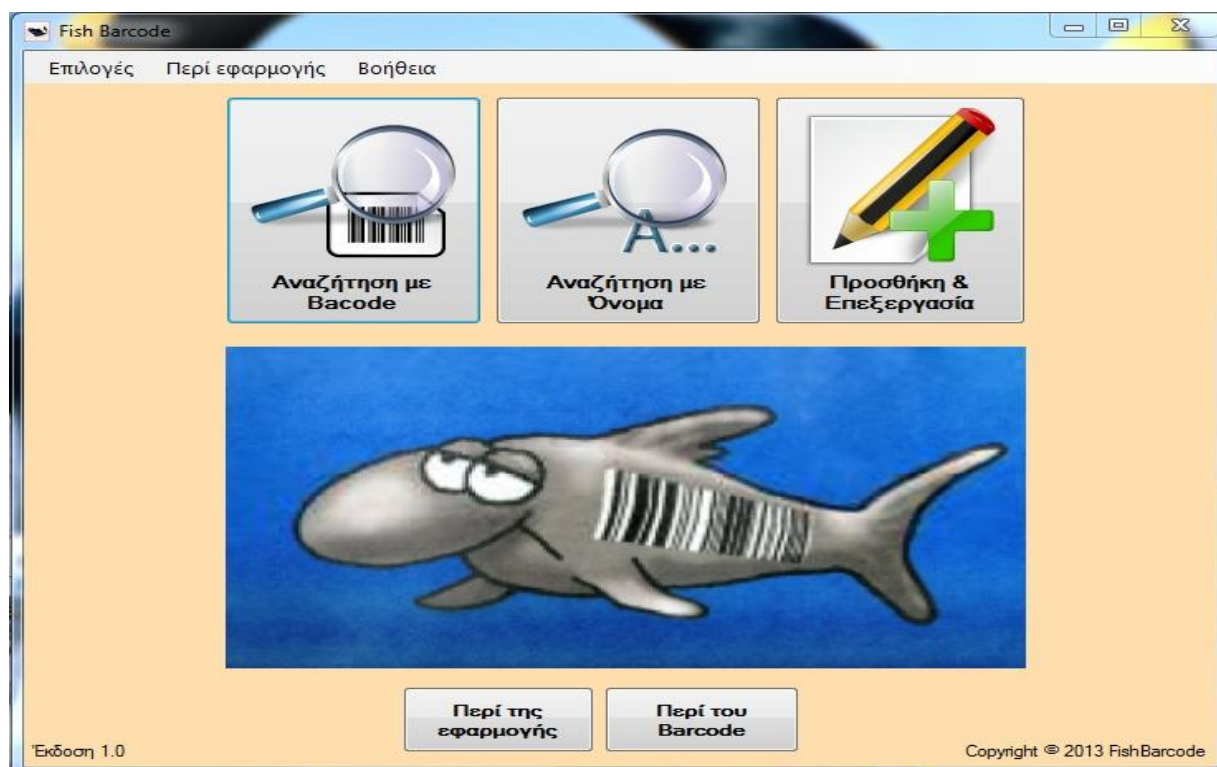
Στον απλό χρήστη δίνεται η δυνατότητα να περιηγηθεί στην εφαρμογή και να βρεί πληροφορίες που αφορούν τη συστηματική και το γραμμωτό κώδικα για κάθε είδος ιχθύος μαζί με εικόνες του είδους και της γεωγραφικής εξάπλωσης. Αυτό γίνεται μέσω των επιλογών **‘Αναζήτηση με βάση το barcode’** και **‘Αναζήτηση με βάση το όνομα’**. (Εικόνα 18 – Επιλογές περιήγησης για τον απλό χρήστη)



Εικόνα 18 – Επιλογές περιήγησης για τον απλό χρήστη

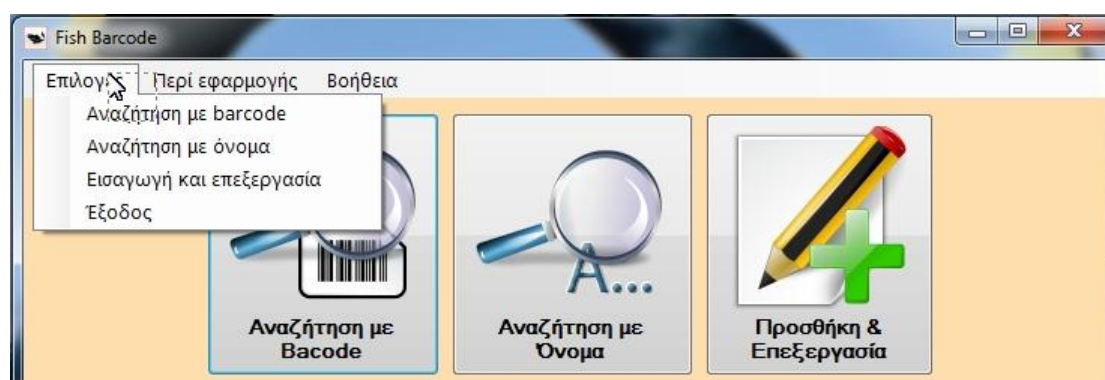
Αντίστοιχα στο διαχειριστή εκτός από τις παραπάνω λειτουργίες, δίνεται και η δυνατότητα προσθήκης ενός νέου είδους και η επεξεργασία των ήδη υπαρχόντων (Εικόνα 19 - Επιλογές περιήγησης για τον διαχειριστή).

Παρακάτω θα αναλυθούν περαιτέρω οι δυνατότητες των επιλογών που αναφέρθηκαν πιο πάνω.



Εικόνα 19 – Επιλογές περιήγησης για τον διαχειριστή

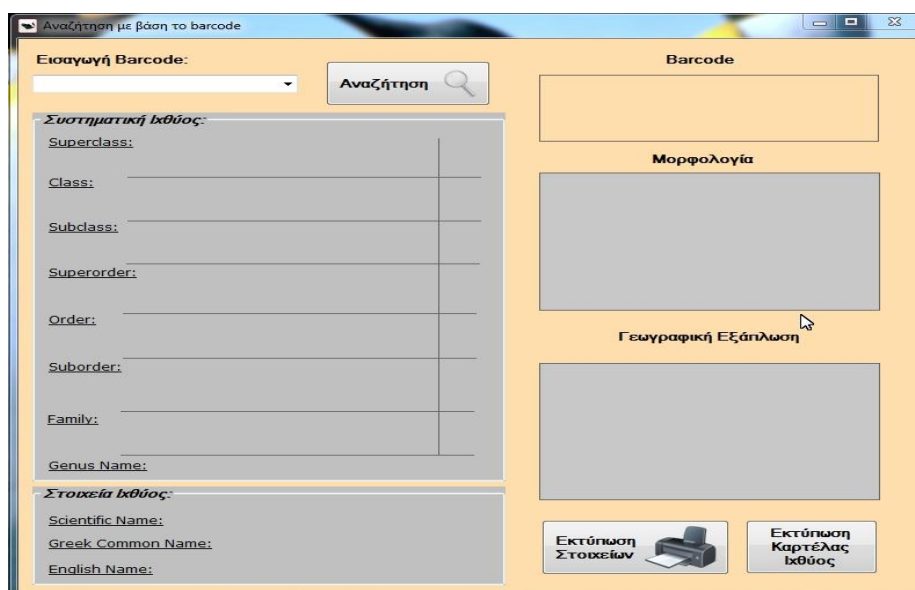
Η εφαρμογή παρέχει την δυνατότητα διπλής εισόδου σε κάθε μία από τις αντίστοιχες λειτουργίες είτε από το αντίστοιχο κουμπί, είτε από την γραμμή επιλογών (Εικόνα 20 - Εισαγωγή στην εφαρμογή από γραμμή επιλογών).



Εικόνα 20 – Εισαγωγή στην εφαρμογή από γραμμή επιλογών

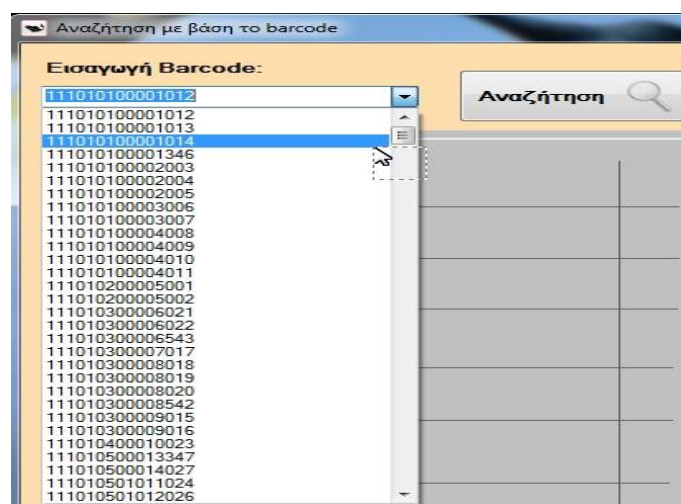
‘Αναζήτηση με βάση το barcode’

Αρχικά θα αναλύσουμε την επιλογή ***‘Αναζήτηση με βάση το barcode’***, η οποία είναι κοινή για τον χρήστη και τον διαχειριστή (Εικόνα 21 - Περιβάλλον εφαρμογής για ‘Αναζήτηση με barcode’).



Εικόνα 21 – Περιβάλλον εφαρμογής για ‘Αναζήτηση με barcode’

Επιλέγοντας λοιπόν να αναζητήσουμε ένα είδος ιχθύος με βάση το barcode, μας ανοίγει το παράθυρο το οποίο αποτελείται από ένα πλαίσιο κειμένου, στο οποίο είτε πληκτρολογούμε την αλληλουχία των ψηφίων που ψάχνουμε, είτε επιλέγουμε ένα από την λίστα προτάσεων που ανοίγει ενεργοποιώντας το πλαίσιο. Αφού πληκτρολογήσουμε ή επιλέξουμε την αλληλουχία των ψηφίων του γραμμωτού κώδικα επιλέγουμε το κουμπί ‘Αναζήτηση’ (Εικόνα 22 - Επιλογή είδους με βάση το barcode και Αναζήτηση).



Εικόνα 22 – Επιλογή είδους με βάση το barcode και Αναζήτηση

Ταυτόχρονα με τη διαδικασία της Αναζήτησης, πραγματοποιείται έλεγχος για το αν υπάρχει η συγκεκριμένη αλληλουχία barcode περασμένη στη βάση και σε περίπτωση λάθους εμφανίζεται το ανάλογο μήνυμα σφάλματος.

Σε αντίθετη περίπτωση το υπάρχον παράθυρο συμπληρώνεται με τις αντίστοιχες πληροφορίες του είδους καθώς και με εικόνες σχετικά με το είδος και τη γεωγραφική του εξάπλωση. Αναλυτικότερα κάτω από το πλαίσιο κειμένου βρίσκονται δύο μεγάλα γκριζα πλαίσια, τα οποία χωρίζονται με μία κάθετη γραμμή.

Στην αριστερή μεριά του πρώτου πλαισίου αναλύονται τα επτά πρώτα επίπεδα της συστηματικής του ιχθύος που επιλέχθηκε,

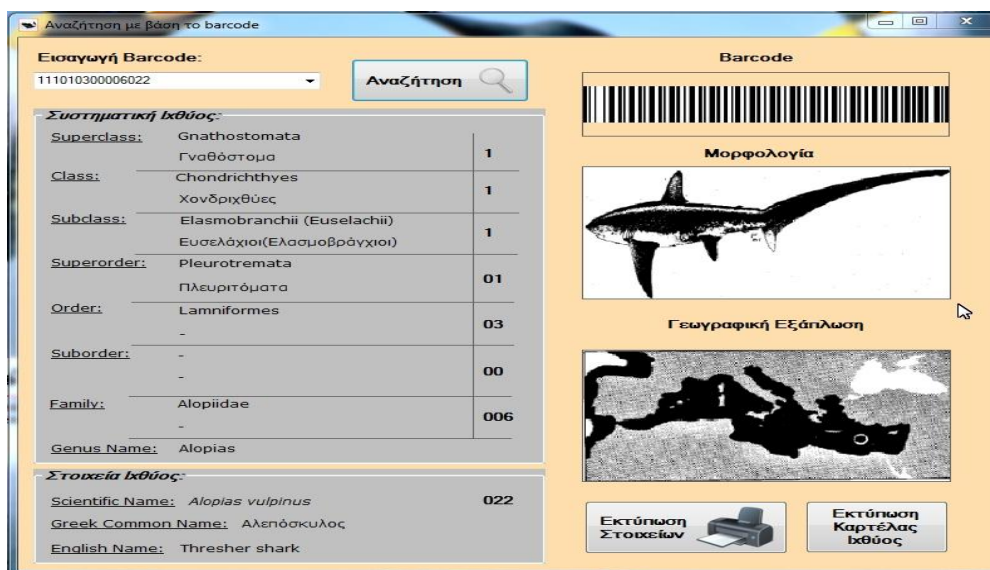
- *υπέρκλαση (superclass),*
- *κλάση (class),*
- *υπέρταξη (superorder),*
- *τάξη (order),*
- *υπόταξη (suborder),*
- *οικογένεια (family)*

ενώ στην δεξιά γίνεται κατακόρυφη ανάλυση της αλληλουχίας των ψηφίων του γραμμωτού κώδικα.

Στο δεύτερο πλαίσιο αναγράφονται τα στοιχεία του ιχθύος και συγκεκριμένα το επιστημονικό όνομα, το κοινό ελληνικό όνομα και το αγγλικό όνομα, ενώ στα δεξιά του πλαισίου αυτού συνεχίζεται η κατακόρυφη ανάλυση της αλληλουχίας των ψηφίων με τα ψηφία που αντιστοιχούν στο είδος.

Στο δεξιό μέρος της οθόνης εμφανίζονται με την αντίστοιχη κατακόρυφη σειρά η εικόνα του γραμμωτού κώδικα, η εικόνα του ιχθύος και η γεωγραφική του εξάπλωση.

Ένα παράδειγμα της ανάλυσης που προηγήθηκε δίνεται στην Εικόνα 23 (Παράδειγμα εφαρμογής με barcode 111010300006022 για αναζήτηση με βάση το barcode), η οποία προέκυψε αναζητώντας το barcode 111010300006022 που αντιστοιχεί στο είδος με επιστημονικό όνομα *alopias vulpinus*.

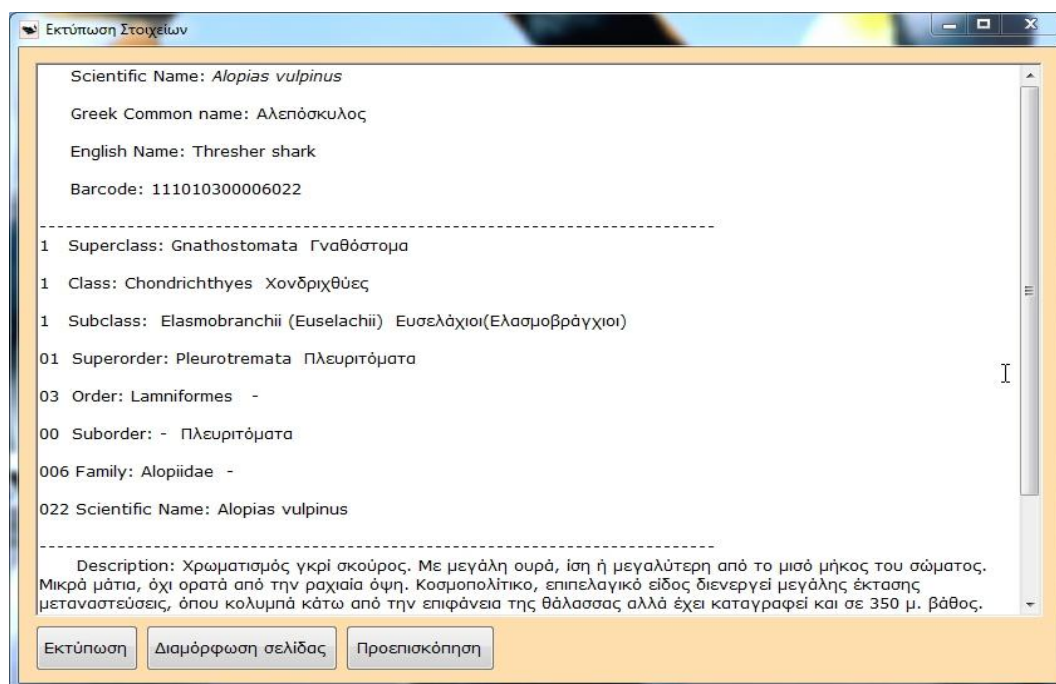


Εικόνα 23 – Παράδειγμα εφαρμογής με barcode 111010300006022 για αναζήτηση με βάση το barcode

Τέλος κάτω από την εικόνα της γεωγραφικής εξάπλωσης εμφανίζονται δύο κουμπία που μας παρέχουν αντίστοιχες δυνατότητες εξαγωγής αποτελεσμάτων.

Το κουμπί **‘Εκτύπωση Στοιχείων’** μας μεταφέρει σε μία συμπληρωμένη φόρμα στοιχείων, τη μορφή της οποίας μπορούμε να δούμε στην Εικόνα 24 (Περιβάλλον εφαρμογής Εκτύπωση Στοιχείων). Η συγκεκριμένη εκτυπώσιμη φόρμα, περιέχει με την ακόλουθη σειρά, τα εξής:

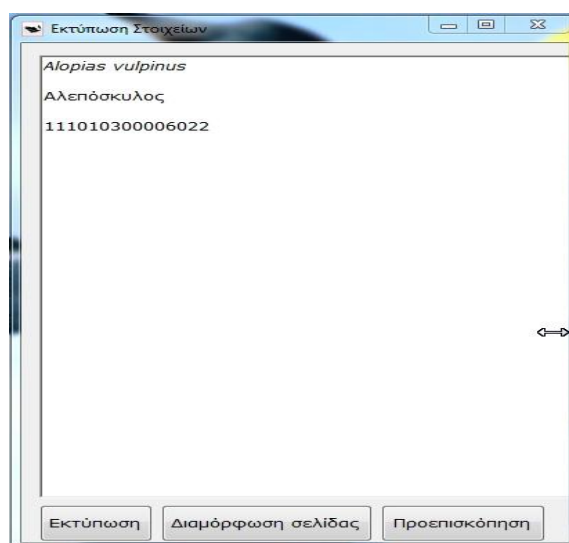
- Το επιστημονικό όνομα του ιχθύος
- Το ελληνικό κοινό όνομα του ιχθύος
- Το αγγλικό όνομα του ιχθύος
- Την αλληλουχία των ψηφίων του barcode
- Ένα πλαίσιο με όλη τη συστηματική του ιχθύος καθώς και την κατακόρυφη ανάλυση της αλληλουχίας των ψηφίων του
- Μία περιγραφή που αντιστοιχεί στο συγκεκριμένο είδος



Εικόνα 24 – Περιβάλλον εφαρμογής Εκτύπωση Στοιχείων

Αντίστοιχα το δεύτερο κουμπί **‘Εκτύπωση Καρτέλας Ιχθύος’** μας μεταφέρει σε μία άλλη συμπληρωμένη εκτυπώσιμη φόρμα, παράδειγμα της οποίας φαίνεται στην Εικόνα 25 (Περιβάλλον εφαρμογής Εκτύπωση Καρτέλας Ιχθύος) και η οποία περιλαμβάνει με την ακόλουθη σειρά τα εξής:

- Το επιστημονικό όνομα του ιχθύος
- Το ελληνικό κοινό όνομα του ιχθύος
- Την αλληλουχία των ψηφίων του barcode

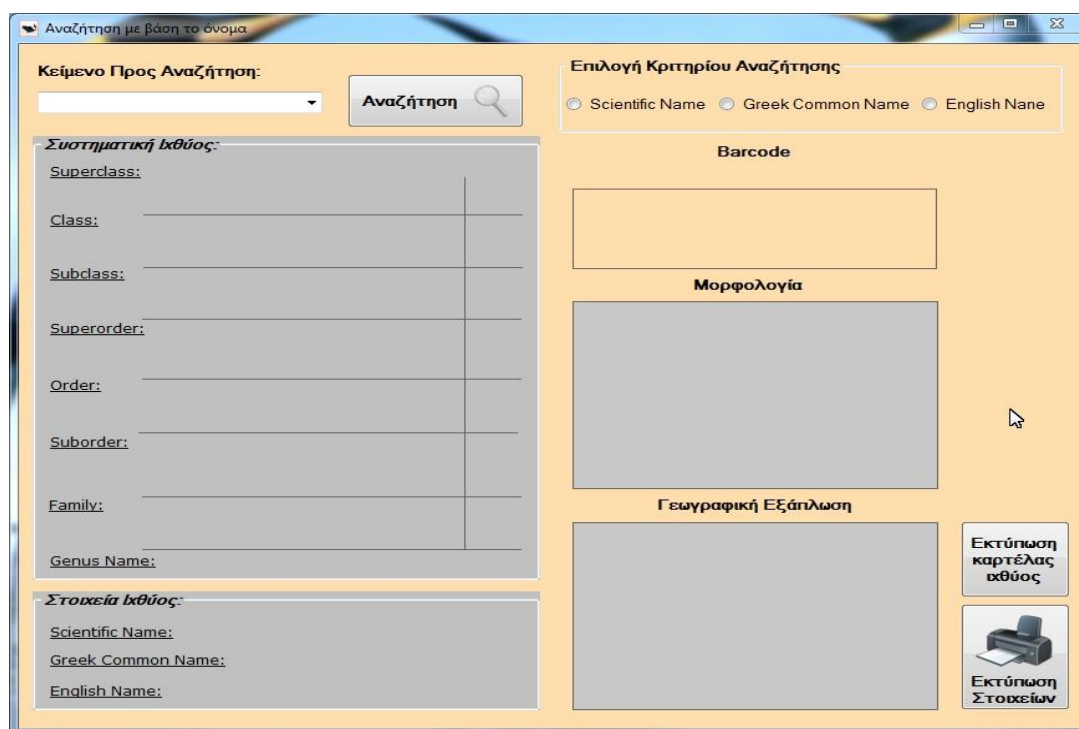


Εικόνα 25 - Περιβάλλον εφαρμογής Εκτύπωση Καρτέλας Ιχθύος

‘Αναζήτηση με βάση το όνομα’

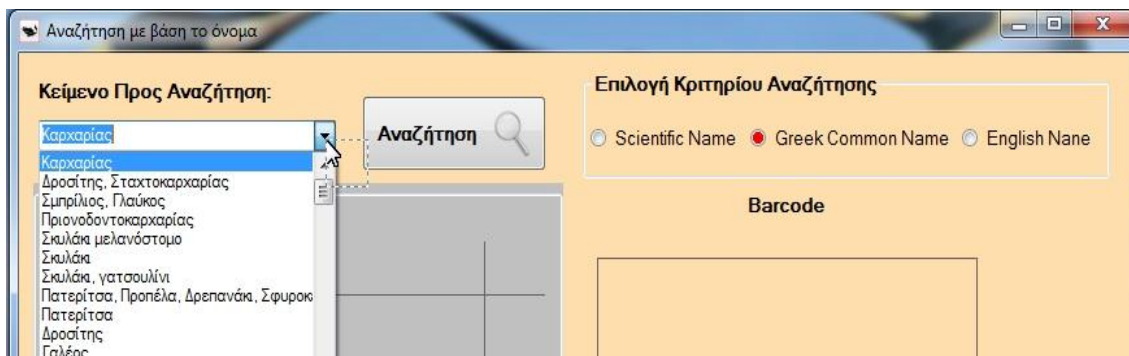
Συνεχίζοντας την περιήγησή μας στην εφαρμογή, επιλέγουμε την δεύτερη λειτουργία της, την ‘Αναζήτηση με βάση το όνομα’.

Επιλέγοντας λοιπόν την συγκεκριμένη λειτουργία, το παράθυρο που εμφανίζεται είναι πανομοιότυπο με το παράθυρο της λειτουργίας ‘Αναζήτηση με barcode’, με την μόνη διαφορά να εστιάζεται στο πλαίσιο επιλογής κριτηρίου, όπως φαίνεται και στην Εικόνα 26 (Περιβάλλον εφαρμογής για ‘Αναζήτηση με βάση το όνομα’).



Εικόνα 26 - Περιβάλλον εφαρμογής για ‘Αναζήτηση με βάση το όνομα’

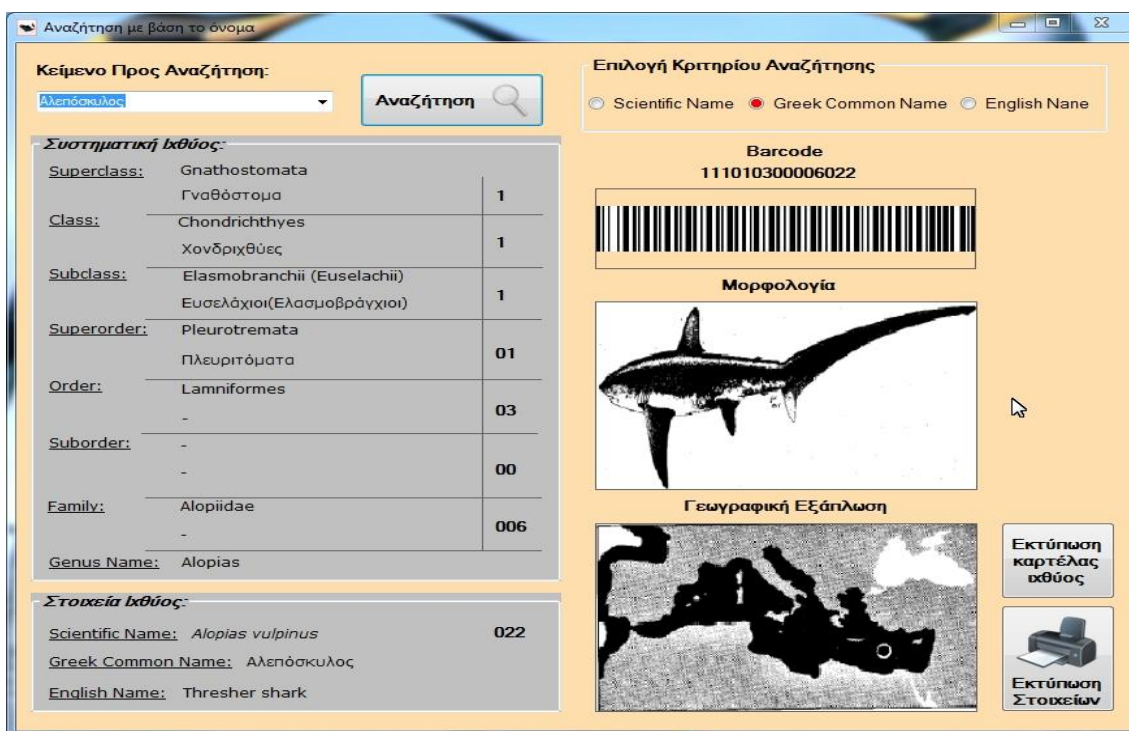
Στο συγκεκριμένο πλαίσιο, ο χρήστης επιλέγει με ποιο από τα τρία ονόματα του ιχθύος- επιστημονικό όνομα, κοινό ελληνικό όνομα, αγγλικό όνομα- με το οποίο θέλει να κάνει την αναζήτηση. Για κάθε ένα από τα τρία κριτήρια αναζήτησης παρέχονται στο χρήστη οι αντίστοιχες προτάσεις επιλογών (Εικόνα 27 - Προτάσεις επιλογών για το κριτήριο Greek Common Name (κοινό ελληνικό όνομα)).



Εικόνα 27 – Προτάσεις επιλογών για το κριτήριο Greek Common Name (κοινό ελληνικό όνομα)

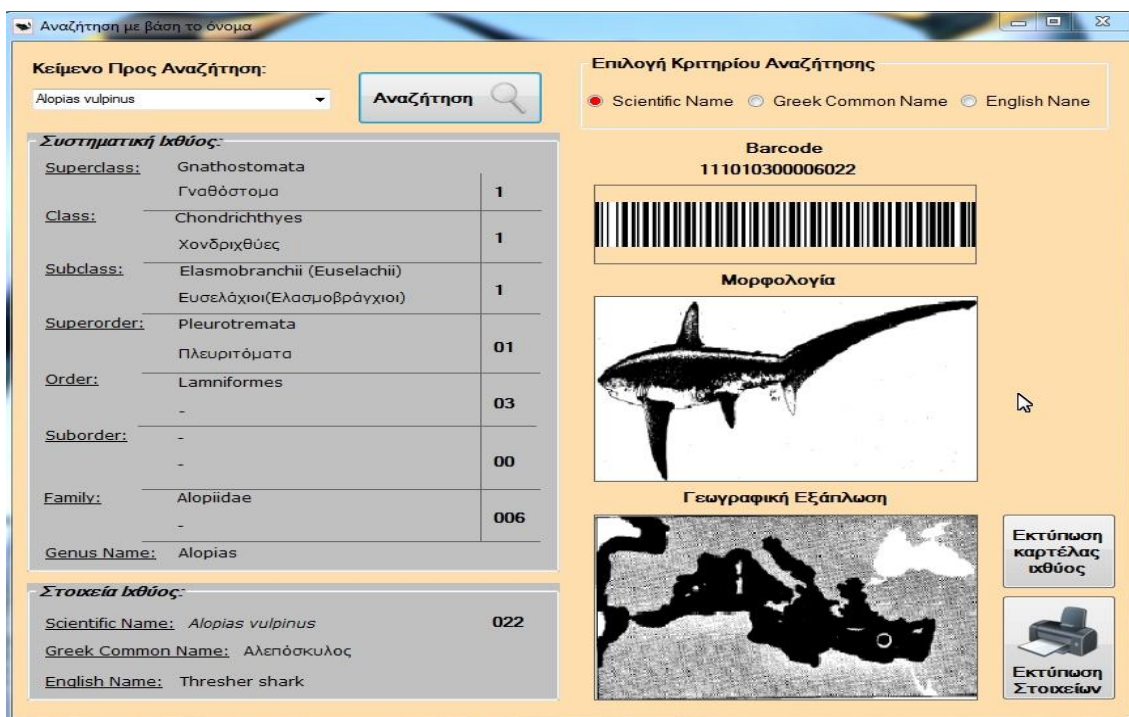
Παρακάτω, στα πλαίσια εφαρμογής-παραδείγματος, πραγματοποιήθηκε αναζήτηση, με κριτήριο αναζήτησης το κοινό ελληνικό όνομα αλεπόσκυλος (Εικόνα 28 - Παράδειγμα εφαρμογής με κριτήριο Greek Common Name (κοινό ελληνικό όνομα) και όνομα αλεπόσκυλος για αναζήτηση με βάση το όνομα).

Ανατρέχοντας στη Εικόνα 25 διαπιστώνουμε ότι τα στοιχεία – πληροφορίες που μας παρέχουν και οι δύο αυτές λειτουργίες είναι ακριβώς ίδια.



Εικόνα 28 - Παράδειγμα εφαρμογής με κριτήριο Greek Common Name (κοινό ελληνικό όνομα) και όνομα αλεπόσκυλος για αναζήτηση με βάση το όνομα

Στο ίδιο ακριβώς συμπέρασμα καταλήγουμε κάνοντας την αναζήτηση με κριτήριο το επιστημονικό όνομα του ιχθύος Alopias vulpinus (Εικόνα 28 - Παράδειγμα εφαρμογής με κριτήριο Greek Common Name (κοινό ελληνικό όνομα) και όνομα αλεπόσκυλος για αναζήτηση με βάση το όνομα).



Εικόνα 29 - Παράδειγμα εφαρμογής με κριτήριο Scientific Name (επιστημονικό όνομα) και όνομα *Alopias vulpinus* για αναζήτηση με βάση το όνομα

Αντίστοιχα οι φόρμες 'Εκτύπωση στοιχείων' και 'Εκτύπωση καρτέλας' είναι οι ίδιες που εμφανίζονται στην Εικόνα 24 και στην Εικόνα 25.

‘Προσθήκη είδους’ και ‘Επεξεργασία Είδους’

Η επόμενη λειτουργία της εφαρμογής είναι αποκλειστικά για τον διαχειριστή της εφαρμογής και αυτό γιατί αναφέρεται είτε στην επεξεργασία ενός ήδη υπάρχον ιχθύος είτε στην προσθήκη ενός νέου.

Στην συγκεκριμένη λειτουργία αποφασίστηκε να μη δοθούν δυνατότητες διαγραφής, αφού δεν υπάρχει περίπτωση να εξαληφθεί ένα ολοκλήρο είδος πότε.

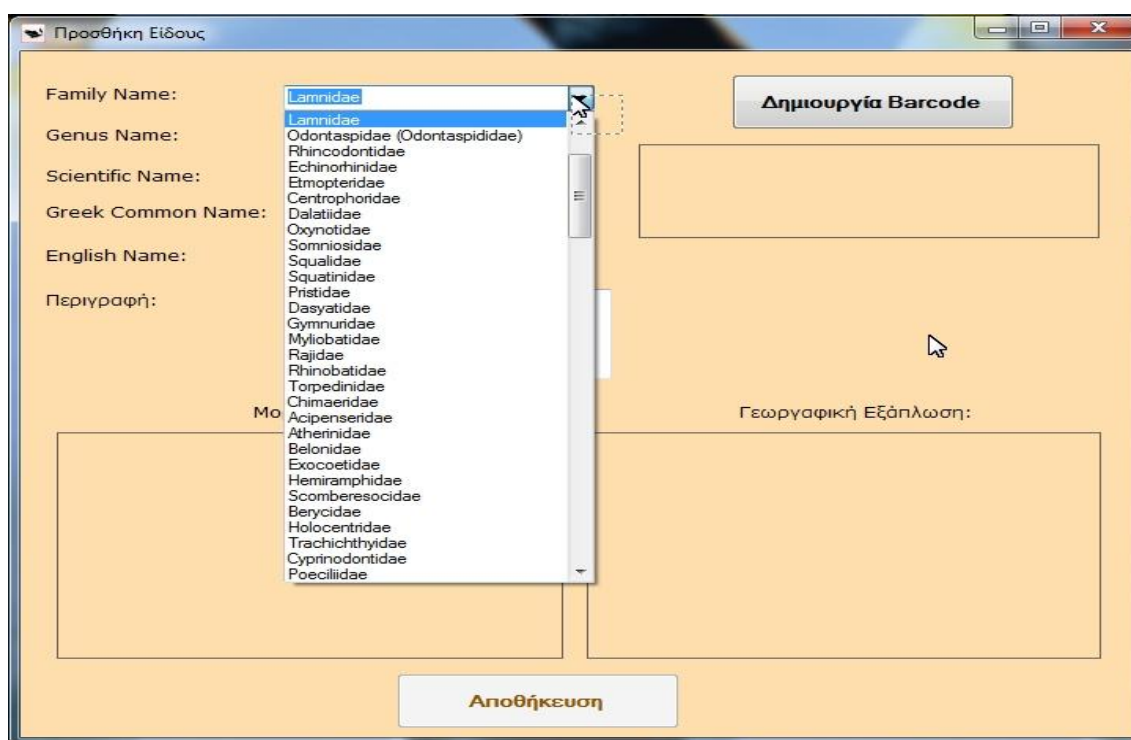
Επιλέγοντας το κουμπί **‘Προσθήκη και Επεξεργασία’**, ανοίγει ένα βοηθητικό παράθυρο, το οποίο διαχωρίζει τις δύο λειτουργίες μέσω των αντίστοιχων κουμπιών **‘Προσθήκη είδους’** και **‘Επεξεργασία Είδους’** (Εικόνα 30 - Περιβάλλον εφαρμογής Προσθήκη-Επεξεργασία).



Εικόνα 30 – Περιβάλλον εφαρμογής Προσθήκη-Επεξεργασία

Επιλέγοντας το κουμπί ‘Προσθήκη Είδους’, ανοίγει το αντίστοιχο παράθυρο της προσθήκης ενός νέου είδους, η οποία αποτελείται από εξής ακόλουθα στοιχεία:

- Μία φόρμα συμπλήρωσης των στοιχείων family name, genus name, scientific name, greek common name, English name και την περιγραφή.
- Ένα κουμπί για την δημιουργία του γραμμωτού κώδικα.
- Ένα πλαίσιο, το οποίο βρίσκεται κάτω ακριβώς από το κουμπί ‘Δημιουργία Barcode’, στο οποίο εμφανίζεται η εικόνα του γραμμωτού κώδικα.
- Ένα πλαίσιο μέσα στο οποίο θα προστεθεί θα προστεθεί η εικόνα που αφορά τη μορφολογία του ιχθύος
- Ένα πλαίσιο μέσα στο οποίο θα προστεθεί θα προστεθεί η εικόνα που αφορά τη γεωγραφική εξάπλωση του ιχθύος.
- Ένα δεύτερο κουμπί, το οποίο ονομάζεται ‘Αποθήκευση’, το οποίο ενεργοποιείται αφού επιλεγεί πρώτα το κουμπί ‘Δημιουργία Barcode’.



Εικόνα 31 – Περιβάλλον εφαρμογής για την προσθήκη νέου είδους

Για να γίνει μία προσθήκη είδους πρέπει αρχικά ο διαχειριστής να επιλέξει, μέσω της λίστας επιλογών που του προτείνεται, το όνομα της οικογένειας του ιχθύος. Μόνο με το όνομα της οικογένειας του ιχθύος η εφαρμογή μπορεί και σου δημιουργεί μία μοναδική αλληλουχία 15 ψηφίων.

Μόλις ο διαχειριστής επιλέξει το κουμπί **‘Δημιουργία Barcode’**, στο πλαίσιο που βρίσκεται κάτω ακριβώς από το κουμπί εμφανίζεται η εικόνα του γραμμωτού κώδικα, ενώ ακριβώς κάτω από το πλαίσιο εμφανίζεται η αλληλουχία των 15 ψηφίων, όπως φαίνεται και στο παράδειγμα της Εικόνας 32 (Δημιουργία barcode μόλις δοθεί το όνομα της οικογένειας που ανήκει (Family Name)).



Εικόνα 32 – Δημιουργία barcode μόλις δοθεί το όνομα της οικογένειας που ανήκει (Family Name)

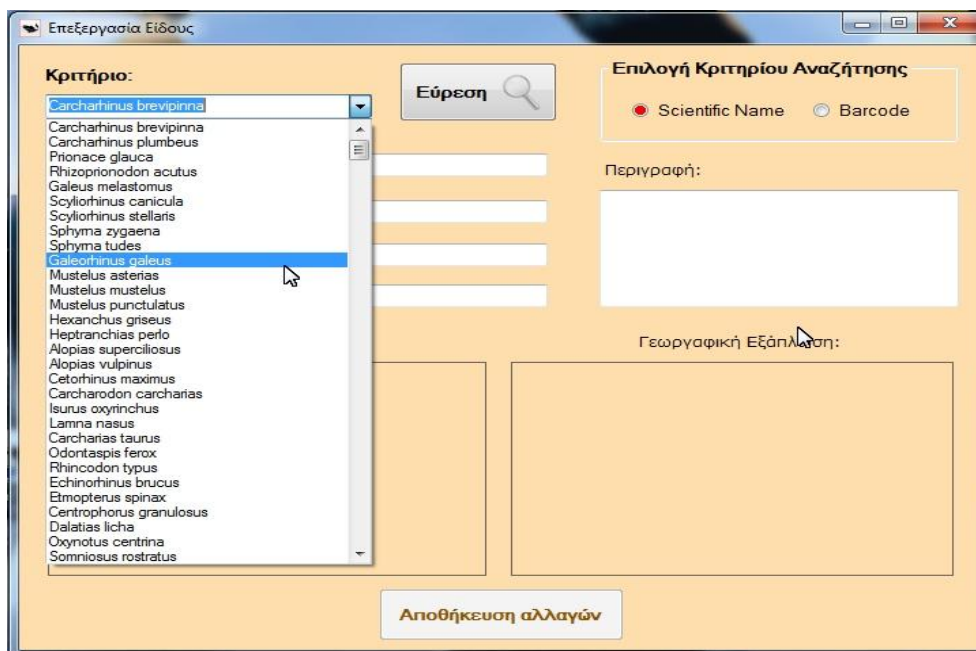
Για να ολοκληρωθεί όμως η αποθήκευση πρέπει ο διαχειριστής να συμπληρώσει τα αντίστοιχα πεδία πληροφοριών που αναφέρθηκαν πιο πάνω.

Τέλος ο διαχειριστής, με ένα click πάνω στο πλαίσιο είτε της μορφολογίας, είτε της γεωγραφικής εξάπλωσης μπορεί να προσθέσει τις αντίστοιχες φωτογραφίες.

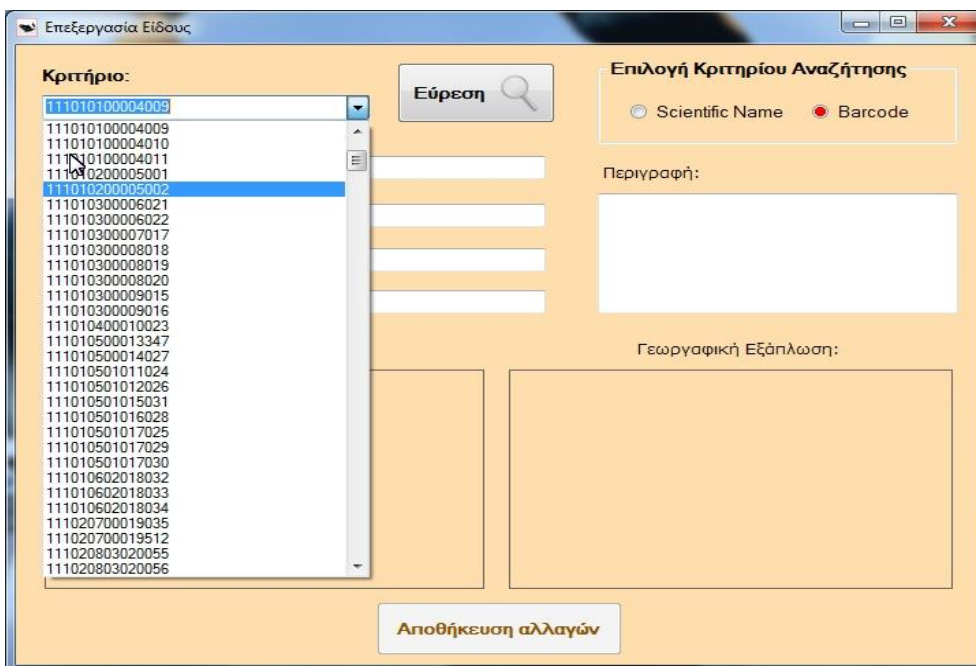
Επιλέγοντας το κουμπί **‘Επεξεργασία Είδους’**, ανοίγει το αντίστοιχο παράθυρο της επεξεργασία ενός ήδη υπάρχοντος ιχθύος, η οποία αποτελείται από εξής ακόλουθα στοιχεία:

- Ένα πλαίσιο επιλογής κριτηρίου, το οποίο ονομάζεται **‘Επιλογή κριτηρίου αναζήτησης’** και δίνει την δυνατότητα στο διαχειριστή να επιλέξει σαν κριτήριο αναζήτησης είτε το **scientific name** του ιχθύος, είτε το **‘Barcode’**.
- Ένα πλαίσιο αναζήτησης, στο οποίο ανάλογα με το κριτήριο που επιλέγεται, εμφανίζει και τις αντίστοιχες προτεινόμενες λίστες τόσο για το **scientific name** (Εικόνα 33 - Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους σύμφωνα με το Scientific Name (επιστημονικό όνομα)) όσο και για το **barcode** (Εικόνα 34 - Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους σύμφωνα με το barcode)
- Ένα κουμπί, το οποίο ονομάζεται **‘Εύρεση’**.
- Μία φόρμα επεξεργασίας των ήδη υπάρχων στοιχείων, που αφορούν το family name, το genus name, το scientific name, το greek common name, το english name και την περιγραφή, στα οποία ο διαχειριστής μπορεί να κάνει όποια αλλαγή επιθυμεί.
- Ένα πλαίσιο μέσα στο οποίο υπάρχει η εικόνα που αφορά τη μορφολογία του ιχθύος

- Ένα πλαίσιο μέσα στο οποίο υπάρχει η εικόνα που αφορά τη γεωγραφική εξάπλωση του ιχθύος.
- Ένα δεύτερο κουμπί, το οποίο ονομάζεται **‘Αποθήκευση’**, το οποίο ενεργοποιείται αφού επιλεγεί πρώτα το κουμπί **‘Δημιουργία Barcode’**.



Εικόνα 33 – Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους σύμφωνα με το Scientific Name (επιστημονικό όνομα)



Εικόνα 34 – Περιβάλλον εφαρμογής για την Επεξεργασία του Είδους σύμφωνα με το barcode

6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια της έρευνας που έγινε για την περαιώση αυτής της πτυχιακής εργασίας, δημιουργήθηκε η ανάγκη πλήρους κατανόησης της λειτουργίας τόσο του γραμμωτού κώδικα όσο και της συστηματικής ταξινόμησης των ιχθύων. Ο γραμμωτός κώδικας που δημιουργήθηκε κωδικοποιεί σε ένα δεκαπενταψήφιο αριθμό όλη την συστηματική ταξινόμηση των ιχθύων. Για κάθε έναν ιχθύ δίνεται ένας μοναδικός δεκαπενταψήφιος αριθμός που περιέχει όλη την παραπάνω πληροφορία.

Η εφαρμογή δεν προορίζεται για εμπορική χρήση αλλά είναι μια εφαρμογή που προσφέρει τα εξής ακόλουθα πλεονεκτήματα. Κατά πρώτων, συνδέει την λειτουργία του γραμμωτού κώδικα με την συστηματική ταξινόμηση των ιχθύων. Αν και το σύστημα κωδικοποίησης που επιλέχτηκε δεν ανταποκρίνεται στα εμπορικά πρότυπα, ο σχεδιασμός της και η υλοποίηση της παρέχει μία έτοιμη και ευέλικτη βάση δεδομένων και ένα μεγάλο εύρος αναβαθμίσεων. Οι δύο κυριότερες από αυτές θα μπορούσαν να είναι :

1. Η συγχώνευση των ήδη υπαρχών ψηφίων σε πολύ λιγότερα. Παραδείγματος χάρη θα μπορούσαν οι δεκαπενταψήφιοι αριθμοί να γίνουν τριψήφιοι, αλλά χωρίς να αλλοιωθεί η πληροφορία που ήδη περιέχουν. Με αυτόν τον τρόπο θα μπορούν να αποτελούν από μόνα τους ένα πεδίο σε οποιοδήποτε εμπορικό σύστημα γραμμωτού κώδικα.
2. Για κάθε μία γεωγραφική περιοχή, δηλαδή για κάθε FAO, μπορεί να δημιουργηθεί μία αντίστοιχη βάση δεδομένων και με μικρές τροποποιήσεις στον πηγαίο κώδικα να δημιουργηθούν οι αντίστοιχοι γραμμωτοί κώδικες. Για τον λόγο αυτό, η αρχική φόρμα της εφαρμογής ζητάει την επιλογή ενός FAO. Η επιλογή να μην ενσωματωθούν όλα μαζί σε μία βάση πάρθηκε με γνώμονα το γεγονός ότι ένα τέτοιο εγχείρημα απαιτούσε την διαχείριση τεράστιου όγκου πληροφορίας, το οποίο η εφαρμογή δεν το απαιτούσε.

Πολύ σημαντικό ρόλο στη δημιουργία αυτής της εφαρμογής είχε η βάση δεδομένων που δημιουργήθηκε. Μέσα στη βάση δημιουργήθηκε μία πλήρης δενδροειδής μορφή, παρόλο που η MySQL δεν υποστηρίζει αντικειμενοστρέφια. Αυτή η καινοτομία στη δημιουργία της βάσης, αποτελεί και καινοτομία γενικότερα της εφαρμογής.

Στην εφαρμογή, οι δύο βασικές λειτουργίες είναι η αναζήτηση με βάση το barcode, στην οποία ο χρήστης δίνει το barcode και παίρνει σαν έξοδο τις πληροφορίες του ιχθύ με το συγκεκριμένο barcode και η αναζήτηση με βάση το όνομα, στην οποία ο χρήστης δίνει ένα από τα ονόματα του ιχθύ και παίρνει σαν έξοδο το barcode. Όμως και στις δύο αυτές λειτουργίες ο χρήστης έχει τη δυνατότητα να δει την αντιστοιχία του barcode με την αντίστοιχη συστηματική ταξινόμηση, χρήσιμες

περιγραφές αλλά και εικόνες με την μορφολογία του κάθε ιχθύ και την συστηματική ταξινόμηση του. Εκτός των εικόνων για όλα τα υπόλοιπα στοιχεία παρέχεται η δυνατότητα εκτύπωσης.

Ακόμα, η εφαρμογή παρέχει δυνατότητες επεξεργασίας των ήδη υπάρχων εγγραφών και την προσθήκη νέων εγγραφών, οι οποίες όμως μπορεί να τις υλοποιήσει μόνο ο διαχειριστής της εφαρμογής. Η εφαρμογή διαχωρίζει δύο ειδών χρήστες, τον διαχειριστή και τον απλό χρήστη. Στο υπάρχων επίπεδο της εφαρμογής σαν διαχειριστής μπορεί να ενεργήσει οποιοσδήποτε χρήστης με μόνη προϋπόθεση, στην φόρμα Login, να κάνει είσοδο σαν διαχειριστής με τον αντίστοιχο κωδικό.

Η εφαρμογή δεν παρέχει δυνατότητα διαγραφής κάποιου είδους, αφού ακόμα και αν εκλείψει παραμένει ένα μουσειακό είδος. Στη λειτουργία της επεξεργασίας μπορούν να τροποποιηθούν μόνο τα ονόματα, οι εικόνες και η περιγραφή. Στην εισαγωγή εισάγεις ή επιλέγεις από την λίστα προτάσεων, μόνο το όνομα της οικογένειας, το οποίο είναι αρκετό για την εύρεση της υπόλοιπης συστηματικής ταξινόμησης. Εκτός από την οικογένεια μπορείς να προσθέσεις τα ονόματα, τις εικόνες και την περιγραφή. Γενικά, είναι μία εφαρμογή που μπορεί να δώσει ένα πλήθος πληροφοριών σε οποιοδήποτε χρήστη επιλέξει να περιηγηθεί σε αυτή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Κασπίρης, Π. (2000), Τα ψάρια της Ελλάδας (Κλείδες προσδιορισμού), ΤΥΠOffset Κ. Μανούδη-Στανίδη, Πάτρα.

Μίνος, Γ. (2010), Σημειώσεις Μαθήματος "Βιολογία & Συστηματική Ιχθύων. Τεύχος Δεύτερο". ΑΤΕΙ Θεσσαλονίκης, Παράρτημα Ν. Μουδανιών, Τμήμα Τεχνολογίας Αλιείας & Υδατοκαλλιεργειών.

Ramakrishnan R., Gehrke J. (2002), Συστήματα Διαχείρισης Βάσεων Δεδομένων, 2η έκδοση, Α' Τόμος, εκδόσεις "Τζιόλα" (Μεταφρασμένο).

Ridgeway M. (2008), Visual Basic .NET Developer's Handbook - Mark Ridgeway, εκδόσεις "Sybex" (Μεταφρασμένο).

Πετρούτσος Ε. (2008), Mastering Microsoft Visual Basic 2008, εκδόσεις "Sybex".

Τόλης Μ., Μίνος Γ. & Κόστογλου Β. (2011), Σχεδιασμός και υλοποίηση ηλεκτρονικής κλείδας προσδιορισμού ιχθύων με δυνατότητες ενημέρωσης, Πτυχιακή Εργασία Α.Τ.Ε.Ι. Θεσσαλονίκης

ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Nelson, J. S. (1994), Fishes of the world. 3rd edition. John Wiley & Sons, Inc., New York.

Nelson, J. S. (2006). Fishes of the world. 4th edition. John Wiley & Sons, Inc. Hoboken, New Jersey, USA.

ΔΗΜΟΣΙΕΥΣΕΙΣ

Άρθρο, Mary Bellis, What is barcode? The history of barcode, Διαθέσιμο στο:
<http://inventors.about.com/od/bstartinventions/a/Bar-Codes.htm>

Άρθρο, Marshall Brain, How UPC Bar Codes Work, Διαθέσιμο στο:
<http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/upc.htm>

ΣΥΝΔΕΣΜΟΙ ΔΙΑΔΙΚΤΥΟΥ

MySQL, Από Βικιπαίδεια, ελεύθερη εγκυκλοπαίδεια, Διαθέσιμο στο:
<http://en.wikipedia.org/wiki/MySQL>

Barcode, Από Βικιπαίδεια, ελεύθερη εγκυκλοπαίδεια, Διαθέσιμο στο:
<http://en.wikipedia.org/wiki/Barcode>

Species, Από Βικιπαίδεια, ελεύθερη εγκυκλοπαίδεια, Διαθέσιμο στο:

<http://en.wikipedia.org/wiki/Species>

Microsoft Visual Studio, Από Βικιπαίδεια, ελεύθερη εγκυκλοπαίδεια, Διαθέσιμο στο:

http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

Visual Basic .NET 2010 Express - A "From the Ground Up" Tutorial Διαθέσιμο στο:

http://visualbasic.about.com/od/learnvbnet/a/LVBE_L1.htm

7 Visual Basic .NET Programming for Beginners Διαθέσιμο στο:

<http://www.homeandlearn.co.uk/net/vbNet.html>

8 Visual Basic 2010 Express Tutorial – VB.Net, Διαθέσιμο στο:

http://www.schoolfreeware.com/Visual_Basic_2010_Express_Tutorials_-_VB.Net_-_Beginning_-_Game_Programming.html

9 Visual Basic 2010 Διαθέσιμο στο: <http://www.vbtutor.net/vb2010/>

Θεοδώρου Αυτοματισμοί ABETE, Διαθέσιμο στο: <http://www.theodorou.gr/>

520BarcodeHellas, Διαχείριση διεθνών προτύπων σύμανσης, Διαθέσιμο στο:

<http://www.520barcodehellas.com/>

ΠΑΡΑΡΤΗΜΑ

<p style="text-align: center;">Παράρτημα Α ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΦΟΡΜΑΣ MENUFORM.VB</p>

Public Class MenuForm

```
Private Sub MainForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    LoginForm.Close()
End Sub
```

```
Private Sub btnBcodeSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBcodeSearch.Click
    BarcodeSearch.Show()
End Sub
```

```
Private Sub btnNameSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNameSearch.Click
    NameSearch.Show()
End Sub
```

```
Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click
    MenuForm2.Show()
End Sub
Private Sub BarcodeToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles BarcodeToolStripMenuItem.Click
    BarcodeSearch.Show()
End Sub
```

```
Private Sub NameToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles NameToolStripMenuItem.Click
    NameSearch.Show()
End Sub
```

```
Private Sub VersionToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles VersionToolStripMenuItem.Click
    AboutAdd.Show()
End Sub
```

```
Private Sub ExitToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles ExitToolStripMenuItem.Click
    Application.Exit()
End Sub
```

```
Private Sub EditToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles EditToolStripMenuItem.Click
    MenuForm2.Show()
End Sub
```

```
Private Sub AboutAppToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs)
Handles AboutAppToolStripMenuItem.Click
    AboutBS.Show()
End Sub
```

```
Private Sub AboutBarcodeToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs)
Handles AboutBarcodeToolStripMenuItem.Click
    AboutNS.Show()
End Sub
```

```
Private Sub btnAboutApp_Click(sender As System.Object, e As System.EventArgs) Handles
btnAboutApp.Click
    AboutApp.Show()
End Sub
```

```
Private Sub btnAboutBcode_Click(sender As System.Object, e As System.EventArgs) Handles
btnAboutBcode.Click
    AboutBcode.Show()
End Sub
```

```
Private Sub ΠεριγραφήΕφαρμογήςToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    AboutApp.Show()
End Sub
```

```
Private Sub ΓενικάΓιαΤοBarcodeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    AboutBcode.Show()
End Sub
End Class
```

Παράρτημα Β ΚΩΔΙΚΑΣ BUTTON1_CLICK
--

1^ο μέρος

```
Dim maxidnum As Integer
Dim spid As Integer
Dim fn As New AutoCompleteStringCollection

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnGenBcode.Click
    If txtFamily.Text.Length > 0 And txtScient.Text.Length > 0 Then

        Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
        Dim SQLDataAdapter As MySqlDataAdapter
        Dim SQLDataSet As DataSet
        Dim strSQL As String

        strSQL = "SELECT t1.num_bar AS lev1bc, " & _
            " t2.num_bar AS lev2bc, " & _
            " t3.num_bar AS lev3bc, " & _
            " t4.num_bar AS lev4bc, " & _
            " t5.num_bar AS lev5bc, " & _
            " t6.num_bar AS lev6bc, " & _
            " t7.num_bar AS lev7bc " & _
```

```
" FROM dbtree1 AS t1 " & _  
" LEFT JOIN dbtree1 AS t2 ON t2.parent_id = t1.id " & _  
" LEFT JOIN dbtree1 AS t3 ON t3.parent_id = t2.id " & _  
" LEFT JOIN dbtree1 AS t4 ON t4.parent_id = t3.id " & _  
" LEFT JOIN dbtree1 AS t5 ON t5.parent_id = t4.id " & _  
" LEFT JOIN dbtree1 AS t6 ON t6.parent_id = t5.id " & _  
" LEFT JOIN dbtree1 AS t7 ON t7.parent_id = t6.id " & _  
" WHERE t7.id IN (Select family_id FROM species WHERE family_name = " & txtFamily.Text & " )"  
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

2^ο μέρος:

```
Dim bcode, thesuperclass, theclass, thesubclass, thesuperorder, theorder, thesuborder, thefamily As String
```

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows
```

```
thesuperclass = row.Item("lev1bc")  
theclass = row.Item("lev2bc")  
thesubclass = row.Item("lev3bc")  
thesuperorder = row.Item("lev4bc")  
If thesuperorder.Length < 2 Then  
thesuperorder = "0" & thesuperorder  
End If
```

```
theorder = row.Item("lev5bc")  
If theorder.Length < 2 Then  
theorder = "0" & theorder  
End If
```

```
thesuborder = row.Item("lev6bc")  
If thesuborder.Length < 2 Then  
thesuborder = "0" & thesuborder  
End If
```

```
thefamily = row.Item("lev7bc")  
If thefamily.Length = 2 Then  
thefamily = "0" & thefamily  
Elseif thefamily.Length = 1 Then  
thefamily = "00" & thefamily  
End If
```

```
Next
```

```
bcode = thesuperclass & theclass & thesubclass & thesuperorder & theorder & thesuborder & thefamily
```

```
strSQL = "Select family_id FROM species WHERE family_name = " & txtFamily.Text & " "
```

```
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```


3^ο μέρος

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows  
    spid = row.Item("family_id")
```

```
Next
```

```
strSQL = "SELECT MAX(species_id) FROM species"
```

```
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

```
Dim maxid As String
```

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows  
    maxid = row.Item("max(species_id)")  
    maxidnum = Integer.Parse(maxid)  
    maxidnum += 1
```

```
Next
```

```
bcode = bcode & maxidnum.ToString  
bcodeImage.Image = Code128Rendering.MakeBarcodeImage(bcode, 2, False)  
bcodeLbl.Text = bcode  
bcodeLbl.Visible = True
```

```
btnSaveSpecies.Enabled = True
```

```
Else
```

```
    MessageBox.Show("Θα πρέπει να εισάγετε τουλάχιστο Family Name και Scientific Name!",  
"Προειδοποίηση!", MessageBoxButtons.OK, MessageBoxIcon.Warning)
```

```
End If
```

```
End Sub
```

```
.  
.
```

Παράρτημα Γ

**ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ PICTUREBOX1_CLICK, PICTUREBOX2_CLICK
ΚΑΙ BTNSAVESPECIES_CLICK**

```
.  
.  
Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles  
PictureBox1.Click
```

```
    OpenFileDialog1.ShowDialog()
```

```
End Sub
```

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

```
    Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user  
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
```

```
    Dim sqlCommand As New MySqlCommand
```

```
    Dim strSQL As String
```

```
    Dim memImage As New System.IO.MemoryStream
```

```
Dim bytImage() As Byte

Dim memImage2 As New System.IO.MemoryStream
Dim bytImage2() As Byte

If Not IsNothing(PictureBox1.Image) Then
    PictureBox1.Image.Save(memImage, PictureBox1.Image.RawFormat)
    bytImage = memImage.GetBuffer()
End If

If Not IsNothing(PictureBox2.Image) Then
    PictureBox2.Image.Save(memImage2, PictureBox2.Image.RawFormat)
    bytImage2 = memImage2.GetBuffer()
End If
strSQL = "INSERT INTO species VALUES (" & maxidnum & ", " & txtScient.Text & ", " &
txtGrcommon.Text & ", " & txtEng.Text & ", " & txtFamily.Text & ", " & spid & ", " & txtGenus.Text & ", " &
txtdesc.Text & ", " & bcodelbl.Text & ", @data , @data2)"
sqlCommand.Connection = m_DatabaseConnection
sqlCommand.CommandText = strSQL
sqlCommand.Parameters.AddWithValue("@data", bytImage)
sqlCommand.Parameters.AddWithValue("@data2", bytImage2)
m_DatabaseConnection.Open()
sqlCommand.ExecuteNonQuery()

m_DatabaseConnection.Close()

MessageBox.Show("Το είδος αποθηκεύτηκε!")

End Sub
:
```

Παράρτημα Δ

ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ OPENFILEDIALOG1_FILEOK, OPENFILEDIALOG2_FILEOK ΚΑΙ ADDFORM_LOAD

```
Private Sub OpenFileDialog1_FileOk(ByVal sender As System.Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles OpenFileDialog1.FileOk
    PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
End Sub

Private Sub AddForm_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
    Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
    Dim SQLDataAdapter As MySqlDataAdapter
    Dim SQLDataSet As DataSet
    Dim strSQL As String

    strSQL = "SELECT DISTINCT family_name FROM species"

    SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)
    SQLDataSet = New DataSet()
    SQLDataAdapter.Fill(SQLDataSet)

    For Each row As DataRow In SQLDataSet.Tables(0).Rows

        If Not IsDBNull(row.Item("family_name")) Then
```

```
        fn.Add(row.Item("family_name"))
    End If
Next

m_DatabaseConnection.Close()

txtFamily.AutoCompleteMode = AutoCompleteMode.Suggest
txtFamily.AutoCompleteSource = AutoCompleteSource.CustomSource
txtFamily.AutoCompleteCustomSource = fn
txtFamily.DataSource = fn

End Sub

End Class
```

Παράρτημα Ε ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΤΗΣ ΜΕΘΟΔΟΥ BTNSEARCH_CLICK
--

1^ο μέρος

```
Imports MySql.Data.MySqlClient
Imports System.IO
```

```
Public Class EditForm
```

```
    Dim sn As New AutoCompleteStringCollection
    Dim bc As New AutoCompleteStringCollection
```

```
    Dim bcode As String
```

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

```
        If creteria.Text.Length > 0 Then
            Dim qstr As String
```

```
                qstr = "scientific_name"
```

```
                If RadioButton1.Checked = True Then
```

```
                    qstr = "scientific_name"
```

```
                ElseIf RadioButton2.Checked = True Then
```

```
                    qstr = "barcode"
```

```
                End If
```

```
                Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
```

```
                Dim SQLDataAdapter As MySqlDataAdapter
```

```
                Dim SQLDataSet As DataSet
```

```
                Dim strSQL As String
```

```
                strSQL = "SELECT barcode, genus_name, scientific_name, greek_common_name, english_name,
description, fish_photo, fish_diffusion FROM species WHERE " & qstr & " = " & creteria.Text & """
```

```
                SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)
```

```
                SQLDataSet = New DataSet()
```

```
                SQLDataAdapter.Fill(SQLDataSet)
```

```
                .
                .
```

2^ο μέρος

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows bcode = row.Item("barcode")
txtsc.Text = row.Item("scientific_name")
txtgr.Text = row.Item("greek_common_name")
txten.Text = row.Item("english_name")
txtg.Text = row.Item("genus_name")
txtdesc.Text = row.Item("description")

If Not (row.Item("fish_photo").Equals(String.Empty)) Then
    Try
        Dim myData() As Byte = row.Item("fish_photo")
        Dim ms As New MemoryStream(myData)
        Dim im As Image = Image.FromStream(ms)
        PictureBox1.Image = im
    Catch ex As Exception
        PictureBox1.Image = Nothing
    End Try
End If

If Not (row.Item("fish_diffusion").Equals(String.Empty)) Then
    Try
        Dim myData() As Byte = row.Item("fish_diffusion")
        Dim ms As New MemoryStream(myData)
        Dim im As Image = Image.FromStream(ms)
        PictureBox2.Image = im
    Catch ex As Exception
        PictureBox2.Image = Nothing
    End Try
End If
Next
Else
    MessageBox.Show("Θα πρέπει να εισάγετε κάποιο κριτήριο", "Προειδοποίηση!",
    MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If

    btnSaveSpecies.Enabled = True
End Sub
.
```

<p style="text-align: center;">Παράρτημα ΣΤ ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΩΝ RADIOBUTTON_1PAINT ΚΑΙ BTNSAVESPECIES_CLICK</p>
--

```
Private Sub RadioButton1_Paint(sender As System.Object, e As System.Windows.Forms.PaintEventArgs)
Handles RadioButton1.Paint
    If DirectCast(sender, RadioButton).Checked Then
        e.Graphics.FillEllipse(Brushes.Red, New RectangleF(1.5, 5.5, 8.0, 8.0))
    End If
End Sub
Private Sub btnSaveSpecies_Click(sender As System.Object, e As System.EventArgs) Handles
btnSaveSpecies.Click
    Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
    Dim sqlCommand As New MySqlCommand
```

```
Dim strSQL As String
```

```
Dim memImage As New System.IO.MemoryStream  
Dim bytImage() As Byte
```

```
Dim memImage2 As New System.IO.MemoryStream  
Dim bytImage2() As Byte
```

```
If Not IsNothing(PictureBox1.Image) Then  
    PictureBox1.Image.Save(memImage, PictureBox1.Image.RawFormat)  
    bytImage = memImage.GetBuffer()  
End If
```

```
If Not IsNothing(PictureBox2.Image) Then  
    PictureBox2.Image.Save(memImage2, PictureBox2.Image.RawFormat)  
    bytImage2 = memImage2.GetBuffer()  
End If
```

```
strSQL = "UPDATE species SET scientific_name = " & txtsc.Text & ", greek_common_name = " &  
txtgr.Text & ", english_name = " & txten.Text & ", genus_name = " & txtg.Text & ", description = " &  
txtdesc.Text & ", fish_photo = @data, fish_diffusion = @data2 WHERE barcode=" & bcode & ""
```

```
sqlCommand.Connection = m_DatabaseConnection  
sqlCommand.CommandText = strSQL  
sqlCommand.Parameters.AddWithValue("@data", bytImage)  
sqlCommand.Parameters.AddWithValue("@data2", bytImage2)  
m_DatabaseConnection.Open()  
sqlCommand.ExecuteNonQuery()
```

```
m_DatabaseConnection.Close()
```

```
MessageBox.Show("Οι αλλαγές αποθηκεύτηκαν!")
```

```
End Sub
```

```
:  
:
```

Παράρτημα Z ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΟΥ BARCODESEARCH.VB

```
Imports MySql.Data.MySqlClient  
Imports System.IO
```

```
Public Class BarcodeSearch
```

```
    Dim bc As New AutoCompleteStringCollection
```

```
    Dim s As String
```

```
    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles btnSearch.Click
```

```
        If bcodeText.Text.Length = 15 Then
```

```
            bcodeImage.Image = Code128Rendering.MakeBarcodeImage(bcodeText.Text, 2, False)
```

```
            Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user  
id=root; password=123; database=fish_database; pooling=false;")  
            ' database=fish_database; pooling=false;")  
            Dim SQLDataAdapter As MySqlDataAdapter  
            Dim SQLDataSet As DataSet
```

Dim strSQL As String

```
strSQL = "SELECT t1.scientific_name AS lev1sc, t1.greek_common_name AS lev1gr, t1.description  
AS lev1d, " & _  
" t2.scientific_name AS lev2, t2.greek_common_name AS lev2gr, t2.description AS lev2d, " & _  
" t3.scientific_name AS lev3, t3.greek_common_name AS lev3gr, t3.description AS lev3d, " & _  
" t4.scientific_name AS lev4, t4.greek_common_name AS lev4gr, t4.description AS lev4d, " & _  
" t5.scientific_name AS lev5, t5.greek_common_name AS lev5gr, t5.description AS lev5d, " & _  
" t6.scientific_name AS lev6, t6.greek_common_name AS lev6gr, t6.description AS lev6d, " & _  
" t7.scientific_name AS lev7, t7.greek_common_name AS lev7gr, t7.description AS lev7d " & _  
" FROM dbtree1 AS t1 " & _  
" LEFT JOIN dbtree1 AS t2 ON t2.parent_id = t1.id " & _  
" LEFT JOIN dbtree1 AS t3 ON t3.parent_id = t2.id " & _  
" LEFT JOIN dbtree1 AS t4 ON t4.parent_id = t3.id " & _  
" LEFT JOIN dbtree1 AS t5 ON t5.parent_id = t4.id " & _  
" LEFT JOIN dbtree1 AS t6 ON t6.parent_id = t5.id " & _  
" LEFT JOIN dbtree1 AS t7 ON t7.parent_id = t6.id " & _  
" WHERE t7.id IN (Select family_id FROM species WHERE barcode = " & bcodeText.Text & " )"
```

.
. .
.

Try

```
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

For Each row As DataRow In SQLDataSet.Tables(0).Rows

```
lblSuperClass.Text = row.Item("lev1sc")  
lblSuperClassGR.Text = row.Item("lev1gr")  
ToolTip1.SetToolTip(lblSuperClass, row.Item("lev1d"))  
lblClass.Text = row.Item("lev2")  
lblClassGR.Text = row.Item("lev2gr")  
ToolTip1.SetToolTip(lblClass, row.Item("lev2d"))  
lblSubClass.Text = row.Item("lev3")  
lblSubClassGR.Text = row.Item("lev3gr")  
ToolTip1.SetToolTip(lblSubClass, row.Item("lev3d"))  
lblSuperOrder.Text = row.Item("lev4")  
lblSuperOrderGR.Text = row.Item("lev4gr")  
ToolTip1.SetToolTip(lblSuperOrder, row.Item("lev4d"))  
lblOrder.Text = row.Item("lev5")  
lblOrderGR.Text = row.Item("lev5gr")  
ToolTip1.SetToolTip(lblOrder, row.Item("lev5d"))  
lblSubOrder.Text = row.Item("lev6")  
lblSubOrderGR.Text = row.Item("lev6gr")  
ToolTip1.SetToolTip(lblSubOrder, row.Item("lev6d"))  
lblFamily.Text = row.Item("lev7")  
lblFamilyGR.Text = row.Item("lev7gr")  
ToolTip1.SetToolTip(lblFamily, row.Item("lev7d"))
```

Next

```
strSQL = "SELECT barcode, genus_name, scientific_name, greek_common_name, english_name,  
description, fish_photo, fish_diffusion FROM species WHERE barcode = " & bcodeText.Text & ""  
'fish_photo, fish_diffusion FROM species WHERE barcode = " & bcodeText.Text & ""
```

```
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

```
.  
. .  
For Each row As DataRow In SQLDataSet.Tables(0).Rows  
  
    lblGenusName.Text = row.Item("genus_name")  
    lblScientName.Text = row.Item("scientific_name")  
    ToolTip1.SetToolTip(lblScientName, row.Item("description"))  
    lblGrCommon.Text = row.Item("greek_common_name")  
    lblDesc.Text = row.Item("description")  
    lblEngName.Text = row.Item("english_name")  
    s = row.Item("barcode")  
  
    If Not (row.Item("fish_photo").Equals(String.Empty)) Then  
        Try  
            Dim myData() As Byte = row.Item("fish_photo")  
            Dim ms As New MemoryStream(myData)  
            Dim im As Image = Image.FromStream(ms)  
            pbFishPhoto.Image = im  
        Catch ex As Exception  
            pbFishPhoto.Image = Nothing  
        End Try  
    End If  
  
    If Not (row.Item("fish_diffusion").Equals(String.Empty)) Then  
        Try  
            Dim myData2() As Byte = row.Item("fish_diffusion")  
            Dim ms2 As New MemoryStream(myData2)  
            Dim im2 As Image = Image.FromStream(ms2)  
            pbFishDiff.Image = im2  
        Catch ex As Exception  
            pbFishDiff.Image = Nothing  
        End Try  
  
    End If  
Next  
  
. . .  
m_DatabaseConnection.Close()  
  
bcode1.Visible = True  
bcode1.Text = bcodeText.Text.Substring(0, 1)  
  
bcode2.Visible = True  
bcode2.Text = bcodeText.Text.Substring(1, 1)  
  
bcode3.Visible = True  
bcode3.Text = bcodeText.Text.Substring(2, 1)  
  
bcode4.Visible = True  
bcode4.Text = bcodeText.Text.Substring(3, 2)  
  
bcode5.Visible = True  
bcode5.Text = bcodeText.Text.Substring(5, 2)  
  
bcode6.Visible = True  
bcode6.Text = bcodeText.Text.Substring(7, 2)  
  
bcode7.Visible = True
```

```
bcode7.Text = bcodeText.Text.Substring(9, 3)

bcode8.Visible = True
bcode8.Text = bcodeText.Text.Substring(12, 3)

Catch ex As Exception
    MessageBox.Show("Σφάλμα επικοινωνίας με την βάση δεδομένων", "Σφάλμα!",
    MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

Else
    MessageBox.Show("Το barcode δεν είναι έγκυρο", "Προειδοποίηση!", MessageBoxButtons.OK,
    MessageBoxIcon.Warning)

End If

End Sub

.
.
.

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
    PrintForm.Show()
    PrintForm.RichTextBox1.Text = "    Scientific Name: " + Me.IblScientName.Text + " " + vbNewLine +
vbNewLine + _
    "    Greek Common name: " + Me.IblGrCommon.Text + " " + vbNewLine + vbNewLine + _
    "    English Name: " + Me.IblEngName.Text + " " + vbNewLine + vbNewLine + _
    "    Barcode: " + Me.bcodeText.Text + vbNewLine + " " + vbNewLine + _
    "-----" + vbNewLine + _
    bcode1.Text + "    Superclass: " + Me.IblSuperClass.Text + " " + Me.IblSuperClassGR.Text + vbNewLine
+ vbNewLine + _
    bcode2.Text + "    Class: " + Me.IblClass.Text + " " + Me.IblClassGR.Text + vbNewLine + vbNewLine + _
    bcode3.Text + "    Subclass: " + Me.IblSubClass.Text + " " + Me.IblSubClassGR.Text + vbNewLine +
vbNewLine + _
    bcode4.Text + "    Superorder: " + Me.IblSuperOrder.Text + " " + Me.IblSuperOrderGR.Text + vbNewLine
+ vbNewLine + _
    bcode5.Text + "    Order: " + Me.IblOrder.Text + " " + Me.IblOrderGR.Text + vbNewLine + vbNewLine + _
    bcode6.Text + "    Suborder: " + Me.IblSubOrder.Text + " " + Me.IblSuperOrderGR.Text + vbNewLine +
vbNewLine + _
    bcode7.Text + "    Family: " + Me.IblFamily.Text + " " + Me.IblFamilyGR.Text + vbNewLine + vbNewLine +
_
    bcode8.Text + "    Scientific Name: " + Me.IblScientName.Text + " " + vbNewLine + vbNewLine + _
    "-----" + vbNewLine + _
    "    Description: " + Me.IblDesc.Text

    PrintForm.RichTextBox1.SelectionStart = 22
    PrintForm.RichTextBox1.SelectionLength = Me.IblScientName.Text.Length
    PrintForm.RichTextBox1.SelectionFont = New Font(PrintForm.RichTextBox1.SelectionFont,
    FontStyle.Italic)
    PrintForm.RichTextBox1.SelectionLength = 0
End Sub

.
.
.

Private Sub BarcodeSearch_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
    Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
```



```
Dim SqlDataAdapter As MySqlConnection
Dim DataSet As DataSet
Dim strSQL As String
```

```
strSQL = "SELECT barcode FROM species"
```

```
SqlDataAdapter = New MySqlConnection(strSQL, m_DatabaseConnection)
DataSet = New DataSet()
SqlDataAdapter.Fill(DataSet)
```

```
For Each row As DataRow In DataSet.Tables(0).Rows
```

```
    If Not IsDBNull(row.Item("barcode")) Then
        bc.Add(row.Item("barcode"))
    End If
```

```
Next
```

```
m_DatabaseConnection.Close()
```

```
bc.AutoCompleteMode = AutoCompleteMode.Suggest
bc.AutoCompleteSource = AutoCompleteSource.CustomSource
bc.AutoCompleteCustomSource = bc
bc.DataSource = bc
```

```
End Sub
```

```
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
```

```
    PrintForm.Show()
    PrintForm.RichTextBox1.Text = Me.lblScientName.Text + " " + vbNewLine + vbNewLine + _
    Me.lblGrCommon.Text + " " + vbNewLine + vbNewLine + _
    Me.bcodeText.Text + vbNewLine + " " + vbNewLine
```

```
    PrintForm.RichTextBox1.SelectionStart = 0
    PrintForm.RichTextBox1.SelectionLength = Me.lblScientName.Text.Length
    PrintForm.RichTextBox1.SelectionFont = New Font(PrintForm.RichTextBox1.SelectionFont,
```

```
FontStyle.Italic)
```

```
    PrintForm.RichTextBox1.SelectionLength = 0
```

```
End Sub
```

```
End Class
```

Παράρτημα Η

ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΜΕΘΟΔΟΥ BTNSEARCH_CLICK

1^ο μέρος

Φορμα NameSearch

```
Imports MySql.Data.MySqlClient
Imports System.IO
```

```
Public Class NameSearch
```

```
    Dim sn As New AutoCompleteStringCollection
    Dim gcn As New AutoCompleteStringCollection
    Dim en As New AutoCompleteStringCollection
```

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

```
If nameText.Text.Length > 0 Then
```

```
Dim qstr As String  
qstr = "scientific_name"
```

```
If RadioButton1.Checked = True Then  
qstr = "scientific_name"  
ElseIf RadioButton2.Checked = True Then  
qstr = "greek_common_name"  
ElseIf RadioButton3.Checked = True Then  
qstr = "english_name"  
End If
```

```
Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user  
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")  
Dim SQLDataAdapter As MySqlDataAdapter  
Dim SQLDataSet As DataSet  
Dim strSQL As String
```

2^ο μέρος

```
strSQL = "SELECT t1.scientific_name AS lev1sc, t1.greek_common_name AS lev1gr, t1.description  
AS lev1d, " & _  
" t2.scientific_name AS lev2, t2.greek_common_name AS lev2gr, t2.description AS lev2d, " &  
_  
" t3.scientific_name AS lev3, t3.greek_common_name AS lev3gr, t3.description AS lev3d, " &  
_  
" t4.scientific_name AS lev4, t4.greek_common_name AS lev4gr, t4.description AS lev4d, " &  
_  
" t5.scientific_name AS lev5, t5.greek_common_name AS lev5gr, t5.description AS lev5d, " &  
_  
" t6.scientific_name AS lev6, t6.greek_common_name AS lev6gr, t6.description AS lev6d, " &  
_  
" t7.scientific_name AS lev7, t7.greek_common_name AS lev7gr, t7.description AS lev7d " & _  
" FROM dbtree1 AS t1 " & _  
" LEFT JOIN dbtree1 AS t2 ON t2.parent_id = t1.id " & _  
" LEFT JOIN dbtree1 AS t3 ON t3.parent_id = t2.id " & _  
" LEFT JOIN dbtree1 AS t4 ON t4.parent_id = t3.id " & _  
" LEFT JOIN dbtree1 AS t5 ON t5.parent_id = t4.id " & _  
" LEFT JOIN dbtree1 AS t6 ON t6.parent_id = t5.id " & _  
" LEFT JOIN dbtree1 AS t7 ON t7.parent_id = t6.id " & _  
" WHERE t7.id IN (Select family_id FROM species WHERE " & qstr & " = " & nameText.Text & " )"

SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

3^ο μέρος

```
Dim descriptions(7) As String
```

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows
```

```
    lblSuperClass.Text = row.Item("lev1sc")  
    lblSuperClassGR.Text = row.Item("lev1gr")  
    ToolTip1.SetToolTip(lblSuperClass, row.Item("lev1d"))  
    lblClass.Text = row.Item("lev2")  
    lblClassGR.Text = row.Item("lev2gr")  
    ToolTip1.SetToolTip(lblClass, row.Item("lev2d"))  
    lblSubClass.Text = row.Item("lev3")  
    lblSubClassGR.Text = row.Item("lev3gr")  
    ToolTip1.SetToolTip(lblSubClass, row.Item("lev3d"))  
    lblSuperOrder.Text = row.Item("lev4")  
    lblSuperOrderGR.Text = row.Item("lev4gr")  
    ToolTip1.SetToolTip(lblSuperOrder, row.Item("lev4d"))  
    lblOrder.Text = row.Item("lev5")  
    lblOrderGR.Text = row.Item("lev5gr")  
    ToolTip1.SetToolTip(lblOrder, row.Item("lev5d"))  
    lblSubOrder.Text = row.Item("lev6")  
    lblSubOrderGR.Text = row.Item("lev6gr")  
    ToolTip1.SetToolTip(lblSubOrder, row.Item("lev6d"))  
    lblFamily.Text = row.Item("lev7")  
    lblFamilyGR.Text = row.Item("lev7gr")  
    ToolTip1.SetToolTip(lblFamily, row.Item("lev7d"))
```

```
Next
```

```
strSQL = "SELECT barcode, genus_name, scientific_name, greek_common_name, english_name,  
description, fish_photo, fish_diffusion FROM species WHERE " & qstr & " = " & nameText.Text & ""
```

```
SQLDataAdapter = New MySqlDataAdapter(strSQL, m_DatabaseConnection)  
SQLDataSet = New DataSet()  
SQLDataAdapter.Fill(SQLDataSet)
```

4^ο μέρος

```
For Each row As DataRow In SQLDataSet.Tables(0).Rows
```

```
    lblGenusName.Text = row.Item("genus_name")  
    lblScientName.Text = row.Item("scientific_name")  
    ToolTip1.SetToolTip(lblScientName, row.Item("description"))  
    lblGrCommon.Text = row.Item("greek_common_name")  
    lblEngName.Text = row.Item("english_name")  
    lblDesc.Text = row.Item("description")  
    lblBcode.Text = row.Item("barcode")  
    bcodeImage.Image = Code128Rendering.MakeBarcodeImage(row.Item("barcode"), 2, False)
```

```
If Not (row.Item("fish_photo").Equals(String.Empty)) Then
```

```
    Try
```

```
        Dim myData() As Byte = row.Item("fish_photo")  
        Dim ms As New MemoryStream(myData)  
        Dim im As Image = Image.FromStream(ms)  
        pbFishPhoto.Image = im
```

```
    Catch ex As Exception
```

```
        pbFishPhoto.Image = Nothing
    End Try
End If

If Not (row.Item("fish_diffusion").Equals(String.Empty)) Then
    Try
        Dim myData2() As Byte = row.Item("fish_diffusion")
        Dim ms2 As New MemoryStream(myData2)
        Dim im2 As Image = Image.FromStream(ms2)
        pbFishDiff.Image = im2
    Catch ex As Exception
        pbFishDiff.Image = Nothing
    End Try

End If
Next

m_DatabaseConnection.Close()

Else
    MessageBox.Show("Θα πρέπει να εισάγετε κάποιο κριτήριο", "Προειδοποίηση!",
    MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If

bcode1.Visible = True
bcode1.Text = lblBcode.Text.Substring(0, 1)

bcode2.Visible = True
bcode2.Text = lblBcode.Text.Substring(1, 1)

bcode3.Visible = True
bcode3.Text = lblBcode.Text.Substring(2, 1)

bcode4.Visible = True
bcode4.Text = lblBcode.Text.Substring(3, 2)

bcode5.Visible = True
bcode5.Text = lblBcode.Text.Substring(5, 2)

bcode6.Visible = True
bcode6.Text = lblBcode.Text.Substring(7, 2)

bcode7.Visible = True
bcode7.Text = lblBcode.Text.Substring(9, 3)

bcode8.Visible = True
bcode8.Text = lblBcode.Text.Substring(12, 3)

Catch ex As Exception
    MessageBox.Show("Σφάλμα επικοινωνίας με την βάση δεδομένων", "Σφάλμα!",
    MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

End Sub

Private Sub NameSearch_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

    Dim m_DatabaseConnection As MySqlConnection = New MySqlConnection("server=localhost; user
id=root; password=123; database=fish_database; charset=utf8; pooling=false;")
```

```
Dim SqlDataAdapter As MySqlConnection
Dim DataSet As DataSet
Dim strSQL As String
```

```
strSQL = "SELECT scientific_name, greek_common_name, english_name FROM species"
```

```
SqlDataAdapter = New MySqlConnection(strSQL, m_DatabaseConnection)
DataSet = New DataSet()
SqlDataAdapter.Fill(DataSet)
```

```
For Each row As DataRow In DataSet.Tables(0).Rows
```

```
If Not IsDBNull(row.Item("scientific_name")) Then
    sn.Add(row.Item("scientific_name"))
End If
```

```
If Not IsDBNull(row.Item("greek_common_name")) Then
    gcn.Add(row.Item("greek_common_name"))
End If
```

```
If Not IsDBNull(row.Item("english_name")) Then
    en.Add(row.Item("english_name"))
End If
```

```
Next
```

```
m_DatabaseConnection.Close()
```

```
End Sub
End Class
```