

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΗΜ/ΝΙΑ: 13/4/2005

ΤΕΙ: ΜΕΣΟΛΟΓΓΙΟΥ

ΣΧΟΛΗ : ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ: ΕΦΑΡΜΟΓΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΤΗΝ
ΟΙΚΟΝΟΜΙΑ

ΘΕΜΑ: ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΩΤΟΡΙΝΟΛΑΡΥΓΓΟΛΟΓΙΚΟΥ
ΙΑΤΡΕΙΟΥ

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ: ΤΣΙΡΟΓΙΑΝΝΗΣ ΓΙΩΡΓΟΣ

ΦΟΙΤΗΤΕΣ: ΚΟΥΡΤΗΣ ΔΗΜΗΤΡΗΣ (8598)
ΜΑΜΑΣΟΥΛΑΣ ΧΡΗΣΤΟΣ (9683)



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ.....	1
Εισαγωγή	4
Κύκλος ζωής της ανάπτυξης συστημάτων	5
Η σημασία της συστηματικής μεθοδολογίας.....	5
Οι κίνδυνοι της σχεδίασης ενός περιβάλλοντος της Access φιλικού προς τον χρήστη.....	6
1. Φάση ορισμού	6
Ορισμός του προβλήματος που θα λυθεί.....	6
Προσδιορισμός της εμβέλειας ενός έργου.....	7
Προσδιορισμός αν το έργο είναι εφικτό.....	7
Διαίρεση του έργου σε λειτουργικά μέρη.....	8
Επιλογή της ομάδας έργου.....	8
Τεκμηρίωση του ορισμού του προβλήματος, της εμβέλειας, της εφικτότητας, της τιμηματοποίησης και της ομάδας.....	8
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	9
2. Φάση απαιτήσεων.....	9
Συνέντευξη με ένα αντιπροσωπευτικό δείγμα όλων των μετόχων	9
Σχηματισμός των απαιτήσεων του συστήματος με τις οποίες να συμφωνούν όλοι οι μέτοχοι.....	10
Δημιουργία του μοντέλου δεδομένων των χρηστών.....	10
Δημιουργία μιας τυπικής Αναφοράς Απαιτήσεων.....	10
Υπογραφή της αναφοράς απαιτήσεων από τον πελάτη.....	11
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	11
3. Φάση εκτίμησης	12
Επιλογή των κατάλληλων εργαλείων ανάπτυξης.....	12
Δημιουργία λίστας δεξιοτήτων για τα μέλη της ομάδας	12
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	13
4. Φάση Σχεδίασης.....	13
Σχεδίαση της βάσης δεδομένων.....	13
Μεταφράση του Μοντέλου Δεδομένων των Χρηστών σε ένα Τυπικό Μοντέλο Οντότητας-Σχέσης.....	16
Είδη δυαδικών σχέσεων.....	17
Παράδειγμα: το κολέγιο.....	18
Μετασχηματισμός του Μοντέλου E-R σε ένα Σχισιακό Μοντέλο	18
Κανονικοποίηση του Σχισιακού Μοντέλου	20
Τεκμηρίωση των Αποφάσεων Σχεδίασης της Βάσης Δεδομένων	21
Σχεδίαση της εφαρμογής της βάσης δεδομένων.....	22
Σχεδίαση του Περιβάλλοντος Χρήστη μιας Εφαρμογής.....	22
Σχεδίαση της Ροής Ελέγχου της Εφαρμογής.....	23
Παγίδευση Λαθών Σχεδίασης.....	23
Σχεδίαση Λειτουργιών Ασφάλειας.....	23
Τεκμηρίωση της Σχεδίασης.....	24
Υπογραφή του Πελάτη του Εγγράφου Σχεδίασης.....	24
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	24
5. Φάση Υλοποίησης.....	25
Δημιουργία της βάσης δεδομένων	25
Τεκμηρίωση της βάσης δεδομένων.....	26
Σύνταξη της εφαρμογής της βάσης δεδομένων	26
Τεκμηρίωση του κώδικα του προγράμματος.....	27

Συμπλήρωση της βάσης δεδομένων με δεδομένα.....	27
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	28
6. Φάση Δοκιμών και τελικής τεκμηρίωσης.....	29
Χρησιμοποίηση ένα επαγγελματία για τις δοκιμές.....	29
Επαναληπτικές Δοκιμές.....	29
Δοκιμή του συστήματος για λειτουργικότητα, απόδοση και συμβατότητα.....	30
Έλεγχος της απόκρισης του συστήματος σε αναπάντεχη.....	31
συμπεριφορά του χρήστη.....	31
Online βοήθεια.....	32
Τελική έντυπη τεκμηρίωση.....	32
Αποδοχή του πελάτη και υπογραφές.....	32
Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή.....	33
7. Φάση Συντήρησης.....	33
Διόρθωση κρυφών λαθών.....	33
Βελτιώσεις και ενημερώσεις.....	34
Επικοινωνία με Πελάτη.....	34
Έλεγχος των προσδοκιών του πελάτη.....	34
Αντίσταση στην πίεση του χρονοδιαγράμματος.....	35
Όροι βάσεων δεδομένων.....	35
Βάση δεδομένων (database).....	35
Αντικείμενο (object).....	35
Πίνακας (Table).....	36
Φόρμες (Forms).....	36
Ερωτήματα (Queries).....	36
Εκθέσεις (Reports).....	37
Μακροεντολές (Macros).....	37
Λειτουργικές μονάδες (Modules).....	37
Ανάλυση της λειτουργίας του πληροφοριακού συστήματος ΩΡΛ.....	38
Ανάλυση των πινάκων και των συσχετίσεων μεταξύ τους (Μοντέλο οντοτήτων συσχετίσεων).....	38
Ανάλυση των πινάκων της εφαρμογής.....	40
Ανάλυση της λειτουργίας της εφαρμογής.....	43
1. Φόρμα εκκίνησης.....	43
2. Φόρμα επίσκεψης.....	45
3. Φόρμα αντί.....	48
4. Φόρμα φάρυγγας.....	50
5. Φόρμα στόμα.....	51
6. Φόρμα ιστορικό.....	52
7. Φόρμα επεμβάσεις.....	53
8. Φόρμα ραντεβού.....	54
Βιβλιογραφία.....	57

δεδομένων της εξέτασης. Επίσης αποφασίστηκε το χρονικό περιθώριο υλοποίησης ότι θα είναι 6 μήνες.

3. Φάση εκτίμησης

Η συγκεκριμένη φάση είναι ένας χάρτης που μας λέει ακριβώς που θέλουμε να πάμε με το συγκεκριμένο έργο (οπότε τότε θα είμαστε σε θέση να αποφασίσουμε πως θα φτάσουμε εκεί) Επίσης εδώ επιλέγουμε τα κατάλληλα εργαλεία ανάπτυξης.

Επιλογή των κατάλληλων εργαλείων ανάπτυξης

Αφού γνωρίζουμε το μέγεθος και την πολυπλοκότητα της δουλειάς που έχουμε μπορούμε να επιλέξουμε τα εργαλεία ανάπτυξης που θα κάνουν την καλύτερη δουλειά. Για εφαρμογές βάσεων δεδομένων μέχρι 10 χρήστες η Microsoft Access είναι ένα βολικό και φτηνό εργαλείο. Η Access πιθανόν δεν είναι κατάλληλη για μια μεγάλη εφαρμογή που βασίζεται σε μια μεγάλη βάση δεδομένων πελάτη/διακομιστή. Ωστόσο η Access ίσως είναι η καλύτερη επιλογή ανάπτυξης επειδή προσφέρει μια ομαλή διαδρομή μετακίνησης σε μια βάση δεδομένων Microsoft SQL Server. Ο SQL διακομιστής είναι ικανός να χειρίζεται μεγάλες επαγγελματικές βάσεις δεδομένων και εφαρμογές.

Πολλοί προγραμματιστές μεγάλων εφαρμογών, κάνουν πρωτότυπα των βάσεων δεδομένων τους και των εφαρμογών τους στην Access και μετά χρησιμοποιούν τον SQL Server για την τελική έκδοση. Αυτή η μέθοδος εξοικονομεί χρόνο, επειδή είναι ευκολότερο να δουλέψει κάποιος στην Access παρά με τον SQL Server ή με κάποιο άλλο δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων πελάτη/διακομιστή, όπως είναι το DB2 της IBM ή το Oracle της Oracle Corporation ή ακόμα το πολύ ισχυρό Rose της Rational.

Δημιουργία λίστας δεξιοτήτων για τα μέλη της ομάδας

Μια τυπική περιγραφή της εργασίας κάθε μέλους της ομάδας μειώνει την σύγχυση για το ποιος είναι υπεύθυνος για κάθε μέρος της συνολικής εργασίας. Αυτή

η τυπική περιγραφή απαλείφει επίσης τις άχρηστες διπλές προσπάθειες και βοηθάει να διασφαλιστεί ότι δεν έχει παραλειφθεί κάτι σημαντικό.

Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή

Στη συγκεκριμένη εφαρμογή επιλέχτηκε ως εργαλείο υλοποίησης η Access γιατί αποτελεί ολοκληρωμένο περιβάλλον ανάπτυξης με ευκολία στη χρήση και τη διόρθωση. Επιπλέον με την ενσωματωμένη Visual Basic μας δίνει την δυνατότητα να δημιουργήσουμε προχωρημένες δυνατότητες, μερικές από τις οποίες θα αναλυθούν παρακάτω.

4. Φάση Σχεδίασης

Η συγκεκριμένη φάση έχει δύο βασικά χαρακτηριστικά:

- ✓ **Βάση δεδομένων:** είναι το μέρος που αποθηκεύονται όλα τα δεδομένα που μας ενδιαφέρουν. Η σωστή δημιουργία της βάσης είναι το κλειδί για να παραχθεί ένα αποτελεσματικό και αξιόπιστο σύστημα.
- ✓ **Εφαρμογής βάσης δεδομένων:** είναι το πρόγραμμα που αλληλεπιδρά με τη βάση δεδομένων και δίνει τη δυνατότητα στους χρήστες να
 - Προσθέτουν, τροποποιούν και διαγράφουν δεδομένα από τη βάση δεδομένων
 - Να τρέχουν ερωτήματα
 - Να δημοσιεύουν εκθέσεις

Σχεδίαση της βάσης δεδομένων

Τα σημαντικά σημεία μίας σχεσιακής βάσης δεδομένων είναι τα παρακάτω:

- **Ευελξία** Εάν τα δεδομένα αλλάξουν, χρειάζεται να ενημερώνουμε την τιμή μόνο σε ένα σημείο. Όλα τα ερωτήματα, οι φόρμες και οι εκθέσεις κοιτάζουν αυτό το σημείο για τιμές.

- **Απλότητα** Το επίπεδο μοντέλο πινάκων που χρησιμοποιείται ως η βάση για ένα σχεσιακό σύστημα δεδομένων υπαγορεύει μία απλή, χωρίς πλεονασμούς, μέθοδο αποθήκευσης δεδομένων. Κάθε πίνακας στην σχεσιακή σχεδίαση είναι ένα μοναδικό αντικείμενο, που περιέχει δεδομένα σχετικά με τη βάση, όπως για παράδειγμα, έναν εργαζόμενο, ένα προϊόν ή μία παραγγελία.
- **Δύναμη** Η αποθήκευση των δεδομένων σε ξεχωριστούς, αλλά συνδεδεμένους πίνακες επιτρέπει την ομαδοποίηση των στοιχείων, την αναζήτηση, αλλά και την ανάκληση των πληροφοριών με σχεδόν αμέτρητους τρόπους.
- **Ευκολία στην διαχείριση** Με μικρούς, λιγότερο πολύπλοκους πίνακες, η πληροφορία εντοπίζεται και γίνεται ο χειρισμός της ευκολότερα.

Για παράδειγμα, εάν παρακολουθούμε τις αναθέσεις εργασίας των πελατών μας, μπορούμε να βάλουμε όλα τα στοιχεία σε έναν πίνακα, δημιουργώντας μία βάση δεδομένων ενός πίνακα. Θα υπάρχει μία ξεχωριστή καταχώρηση για κάθε νέα παραγγελία με συμβόλαιο. Η αποθήκευση όλων των στοιχείων των πελατών, σε συνδυασμό με τις αναθέσεις εργασίας, θα σήμαινε επανάληψη της ίδιας πληροφορίας. Σε αντίθεση, εάν το τηλέφωνο ενός πελάτη άλλαζε, κάθε καταχώρηση που θα περιείχε τις πληροφορίες του πελάτη, θα έπρεπε να ενημερωθεί με τα πιο πρόσφατα στοιχεία. Οι πληροφορίες για τον εργαζόμενο στον πίνακα των αναθέσεων εργασίας, θα ήταν επίσης επαναλαμβανόμενες.

Πόσο πιο αποτελεσματικό είναι να έχουμε ένα πίνακα για αναθέσεις εργασιών και να χωρίσουμε τις πληροφορίες πελατών και εργαζομένων. Θα μπορούσε να προστεθεί στους πίνακες ένα σύντομο πεδίο που να περιέχει ένα αναγνωριστικό πελάτη, για να δημιουργήσουμε μία σύνδεση μεταξύ των πινάκων, που ονομάζεται **σχέση**. Παρόμοια, ένα αναγνωριστικό εργαζομένου μπορεί να συνδέσει τις αναθέσεις με τον εργαζόμενο που ενεργεί σαν επιθεωρητής.

Κατά την διάρκεια της κατασκευής μία βάσης δεδομένων, ένα από τα βασικά θέματα που θα μας απασχολήσουν είναι πώς να διαχωρίσουμε τα δεδομένα. Στον σχεδιασμό μιας σχεσιακής βάσης δεδομένων, τρία είναι τα βασικά στοιχεία που μας καθοδηγούν.

Αρχικά, τα δεδομένα μπορούν να διαχωριστούν ανά χρήστη, όπως για παράδειγμα, αν ανήκουν οι χρήστες στο γραφείο προσωπικού ή στο τμήμα παραγωγής.

Το δεύτερο στοιχείο που εξετάζεται είναι η εγκυρότητα των δεδομένων. Για παράδειγμα, σε μία εταιρεία παροχής υπηρεσιών, ένας πελάτης μπορεί να υπογράψει ανάθεση εργασίας στην εταιρεία, για πολλές διαφορετικές εργασίες. Η αποθήκευση όλων των δεδομένων μαζί με τα στοιχεία της ανάθεσης εργασίας είναι κάτι που θα σήμαινε την επανάληψη των δεδομένων που αφορούν τον πελάτη, σε κάθε καταχώρηση. Αυτό θα έχει ως αποτέλεσμα να υπάρχουν πολλά αντίγραφα της ίδιας πληροφορίας (εάν η εταιρεία μας είναι αρκετά πετυχημένη για να έχει τους ίδιους πελάτες).

Το τρίτο στοιχείο είναι η διάσταση του χρόνου. Εάν έχουμε πληροφορίες που χρησιμοποιούμε σπάνια, όπως για παράδειγμα, μία λίστα αρχείων από εκπληρωμένες αναθέσεις εργασιών, πρέπει να την αποθηκεύσουμε τις ξεχωριστά και να την ανοίγουμε όποτε χρειάζεται.

Το μοντέλο δεδομένων που κατασκευάζετε για τους χρήστες, που βασίζεται στις συζητήσεις με τον πελάτη, έχει έννοια τόσο για τον προγραμματιστή όσο και για τους χρήστες, αλλά δεν είναι σε μια μορφή που μπορεί να μεταφραστεί κατευθείαν σε μια σχεδίαση βάσης δεδομένων. Πρέπει πρώτα να μετατραπεί σε ένα μοντέλο που να ακολουθεί ένα από τα τυπικά συστήματα μοντελοποίησης τα οποία γεφυρώνουν το κενό μεταξύ των ανθρώπων και των απαιτήσεων ενός ψηφιακού συστήματος διαχείρισης σχεσιακής βάσης δεδομένων. Υπάρχουν διάφορα τέτοια συστήματα μοντελοποίησης, μερικά εκ των οποίων είναι πιο κατάλληλα για κάποιες εφαρμογές από άλλα. Δύο από τα πιο δημοφιλή παραδείγματα είναι το μοντέλο οντότητας-σχέσης (entity-relationship - E-R) και το εννοιολογικό μοντέλο αντικειμένου (semantic object model - SOM). Θα πρέπει ο προγραμματιστής να είναι ειδικός τουλάχιστον σε ένα από τα διαθέσιμα συστήματα μοντελοποίησης και να ξέρει τουλάχιστον τα βασικά πλεονεκτήματα και μειονεκτήματα κάποιων άλλων. Είναι πολύ πιθανό ότι θα μπορεί να χρησιμοποιεί το σύστημα μοντελοποίησης με το οποίο είναι περισσότερο εξοικειωμένος. Διαφορετικά, θα ξέρει τουλάχιστον πού αλλού να ψάξει για ένα κατάλληλο σύστημα..

κολέγιο είναι ένα παράδειγμα μιας σχέσης πολλά-προς-πολλά. Ένας ΣΠΟΥΔΑΣΤΗΣ μπορεί να γραφτεί σε πολλά ΜΑΘΗΜΑΤΑ σε ένα εξάμηνο και κάθε ΜΑΘΗΜΑ μπορεί να περιέχει πολλούς σπουδαστές

Το μοντέλο E-R που κατασκευάζουμε είναι η βάση για την σχεσιακή βάση δεδομένων που θα δημιουργήσουμε χρησιμοποιώντας την Access. Δημιουργούμε το μοντέλο E-R σχεδιάζοντας ένα διάγραμμα E-R που δείχνει όλες τις οντότητες και τις σχέσεις μεταξύ τους. Όταν βλέπουμε το διάγραμμα E-R, μπορεί να θυμηθούμε σημαντικά πράγματα που έχουμε παραλείψει, ή σχέσεις που δεν είναι σωστές. Όταν εξηγούμε το διάγραμμα E-R στον πελάτη σας, μπορούμε να πάρουμε ανεκτίμητες παρατηρήσεις που επιβεβαιώνουν ότι το σύστημά μας συμφωνεί με αυτό που θέλει ο πελάτης.

Παράδειγμα: το κολέγιο

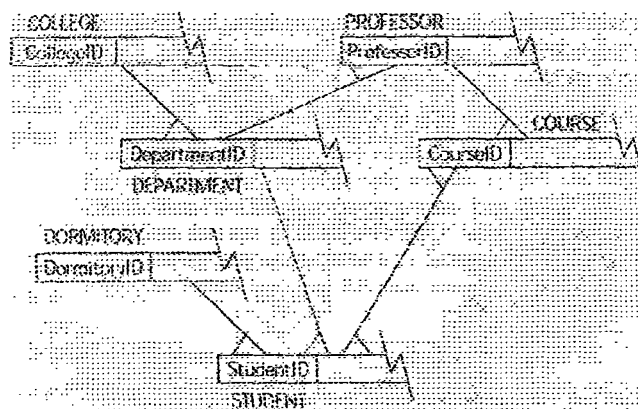
Ένα κολέγιο μπορεί να έχει πολλά τμήματα, αλλά κάθε τμήμα (department) είναι μέλος ενός, και μόνο ενός, κολεγίου. Έτσι η σχέση μεταξύ ΚΟΛΕΓΙΟΥ και ΤΜΗΜΑΤΟΣ είναι ένα-προς-πολλά. Ένα τμήμα μπορεί να έχει πολλούς καθηγητές και πολλούς σπουδαστές, έτσι αυτές οι σχέσεις είναι επίσης ένα-προς-πολλά. Μια εστία μπορεί να παρέχει στέγη σε πολλούς σπουδαστές και ένας καθηγητής μπορεί να διδάσκει σε πολλές τάξεις, οπότε και αυτές οι σχέσεις είναι ένα-προς-πολλά. Ένας σπουδαστής μπορεί να πάρει πολλά μαθήματα και ένα μάθημα μπορεί να έχει πολλούς σπουδαστές, οπότε η σχέση μεταξύ ΣΠΟΥΔΑΣΤΗ και ΜΑΘΗΜΑΤΟΣ είναι πολλά-προς-πολλά.

Μετασχηματισμός του Μοντέλου E-R σε ένα Σχεσιακό Μοντέλο

Οι σχεσιακές βάσεις δεδομένων, που είναι το είδος των βάσεων δεδομένων που σχεδιάζονται στην Access, πρέπει να ακολουθούν το σχεσιακό μοντέλο, που ανακαλύφθηκε από τον E. F. Codd το 1970 (ένα εργαζόμενο της IBM τότε). Ο Codd και άλλοι επιστήμονες της πληροφορικής έχουν βελτιώσει το σχεσιακό μοντέλο με τα

χρόνια προκειμένου να παράγουν ένα σύστημα που να είναι δυνατό και να διατηρεί την ακεραιότητα των δεδομένων.

Το μοντέλο E-R δεν είναι το ίδιο με το σχεσιακό μοντέλο, αλλά ένα συγκεκριμένο μοντέλο E-R μπορεί να μεταφραστεί σε ένα αντίστοιχο σχεσιακό μοντέλο με ένα σχετικά απλό τρόπο. Μέρος της διαδικασίας της μετάφρασης είναι να μετατραπούν οι σχέσεις πολλά-προς-πολλά του μοντέλου E-R, σε ζεύγη από σχέσεις ένα-προς-πολλά. Η απαλοιφή των σχέσεων πολλά-προς-πολλά απλοποιεί κατά πολύ το μοντέλο και περιορίζει την πιθανότητα να χαλάσουν τα δεδομένα όταν γίνουν προσθήκες, διαγραφές και ενημερώσεις. Η παρακάτω εικόνα δείχνει ένα σχεσιακό μοντέλο που αντιστοιχεί στο μοντέλο E-R του κολεγίου.

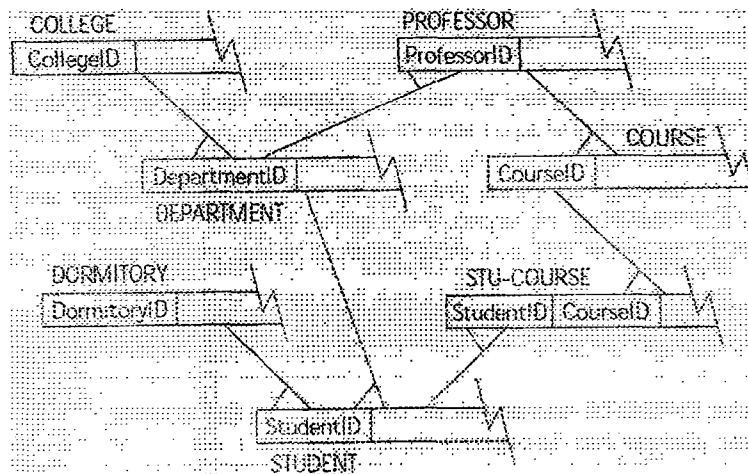


Όταν κάνουμε την μετατροπή από ένα μοντέλο E-R σε ένα σχεσιακό μοντέλο, αλλάζει η ονοματολογία. Μια οντότητα σε ένα μοντέλο E-R αντιστοιχεί σε μια σχέση στο αντίστοιχο σχεσιακό μοντέλο. Ευτυχώς, γιατί μια ιδιότητα σε ένα μοντέλο E-R αντιστοιχεί σε μια ιδιότητα στο σχεσιακό μοντέλο. Ένα στιγμιότυπο οντότητας σε ένα μοντέλο E-R αντιστοιχεί σε μια εγγραφή (tuple) στο σχεσιακό μοντέλο και ένα αναγνωριστικό σε ένα μοντέλο E-R μεταφράζεται σε ένα κλειδί στο σχεσιακό μοντέλο. Αν περισσότερες από μια ιδιότητες μιας οντότητας προσδιορίζουν μοναδικά ένα στιγμιότυπο της οντότητας, επιλέγουμε μια από αυτές για πρωτεύον κλειδί στο σχεσιακό μοντέλο. Μπορούμε να ανακαλέσουμε μία μόνο εγγραφή κάνοντας αναζήτηση με το πρωτεύον κλειδί.

Κάθε ορθογώνιο της παραπάνω εικόνας αντιπροσωπεύει μια σχέση. Βλέπουμε το πρωτεύον κλειδί κάθε σχέσης, αλλά δεν βλέπουμε άλλες ιδιότητες, αφού δεν συμμετέχουν στις σχέσεις. Οι γραμμές μεταξύ των σχέσεων αντιπροσωπεύουν τις σχέσεις. Η πλευρά "πολλά" μιας σχέσης φαίνεται σαν το "πόδι ενός πουλιού". Οι σχέσεις ένα-προς-πολλά έχουν ένα "πόδι πουλιού" στην μια άκρη και οι σχέσεις πολλά-προς-πολλά έχουν "πόδι πουλιού" και στις δυο άκρες.

Περιορίζουμε μια σχέση πολλά-προς-πολλά ενός μοντέλου E-R σε ένα ζεύγος από σχέσεις ένα-προς-πολλά στο αντίστοιχο σχεσιακό μοντέλο, τοποθετώντας μια ενδιάμεση σχέση μεταξύ των δύο σχέσεων που αντιστοιχούν στις δύο αρχικές οντότητες που συνδέονταν από μια σχέση πολλά-προς-πολλά. Κάθε μια από τις δύο αρχικές σχέσεις είναι στην πλευρά "ένα" μιας σχέσης ένα-προς-πολλά με την ενδιάμεση σχέση, αντί να είναι στην πλευρά "πολλά" μιας σχέσης πολλά-προς-πολλά. Η παρακάτω εικόνα δείχνει μια απλοποιημένη έκδοση του σχεσιακού μοντέλου της εικόνας της προηγούμενης σελίδας, έχοντας αφαιρέσει την σχέση πολλά-προς-πολλά. Η ενδιάμεση σχέση STUCOURSE, που δημιουργήσαμε για να αναλύσουμε μια σχέση πολλά-προς-πολλά, έχει ένα πρωτεύον κλειδί που αποτελείται από τα πρωτεύοντα κλειδιά των δύο πινάκων με τους οποίους σχετίζεται, τον πίνακα STUDENT και τον πίνακα COURSE.

Ένα σχεσιακό μοντέλο μεταφράζεται κατευθείαν σε μια σχεσιακή βάση δεδομένων που αποτελείται από πίνακες με στήλες και γραμμές. Οι σχέσεις του μοντέλου γίνονται οι πίνακες της βάσης δεδομένων. Οι ιδιότητες μιας σχέσης γίνονται οι στήλες των αντίστοιχων πινάκων. Οι εγγραφές της σχέσης γίνονται οι γραμμές του πίνακα της βάσης δεδομένων.



Κανονικοποίηση του Σχεσιακού Μοντέλου

Οι σχεσιακές βάσεις δεδομένων μπορεί να έχουν προβλήματα, που ονομάζονται προβλήματα τροποποίησης. Τα προβλήματα τροποποίησης μπορεί να συμβαίνουν όποτε τροποποιείται μια βάση δεδομένων, είτε προσθέτοντας μια νέα γραμμή σε ένα πίνακα, είτε αλλάζοντας τα δεδομένα σε μια υπάρχουσα γραμμή, είτε διαγράφοντας

για γραμμή. Η παρακάτω εικόνα δείχνει ένα τέτοιο παράδειγμα. Δείχνει μερικές γραμμές από τον πίνακα EMPLOYEE. Υποθέτουμε ότι ο εργαζόμενος Kris Howcraft κερδίζει το λόττο και αποφασίζει να παραιτηθεί για να ακολουθήσει μια νέα καριέρα ως τραγουδιστής. Αν διαγράψετε την γραμμή του από τον πίνακα, όχι μόνο τον αφαιρείτε από το πίνακα, αλλά επίσης χάνετε τις πληροφορίες που δείχνουν ότι ο Neil Stryker είναι ο διευθυντής του Τμήματος Logistics.

EMPLOYEE

Employee	Department	Manager
Kris Howcraft	Logistics	Neil Stryker
Luis Mangler	Human Resources	Theodore Ursa
Walter Salvador	Administration	Tim Krieger

Το πρόβλημα συμβαίνει επειδή ο πίνακας EMPLOYEE χρησιμοποιείται για να παρακολουθεί δύο διαφορετικά είδη δεδομένων: το τμήμα στο οποίο ανήκει ένας εργαζόμενος και τα ονόματα των διευθυντών όλων των τμημάτων. Η κανονικοποίηση του πίνακα EMPLOYEE περιλαμβάνει την διαίρεσή του σε δύο πίνακες όπως φαίνεται στην παρακάτω εικόνα: ο ένας πίνακας παρακολουθεί το τμήμα των εργαζομένων και ο άλλος πίνακας καταγράφει τους διευθυντές των τμημάτων.

EMPLOYEE

Employee	Department
Kris Howcraft	Logistics
Luis Mangler	Human Resources
Walter Salvador	Administration

EMPLOYEE

Department	Manager
Logistics	Neil Stryker
Human Resources	Theodore Ursa
Administration	Tim Krieger

Τώρα, αν ο Howcraft φύγει για να κάνει καριέρα τραγουδιστή, δεν χάνονται οι πληροφορίες για το ποιος διευθύνει το τμήμα Logistics.

Τεκμηρίωση των Αποφάσεων Σχεδίασης της Βάσης Δεδομένων

Όταν φτάσουμε στην τελική σχεδίαση της βάσης δεδομένων, τεκμηριώνουμε λεπτομερώς κάθε βήμα της διαδικασίας. Τυπώνουμε και αρχειοθετούμε το διάγραμμα

του μοντέλου E-R, το διάγραμμα του σχεσιακού μοντέλου μας και τις τυπικές περιγραφές κάθε πίνακα - όπως τις στήλες, τις γραμμές, τα κλειδιά, το όνομα των στοιχείων, τον τύπο δεδομένων και την περιγραφή.

Σχεδίαση της εφαρμογής της βάσης δεδομένων

Οι εφαρμογές βάσεων δεδομένων επιτρέπουν στους χρήστες να βάζουν δεδομένα στην βάση δεδομένων, να χειρίζονται τα δεδομένα αφού μπουν στην βάση δεδομένων και να ανακαλούν χρήσιμες πληροφορίες από την βάση δεδομένων. Η Access παρέχει εργαλεία που μας επιτρέπουν να δημιουργούμε μια εφαρμογή χωρίς προγραμματισμό. Αυτά τα προγράμματα έχουν μια γενική εμφάνιση και εκτελούν συνηθισμένες λειτουργίες.

Για να δώσουμε στις εφαρμογές μας μια ξεχωριστή εμφάνιση και για να εκτελούμε μη συνηθισμένες λειτουργίες, μπορούμε να δημιουργούμε εφαρμογές για την βάση δεδομένων χρησιμοποιώντας την γλώσσα προγραμματισμού VBA(Visual Basic for Applications). Είτε χρησιμοποιήσουμε την VBA για να δημιουργήσουμε μια απλή εφαρμογή χωρίς προγραμματισμό είτε την χρησιμοποιήσουμε για μια περίπλοκη εφαρμογή, υπάρχουν διάφορα θέματα που είναι κοινά και στα δύο είδη ανάπτυξης.

Σχεδίαση του Περιβάλλοντος Χρήστη μιας Εφαρμογής

Η διαδικασία της ανάκλησης πληροφοριών πρέπει να είναι πολύ εύκολη και φυσική για τους χρήστες της εφαρμογής . Οι χρήστες μπορεί να έχουν διαφορετικές ανάγκες και η εφαρμογή θα πρέπει να ανταποκρίνεται στις συγκεκριμένες ανάγκες των χρηστών της. Αυτό το θέμα επηρεάζει την εμφάνιση και την λειτουργικότητα των οθονών του περιβάλλοντος χρήστη. Όταν σχεδιάζουμε αυτό το περιβάλλον, πρέπει να σκεφθούμε πώς δουλεύουν οι χρήστες και τι χρειάζονται. Πιθανόν να πρέπει να δημιουργήσουμε πρωτότυπο του προτεινόμενου περιβάλλοντος χρήστη που να δείχνει τις διάφορες κατηγορίες χρηστών. Τα σχόλια του πελάτη σε αυτό το σημείο μπορεί να μας εξοικονομήσουν πολύ χρόνο αργότερα.

Οι εφαρμογές βάσεων δεδομένων θα πρέπει να εμφανίζουν προειδοποιήσεις στον χρήστη πριν εκτελέσουν εντολές που μπορεί να έχουν μεγάλη επίδραση στην βάση δεδομένων, όπως είναι η εντολή DELETE TABLE. Τα λάθη των νόμιμων χρηστών μπορεί να είναι τόσο καταστροφικά, όσο και αυτά των κακόβουλων χρηστών.

Τεκμηρίωση της Σχεδίασης

Η λεπτομερής τεκμηρίωση είναι σημαντική στην Φάση Σχεδίασης. Αυτή περιλαμβάνει την ακριβή δόμηση του συστήματός μας σε χαρτί (ή σε ηλεκτρονικό χαρτί). Πρέπει να γράψουμε γιατί έχουμε επιλέξει να σχεδιάσουμε το σύστημα με κάποιο τρόπο και όχι με κάποιο άλλο τρόπο. Πρέπει να σημειώσουμε τα υπέρ και κατά όλων των δυνατοτήτων σχεδίασης και να δείξουμε ότι, σε σχέση με όλα τα σχετικά στοιχεία, η σχεδίαση που έχουμε επιλέξει είναι η καλύτερη δυνατή.

Υπογραφή του Πελάτη του Εγγράφου Σχεδίασης

Αφού η σχεδίασή μας ολοκληρωθεί και τεκμηριωθεί πλήρως, θα την παρουσιάσουμε στον πελάτη σας. Όταν ο πελάτης διαβάσει το έγγραφο της σχεδίασης και το καταλάβει, πρέπει να τον βάλουμε να υπογράψει και να καταγράψουμε και την ημερομηνία - αυτό δείχνει ότι ο πελάτης συμφωνεί με αυτό που προτείνουμε. Η υπογραφή μας δείχνει ότι έχουμε δεσμευθεί να παράσχουμε αυτό που περιγράφει το έγγραφο σχεδίασης.

Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή

Στη συγκεκριμένη υλοποίηση η Access αποτέλεσε το εργαλείο για την σχεδίαση του μοντέλου οντοτήτων-συσχετίσεων την σχεδίαση των πινάκων, την δημιουργία των σχέσεων μεταξύ των πινάκων και την σχεδίαση της βασικής δομής των φορμών που θα χρησιμοποιηθούν. Εδώ ο πελάτης ζήτησε η κύρια καρτέλα – φόρμα εισόδου να περιέχει τα εξής:

- Λίστα όλων των ασθενών
- Γρήγορη αναζήτηση ασθενούς
- Δυνατότητα προσθήκης και διαγραφής ασθενούς

Η δημιουργία της βάσης δεδομένων θα πρέπει να συμβαίνει μόνο αφού εκτελέσουμε όλες τις εργασίες που καθορίστηκαν στις προηγούμενες φάσεις του κύκλου ανάπτυξης του συστήματος.

Τεκμηρίωση της βάσης δεδομένων

Η Access κάνει πολύ καλή δουλειά στην δημιουργία τεκμηρίωσης της δομής της βάσης δεδομένων που δημιουργούμε, γι' αυτό δεν χρειάζεται να κάνουμε πολλή επιπλέον δουλειά σε αυτό το σημείο:

- Κάνουμε έντυπα αντίγραφα των ορισμών όλων των πινάκων που δημιουργείτε και των σχέσεων μεταξύ των πινάκων.
- Καταγράφουμε γιατί παρεκκλίναμε από την αρχική μας σχεδίαση και τα ακριβή βήματα που ακολουθήσαμε για να φθάσουμε στην τελική μας υλοποίηση.
- Η σχεδίασή μας πιθανόν να αναπτυχθεί καθώς την συνεχίζουμε, και θα επαναληφθεί αρκετές φορές, τουλάχιστον σε μερικούς πίνακες. Αυτές οι πληροφορίες είναι μια ανεκτίμητη αναφορά που θα μας βοηθήσει να κρίνουμε τις αλλαγές που έπρεπε να κάνουμε στην σχεδίαση που καθοριζόταν στο έγγραφο σχεδίασης. Μπορεί επίσης να μας βοηθήσει σε μελλοντικά έργα.

Σύνταξη της εφαρμογής της βάσης δεδομένων

Εδώ είναι που τελικά αρχίζουμε να προγραμματίζουμε. Για ένα προγραμματιστή, είναι μεγάλη η παρόρμηση να αγνοήσει τα προηγούμενα βήματα και να αρχίσει αμέσως την κωδικοποίηση. Αν το κάνει, θα συναντήσει προβλήματα. Αν δεν έχει περάσει από τις Φάσεις Ορισμού, Απαιτήσεων και Εκτίμησης, είναι πιθανόν (επειδή δεν θα υπάρχει επαρκής επικοινωνία με τους χρήστες), να δημιουργήσει ένα σύστημα που δεν θα ανταποκρίνεται πλήρως στις ανάγκες του πελάτη. Αν δεν έχει αφιερώσει αρκετό χρόνο και προσπάθεια στην Φάση Σχεδίασης, ίσως να έχει προβλήματα δομής που θα καταλήξουν σε απώλεια των δεδομένων. Ωστόσο, αν έχει αφιερώσει στις προηγούμενες φάσεις την προσοχή που αξίζουν, μπορεί τώρα να αρχίσει την σύνταξη

του κώδικα που θα μετατρέψει την εργασία σχεδίασης σε μια πραγματικότητα που ανταποκρίνεται στις ανάγκες του πελάτη.

Μπορούμε να δημιουργήσουμε μεγάλο μέρος της εφαρμογής σας χωρίς προγραμματισμό. Η Access έχει ένα γραφικό εργαλείο για δημιουργία των οθονών του περιβάλλοντος χρήστη, των μενού και των εκθέσεων. Δεν χρειάζεται να προγραμματίσουμε για αυτές τις εργασίες. Αυτό εξοικονομεί πολύ χρόνο και επίσης, απαλείφει μια μεγάλη πηγή λαθών. Εκμεταλλευόμαστε την Προβολή σχεδίασης της Access για να δημιουργήσουμε τις φόρμες, τα ερωτήματα και τις εκθέσεις που θα αποτελέσουν το μεγαλύτερο μέρος της αλληλεπίδρασης της εφαρμογής μας με τους χρήστες. Ο προγραμματισμός που θα κάνουμε θα λειτουργήσει στο παρασκήνιο, ελέγχοντας την ροή εκτέλεσης, κάνοντας υπολογισμούς, ή ανακαλώντας δεδομένα.

Τεκμηρίωση του κώδικα του προγράμματος

Ανεξάρτητα από το πόσο καλά γνωρίζουμε την VBA και την SQL, η λογική του προγράμματος μπορεί να είναι περίπλοκη. Έξι μήνες αφού ολοκληρώσουμε το έργο, θα είναι δύσκολο να θυμηθούμε με ποια λογική λειτουργήσαμε την στιγμή που γράφαμε τον κώδικα. Αν μας ζητήσουν να κάνουμε ενημερώσεις ή τροποποιήσεις στο πρόγραμμα, θα βοηθηθούμε πολύ αν έχουμε βάλει πολλά σχόλια στον κώδικα. Πρέπει επίσης να τεκμηριώσουμε τον σκοπό και τον κύριο αλγόριθμο κάθε λειτουργικού τμήματος και να δώσουμε πολλή προσοχή σε ενέργειες που μπορεί να είναι περίπλοκες. Μπορεί να δουλέψουμε σε αυτό τον κώδικα ξανά. Ακόμα πιο πιθανόν είναι ότι ίσως να πρέπει να τροποποιήσει τον κώδικα ένας προγραμματιστής συντήρησης που δεν είναι εξοικειωμένος μαζί του. Ως επαγγελματίες, θα πρέπει να δώσουμε σε αυτό το άτομο όλη την βοήθεια που μπορούμε μέσω μιας καλής τεκμηρίωσης του κώδικα.

Συμπλήρωση της βάσης δεδομένων με δεδομένα

Αφού δημιουργήσουμε την εφαρμογή, τα άτομα που κάνουν εισαγωγή των δεδομένων μπορούν να αρχίσουν να εισάγουν δεδομένα στην βάση δεδομένων χρησιμοποιώντας τις οθόνες που έχουμε δημιουργήσει για αυτό τον σκοπό. Εναλλακτικά, μπορεί να μεταφέρουμε τα δεδομένα κατευθείαν από όπου βρίσκονται

αυτή την στιγμή, αν είναι σε ηλεκτρονική μορφή, χρησιμοποιώντας ένα βοηθητικό πρόγραμμα μεταφοράς των δεδομένων. Οι λεπτομέρειες αυτού του βοηθητικού προγράμματος εξαρτώνται από την μορφή των δεδομένων και το πώς θέλουμε να εμφανίζονται στην βάση δεδομένων σας. Αφού εισαχθεί μια αντιπροσωπευτική ποσότητα δεδομένων στην βάση δεδομένων, μπορούμε να πάμε στην επόμενη φάση, στην Φάση Δοκιμών και Τελικής Τεκμηρίωσης.

Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή

Στη συγκεκριμένη υλοποίηση η Access αποτέλεσε το πολύ – εργαλείο για την σχεδίαση :

- ✓ Την δημιουργία πινάκων
- ✓ Την δημιουργία ερωτημάτων
- ✓ Την δημιουργία φορμών
- ✓ Την δημιουργία εκθέσεων
- ✓ Τον εμπλουτισμό όλων των παραπάνω με τις δυνατότητες της Visual Basic ώστε να είναι ακριβώς προσαρμοσμένες στις ανάγκες του προγραμματιστή.

Για την είσοδο δεδομένων στη συγκεκριμένη εφαρμογή έπρεπε να προσέξουμε τα εξής: έπρεπε οι πελάτες να έχουν διαφορετικό επώνυμο αλλά να υπάρχουν και κάποιοι με ίδιο επώνυμο και διαφορετικό όνομα ώστε να ελέγχουμε πως δουλεύουν τα ερωτήματα. Επίσης να φροντίσουμε στα ραντεβού να υπάρχουν διάφορες ημερομηνίες αλλά και περισσότερα του ενός ραντεβού την ίδια μέρα ώστε να δούμε αν λειτουργεί αποτελεσματικά η αντίστοιχη φόρμα. Επειδή τα δεδομένα που χρησιμοποιούμε είναι αντιπροσωπευτικά, δηλαδή είναι ένα δείγμα από αυτά που θα χρησιμοποιήσει ο πελάτης, φροντίζουμε να έχουμε καταγεγραμμένες τις ιδιαιτερότητές τους για να γνωρίζουμε από πριν τι περιμένουμε να εμφανίσει κάποιο ερώτημα η έκθεση.

6. Φάση Δοκιμών και τελικής τεκμηρίωσης

Αφού δημιουργήσουμε την εφαρμογή μας και την κάνουμε να λειτουργεί με τον τρόπο που θέλουμε, πρέπει να κάνουμε τα εξής: Πρώτον, η εφαρμογή μας πρέπει να ελεγχθεί εκτενώς για να βεβαιωθούμε ότι εκτελεί ότι θα πρέπει να εκτελεί, ότι το εκτελεί όπως θα πρέπει να το εκτελεί και το εκτελεί σε όλα τα σχετικά περιβάλλοντα. Επιπλέον, η εφαρμογή δεν πρέπει να συμπεριφέρεται άσχημα, ανεξάρτητα από το τι κάνουν οι χρήστες μαζί της. Ο μόνος τρόπος να εγγυηθούμε ότι το σύστημα θα ενεργεί όπως επιθυμούμε, είναι να δοκιμάσουμε κάθε λειτουργία σε όλες τις πιθανές λειτουργικές συνθήκες και να χρησιμοποιήσουμε άσχημα το σύστημα με όλους τους τρόπους που μπορεί να το κάνουν οι χρήστες.

Χρησιμοποίηση ένα επαγγελματία για τις δοκιμές

Τα άτομα που δημιουργούν την βάση δεδομένων και την εφαρμογή και αυτοί που γράφουν τον κώδικα που εκτελεί τις λειτουργίες της εφαρμογής, δεν είναι τα καλύτερα άτομα για να εκτελέσουν την Φάση Δοκιμής. Αν έχουν παραβλέψει κάποια πράγματα όταν δημιουργούν την εφαρμογή, είναι πιθανόν να παραβλέψουν τα ίδια σημεία και να μην ανακαλύψουν τα προβλήματα που προκύπτουν. Ιδανικά, αυτοί που κάνουν τις δοκιμές θα πρέπει να είναι άτομα που είναι πολύ καλά εκπαιδευμένοι να ελέγχουν λογισμικό, αλλά που δεν έχουν συμμετάσχει στην ανάπτυξη. Αντίθετα με τους προγραμματιστές, των οποίων ο στόχος είναι να δημιουργήσουν ένα σύστημα χωρίς προβλήματα, ο βασικός σκοπός των ελεγκτών (beta testers) είναι να κάνουν το σύστημα να έχει προβλήματα. Είναι πολύ καλύτερα να βρεθεί ένα πρόβλημα σε αυτή την φάση, παρά να το ανακαλύψει ο πελάτης αργότερα, όταν θα υπάρχουν δεδομένα στην βάση δεδομένων οπότε η διόρθωση θα είναι πιο δύσκολη.

Επαναληπτικές Δοκιμές

Οποιοδήποτε σύστημα που είναι αρκετά μεγάλο και πολύπλοκο, μπορεί να περιέχει προβλήματα όταν τερματίζει η Φάση Υλοποίησης. Αυτά τα προβλήματα μπορεί να είναι καθαρά λάθη ή μπορεί να είναι προβλήματα χρήσης, όπως ένα μπερδεμένο

περιβάλλον χρήστη. Αν, στην διαδικασία εκτέλεσης μιας σειράς από δοκιμές, εντοπισθεί ένα πρόβλημα και μας το δείξουν, η διόρθωσή του είναι δική μας δουλειά.

Ένα πρόβλημα με την ανάπτυξη λογισμικού είναι ότι, όποτε τροποποιείται ένα υπάρχον σύστημα, η τροποποίηση που κάνετε μπορεί να έχει αναπάντεχες συνέπειες. Μπορεί να αλλάξει την συμπεριφορά της εφαρμογής σε μια περιοχή που φαίνεται τελείως άσχετη με την λειτουργία που διορθώσαμε. Εξαιτίας αυτού, όταν επιστρέψουμε την εφαρμογή στα άτομα που κάνουν τις δοκιμές, αυτοί ίσως να μην μπορούν να συνεχίσουν από εκεί που έμειναν όταν ανακάλυψαν το αρχικό πρόβλημα. Πρέπει να εκτελέσουν όλες τις δοκιμές τους ξανά από την αρχή, για να δουν αν τροποποίηση που κάναμε έχει προσθέσει νέα προβλήματα στην λειτουργικότητα που έχουν ήδη ελέγξει. Αυτή η πρακτική της επανεκτέλεσης των δοκιμών από την αρχή μετά την διόρθωση ενός προβλήματος ονομάζεται επαναληπτικές δοκιμές. Οι επαναληπτικές δοκιμές, ακόμα και αν χρειάζονται επιπλέον χρόνο και προσπάθεια, είναι σημαντικό μέρος της ανάπτυξης αν θέλουμε να παραδώσετε ένα αξιόπιστο σύστημα.

Δοκιμή του συστήματος για λειτουργικότητα, απόδοση και συμβατότητα

Στην Φάση Δοκιμής, το πρώτο που πρέπει να προσδιορίσει η επιχείρηση είναι αν το σύστημα κάνει οτιδήποτε θα πρέπει να κάνει. Μπορούμε να το επιτύχουμε αυτό κοιτάζοντας την Αναφορά Απαιτήσεων. Εξετάζουμε το σύστημα με τα δοκιμαστικά δεδομένα που έχουμε εισάγει για να δούμε ότι μπορεί να εκτελέσει όλες τις λειτουργίες που αναφέρονται στην Αναφορά Απαιτήσεων και ότι τα αποτελέσματα όλων των λειτουργιών είναι αυτά που περιμένουμε.

Στην διαδικασία του ελέγχου της λειτουργικότητας, μπορούμε να σημειώσουμε αν η απόδοση είναι ικανοποιητική. Για παράδειγμα, οι ανακλήσεις των δεδομένων από την βάση δεδομένων διαρκούν πολύ; Αυτό είναι κάτι που μπορούμε να ανακαλύψουμε μόνο αν η δοκιμαστική βάση δεδομένων με την οποία δουλεύουμε είναι τόσο μεγάλη όσο η πραγματική βάση δεδομένων. Γενικά, δεν εμφανίζονται προβλήματα απόδοσης όταν ελέγχουμε μια βάση δεδομένων της οποίας οι πίνακες έχουν μόνο ένα μικρό αριθμό από εγγραφές. Πρέπει να βεβαιωθούμε ότι τα δοκιμαστικά δεδομένα μας είναι αντιπροσωπευτικά των πραγματικών δεδομένων.

Η συμβατότητα δεν είναι σίγουρο ότι μπορεί να αποτελέσει πρόβλημα. Αν το πραγματικό σύστημα τρέχει μόνο σε υπολογιστές που έχουν διαμορφωθεί το ίδιο με αυτόν στον οποίο γίνεται η ανάπτυξη, δεν θα πρέπει να συναντήσουμε προβλήματα συμβατότητας. Ωστόσο, θα πρέπει να ελέγξουμε το σύστημα που αναπτύσσουμε σε πολλούς συνδυασμούς λειτουργικών συστημάτων, αν πρόκειται να τρέχει σε υπολογιστές με διαφορετικές διαμορφώσεις μνήμης, δίσκων, εκτυπωτών, επεξεργαστών, ή λειτουργικών συστημάτων. Αυτό γίνεται σχεδόν αδύνατον, εκτός και αν υπάρχουν τυποποιήσεις που περιορίζουν τον αριθμό των πιθανών διαμορφώσεων. Αυτός είναι ένας λόγος που πολλές επιχειρήσεις έχουν την ίδια διαμόρφωση σε όλους τους υπολογιστές της επιχείρησης.

Έλεγχος της απόκρισης του συστήματος σε αναπάντεχη συμπεριφορά του χρήστη

Ποτέ δεν ξέρουμε τι πρόκειται να κάνει ένας χρήστης. Μπορεί να έχουμε πει σαφώς πώς να χρησιμοποιούν το σύστημα, τόσο με έντυπη τεκμηρίωση, όσο και με οδηγίες στην οθόνη. Οι χρήστες βρίσκουν πάντα τρόπους να παρανοούν, ή να ξεχνούν αυτά που πρέπει να κάνουν. Κάνουν ό,τι έχει έννοια για αυτούς εκείνη την στιγμή και όταν το κάνουν, στέλνουν την εφαρμογή μας σε μια κατεύθυνση που δεν προβλέψαμε ποτέ. Όταν συμβεί αυτό, θέλουμε το σύστημά μας να αποκριθεί ομαλά και να κατευθύνει τον χρήστη στον σωστό τρόπο λειτουργίας. Δεν θέλουμε το σύστημα να κλειδώσει ξαφνικά ή, ακόμα χειρότερα, να δώσει στον χρήστη λανθασμένες πληροφορίες.

Η Φάση Δοκιμών θα πρέπει να περιλαμβάνει μια πλήρη γκάμα από ελέγχους, κατά τους οποίους αυτοί που τους κάνουν δεν ακολουθούν τις οδηγίες. Αυτοί οι έλεγχοι θα πρέπει να εξερευνούν οτιδήποτε μπορεί να κάνει ένας χρήστης. Αν υπάρχει μια απόκριση που δεν είναι βοηθητική, θα πρέπει να τροποποιήσουμε το σύστημα ώστε να συμπεριφέρεται με πιο ομαλό τρόπο, ακόμα και όταν χρησιμοποιείται πολύ άσχημα.

Online βοήθεια

Μια από τις πιο χρήσιμες μορφές τεκμηρίωσης που μπορείτε να παρέχουμε με ένα σύστημα είναι ένα πλήρες και καλά οργανωμένο σύστημα online βοήθειας. Θα πρέπει να περιλαμβάνονται επεξηγήσεις σχετικά με το πώς εκτελούνται όλες οι λειτουργίες του συστήματος, με ορισμούς όλων των σχετικών όρων, στα online αρχεία βοήθειας. Συνήθως, η έντυπη τεκμηρίωση και τα φύλλα οδηγιών δεν είναι πάντα κοντά στους χρήστες, αλλά η online βοήθεια είναι πάντα διαθέσιμη και εύκολη στην πρόσβαση. Η online βοήθεια που λέει στους χρήστες αυτό που πιθανόν θέλουν να μάθουν, είναι ένα σημαντικό συστατικό του συνολικού συστήματος.

Τελική έντυπη τεκμηρίωση

Ως προγραμματιστής, θα πρέπει να διατηρήσουμε αυτή την τεκμηρίωση και να δώσουμε ένα αντίγραφο στον πελάτη, σαν ένα πακέτο:

- Η τεκμηρίωση που διατηρείται σε όλες τις φάσεις της ανάπτυξης του συστήματος.
- Πλήρεις καταγραφές των δοκιμών, συμπεριλαμβανομένων των αποτελεσμάτων κάθε ελέγχου, διορθώσεις που έγιναν και αποτελέσματα των επόμενων επαναληπτικών ελέγχων
- ένα έντυπο αντίγραφο της online βοήθειας
- Λειτουργικές οδηγίες για τους χρήστες

Αποδοχή του πελάτη και υπογραφές

Όταν παραδώσουμε το ολοκληρωμένο σύστημα και την τελική τεκμηρίωση στον πελάτη, πρέπει να παρουσιάσουμε τον τρόπο που δουλεύει και να δείξουμε ότι ικανοποιούνται όλα όσα αναφέρονται στην Αναφορά Απαιτήσεων. Αφού ο πελάτης καταλάβει ότι ικανοποιήθηκαν όλα όσα ήθελε, πρέπει να βάλει μια υπογραφή που να υποδηλώνει την αποδοχή του.

Υλοποίηση της συγκεκριμένης φάσης στη συγκεκριμένη εφαρμογή

Στην συγκεκριμένη εφαρμογή τα κείμενα που ακολουθούν (με κατάλληλες αφαιρέσεις και τροποποιήσεις) μπορούν να αποτελέσουν μια πρώτη προσπάθεια έντυπης τεκμηρίωσης της εφαρμογής.

Ο επαγγελματίας που έκανε έλεγχο της εφαρμογής, εντόπισε τα λάθη και βοήθησε να ξεπεραστούν είναι ο υπεύθυνος καθηγητής της συγκεκριμένης διπλωματικής εργασίας.

7. Φάση Συντήρησης

Αυτή η φάση είναι απαραίτητη κυρίως για τους δύο παρακάτω λόγους:

- Διόρθωση κρυφών λαθών
- Βελτιώσεις και ενημερώσεις

Διόρθωση κρυφών λαθών

Ένας λόγος είναι η πιθανότητα να υπάρχουν λάθη στον κώδικα που μπορεί να κάνουν την εφαρμογή να μην λειτουργεί καλά. Πώς μπορεί να υπάρχουν λάθη στον κώδικα; Όλα τα λάθη ανακαλύφθηκαν και διορθώθηκαν στις δοκιμές, έτσι δεν είναι; Η πραγματικότητα είναι, ότι πιθανόν δεν διορθώθηκαν. Ακόμα και οι πολύ εκτενείς έλεγχοι δεν βρίσκουν όλα τα λάθη μιας εφαρμογής. Μερικά λάθη που απαιτούν ένα ασυνήθιστο συνδυασμό συνθηκών, ή υπάρχουν σε ενότητες του κώδικα που χρησιμοποιούνται σπάνια, μπορεί να μην εμφανιστούν παρά εβδομάδες, μήνες, ή ακόμα και χρόνια μετά. Το λάθος Y2K (ιός του έτους 2000), για παράδειγμα, αναγνωρίστηκε σαν πρόβλημα σχεδόν 50 χρόνια μετά. Αφού εντοπίστηκε αυτό το πρόβλημα, έγινε μια έντονη προσπάθεια για να διορθωθεί, που στοίχισε πάρα πολλά χρήματα.

Επειδή κανείς δεν ξέρει καλύτερα την εφαρμογή από την ομάδα που την δημιούργησε, είναι πιθανόν να μας καλέσουν να την διορθώσουμε όταν δημιουργηθούν προβλήματα. Σε αυτό το σημείο, θα είμαστε ευτυχείς που κάναμε εκτενή

τεκμηρίωση. Αν έχουν περάσει δύο εβδομάδες από τότε που τελειώσαμε το έργο, ήδη θα έχει αρχίσει να ξεθωριάζει από την μνήμη μας.

Βελτιώσεις και ενημερώσεις

Επειδή είμαστε αυτοί που γνωρίζουν καλύτερα την εφαρμογή, και έτσι είμαστε ο πιο κατάλληλοι να καταλάβουν πώς να την τροποποιήσουμε, ίσως να πρέπει να κάνουμε βελτιώσεις. Καθώς αλλάζουν οι ανάγκες της επιχείρησης, οι εφαρμογές που υποστηρίζουν την επιχείρηση θα πρέπει επίσης να αλλάξουν. Ίσως να πρέπει να προστεθούν νέες λειτουργίες. Ίσως υπάρχουσες λειτουργίες να είναι άχρηστες πλέον και να πρέπει να διαγραφούν. Μπορεί να έχει αλλάξει το λειτουργικό περιβάλλον κάποιων τμημάτων και έτσι θα πρέπει να αλλάξει και το λογισμικό. Υπάρχουν πολλοί λόγοι που το λογισμικό που έχετε γράψει πρέπει να ενημερώνεται σε συνεχή βάση.

Επικοινωνία με Πελάτη

Ένας σημαντικός παράγοντας επιτυχίας ενός έργου ανάπτυξης λογισμικού είναι να μην σταματήσει η επικοινωνία μεταξύ του προγραμματιστή και του πελάτη. Ο προγραμματιστής πρέπει να έχει μια σαφή ιδέα για αυτό που θέλει ο πελάτης και ο πελάτης πρέπει να έχει μια σαφή ιδέα για το τι να περιμένει από τον προγραμματιστή.

Έλεγχος των προσδοκιών του πελάτη

Πολλά άτομα που δεν ξέρουν καλά τους υπολογιστές, θεωρούν τους υπολογιστές σχεδόν κάτι μαγικό. Πιστεύουν ότι μπορούμε να κάνετε πρακτικά οτιδήποτε με ένα υπολογιστή. Αυτή η λανθασμένη ιδέα συνήθως μεταφράζεται σε μη ρεαλιστικές προσδοκίες για αυτό που μπορεί να κάνει η εφαρμογή μας για αυτούς. Αφού ο πελάτης σας πει τι θέλει, εξαρτάται από εμάς να του πούμε τι μπορεί να γίνει στον διαθέσιμο χρόνο και με τα διαθέσιμα χρήματα. Επειδή σχεδόν πάντα συμβαίνουν απρόβλεπτα προβλήματα, είναι καλό να είμαστε συντηρητικοί στις εκτιμήσεις μας. Είναι πολύ καλύτερα να υποσχεθούμε λιγότερα και να παραδώσουμε περισσότερα παρά το αντίθετο.

Αντίσταση στην πίεση του χρονοδιαγράμματος

Οι πελάτες πάντα θέλουν να παραδίδονται οι νέες εφαρμογές πιο γρήγορα από ό,τι είναι ανθρώπινα δυνατόν. Πιέζουν τον προγραμματιστή να επιταχύνει την δουλειά του και να παραδώσει γρήγορα. Γενικά, ο μόνος τρόπος να παραδώσει γρήγορα ένας προγραμματιστής είναι να κόψει πράγματα και συνήθως αυτό σημαίνει να παραβλεφθούν οι δοκιμές και η τεκμηρίωση. Ο πελάτης πιθανόν να μην το συνειδητοποιεί αυτό, αλλά η πίεση του προγραμματιστή για να επιταχύνει το χρονοδιάγραμμα σε βάρος των δοκιμών ή της τεκμηρίωσης είναι σχεδόν πάντα κακή ιδέα.. Οι μακρόχρονες συνέπειες των ανεπαρκών δοκιμών και της ανεπαρκούς τεκμηρίωσης συνήθως εξισορροπούν τα βραχυχρόνια πλεονεκτήματα του να παραδώσουμε το σύστημα λίγο συντομότερα.

Για να γίνει το καλύτερο για τον πελάτη και για να περιορίσουμε την πίεση που αισθανόμαστε, πρέπει να πείσουμε τον πελάτη ότι είναι καλύτερα να κάνουμε την δουλειά μας σωστά, παρά να την κάνουμε γρήγορα .

Όροι βάσεων δεδομένων

Βάση δεδομένων (database)

Μια βάση δεδομένων είναι ένα σύνολο από αντικείμενα που χρησιμοποιούνται για την αποθήκευση και τη διαχείριση πληροφοριών. Στην access, μια βάση δεδομένων αποθηκεύεται στο δίσκο σ' ένα αρχείο .MDB. Το .MDB αρχείο αποθηκεύει τα αντικείμενα της βάσης δεδομένων και τα δεδομένα που περιέχουν.

Αντικείμενο (object)

Ένα αντικείμενο είναι ένα στοιχείο της βάσης δεδομένων. όπως ένας πίνακας ερώτημα, μία φόρμα ή μία έκθεση.

Πίνακας (Table)

Ένας πίνακας είναι το αντικείμενο της βάσης δεδομένων όπου αποθηκεύονται τα δεδομένα. Ένας πίνακας αποτελείται από πεδία, που ορίζουν ποιες πληροφορίες θα διατηρούνται σε κάθε εγγραφή. Μία εγγραφή αντιπροσωπεύει ένα στοιχείο της βάσης δεδομένων

Για παράδειγμα, ένας πίνακας που περιέχει μία απλή τηλεφωνική λίστα μίας επιχείρησης, μπορεί να έχει τρία πεδία για το Όνομα, το γραφείο και το Εσωτερικό Τηλέφωνο. Κάθε εργαζόμενος της εταιρείας θα πρέπει να έχει μία εγγραφή στον πίνακα.

Στην απλούστερη μορφή τους, τα δεδομένα ενός πίνακα εμφανίζονται σαν φύλλο δεδομένων. Ένα φύλλο δεδομένων είναι σαν ένα λογιστικό φύλλο. Κάθε στήλη στο φύλλο δεδομένων αντιπροσωπεύει ένα πεδίο. Κάθε γραμμή είναι μία εγγραφή.

Φόρμες (Forms)

Τα φύλλα δεδομένων εμφανίζουν όλες τις εγγραφές ενός πίνακα, ταυτόχρονα. Οι φόρμες είναι ένας πιο φιλικός τρόπος να παρουσιάζονται τα δεδομένα. Μία φόρμα εμφανίζει δεδομένα ενός πίνακα, μία εγγραφή κάθε φορά, με τα πεδία διατεταγμένα στην οθόνη.

Είναι επίσης δυνατόν να σχεδιάσουμε διαφορετικές φόρμες, για διαφορετικούς σκοπούς. Για παράδειγμα, μπορεί να έχουμε μία φόρμα για εισαγωγή δεδομένων και μία άλλη για αναζήτηση μέσα στην βάση δεδομένων. Η φόρμα αναζήτησης θα μπορούσε να προστατευτεί ώστε οι χρήστες να μην μπορούν να αλλάξουν τα δεδομένα κατά λάθος.

Ερωτήματα (Queries)

Τα ερωτήματα είναι ένα εργαλείο για την επιλογή και ταξινόμηση δεδομένων, για να δουλεύετε ευκολότερα μαζί τους. Σε πολλές λειτουργίες μίας βάσης δεδομένων μπορεί να μην θέλουμε να δούμε όλες τις εγγραφές ταυτόχρονα ή να αναλύσουμε τα δεδομένα, ή μπορεί να θέλουμε να δούμε ποιες εγγραφές ανταποκρίνονται σε ορισμένα κριτήρια.

Τα ερωτήματα μπορούν να χρησιμοποιηθούν για την ενημέρωση και την τροποποίηση δεδομένων, όπως επίσης και για την εμφάνιση δεδομένων. Μπορούμε επίσης να χρησιμοποιήσουμε εκφράσεις (παραστάσεις) σε ερωτήματα, για να κάνουμε υπολογισμούς με τα δεδομένα μας.

Εκθέσεις (Reports)

Οι πίνακες, τα ερωτήματα και οι φόρμες μπορούν να τυπωθούν όπως τα βλέπουμε στην οθόνη αλλά με τις εκθέσεις, μπορούν να παρουσιαστούν τα δεδομένα σε πιο δομημένη μορφή. Οι εκθέσεις που βασίζονται σε ερωτήματα, τυπώνουν μόνο τα επιλεγμένα πεδία και τις εγγραφές που παράγονται από το ερώτημα. Οι εκθέσεις μπορούν επίσης να περιλαμβάνουν υπολογισμούς και συνοπτικές πληροφορίες των εκτυπωμένων δεδομένων.

Μακροεντολές (Macros)

Οι μακροεντολές είναι λίστες από αποθηκευμένες εντολές, που παρέχουν ένα αυτοματοποιημένο τρόπο εκτέλεσης συχνών ή περίπλοκων δραστηριοτήτων.

Λειτουργικές μονάδες (Modules)

Οι λειτουργικές μονάδες είναι λίστες από εντολές προγραμματισμού που δημιουργούνται στην Visual Basic. Είναι σαν τις μακροεντολές, αλλά επεκτείνουν τη λειτουργία των μακροεντολών και επίσης τρέχουν γρηγορότερα, αλλά απαιτούν γνώσεις προγραμματισμού για να δημιουργηθούν.

Ανάλυση της λειτουργίας του πληροφοριακού συστήματος ΩΡΛ

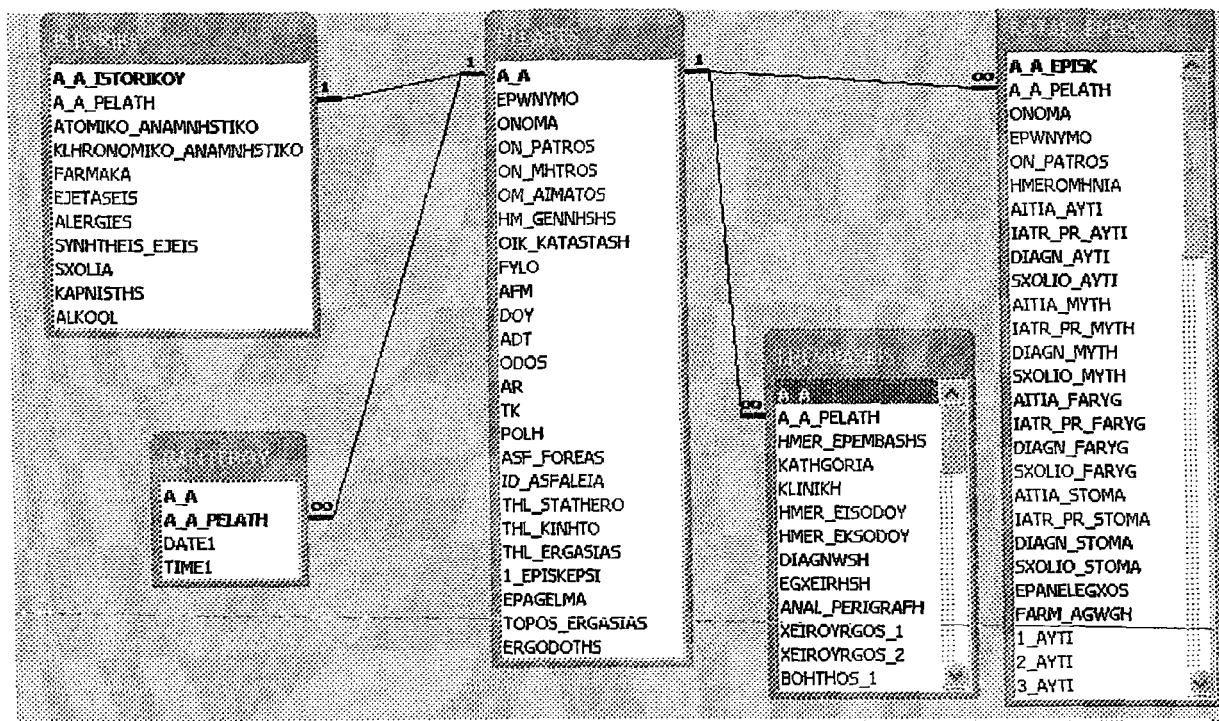
Ανάλυση των πινάκων και των συσχετίσεων μεταξύ τους (Μοντέλο οντοτήτων συσχετίσεων)

Η βάση αποτελείται κυρίως από τρεις πίνακες:

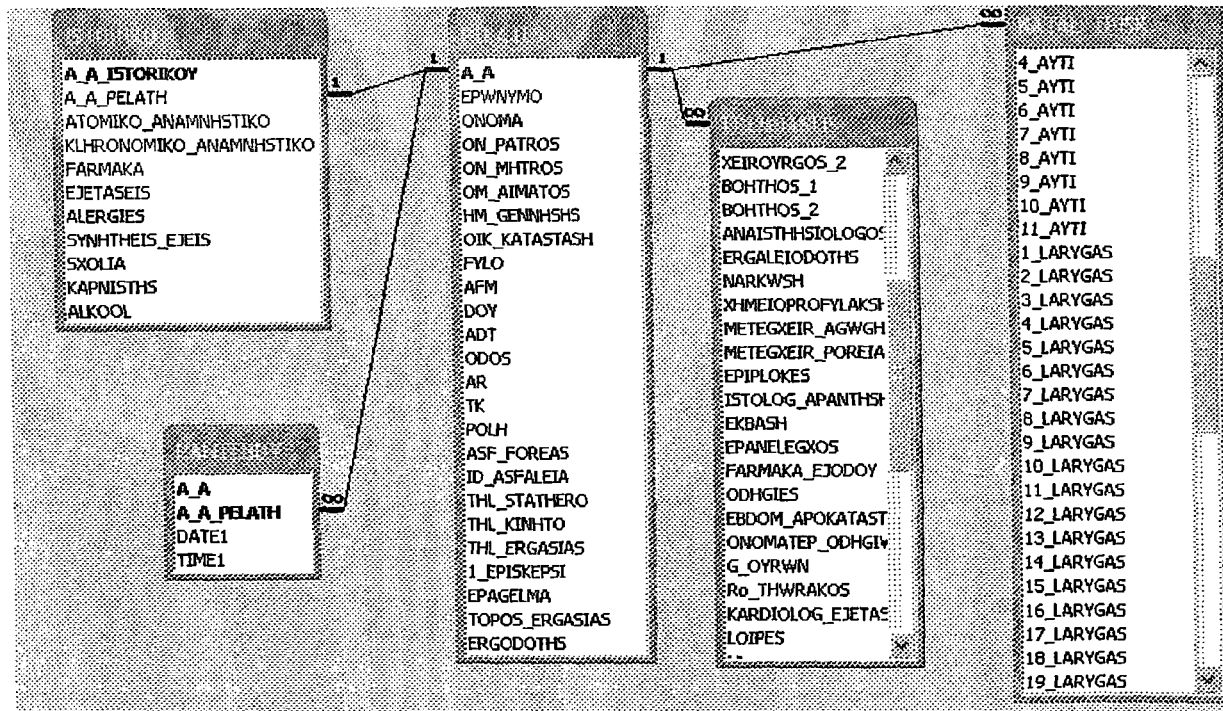
1. Τον πίνακα των πελατών
2. Τον πίνακα με το ιστορικό κάθε πελάτη (ο οποίος έχει σχέση 1 -1 με τον πίνακα των πελατών, απλά έγινε μια διάσπαση για να μην υπάρχουν τόσα πολλά πεδία σε ένα πίνακα)
3. Τον πίνακα με τα δεδομένα της επίσκεψης κάθε πελάτη. Επειδή ο κάθε πελάτης μπορεί να κάνει πολλές επισκέψεις ο πίνακας Πελάτες έχει σχέση 1 προς Πολλά με τον πίνακα των Επισκέψεων.
4. Τον πίνακα με τις επεμβάσεις κάθε ασθενούς – πελάτη, που έχει σχέση 1 προς πολλά με τις εγγραφές του πίνακα Πελάτες.
5. Τον πίνακα στον οποίο κρατούνται τα ραντεβού των πελατών.

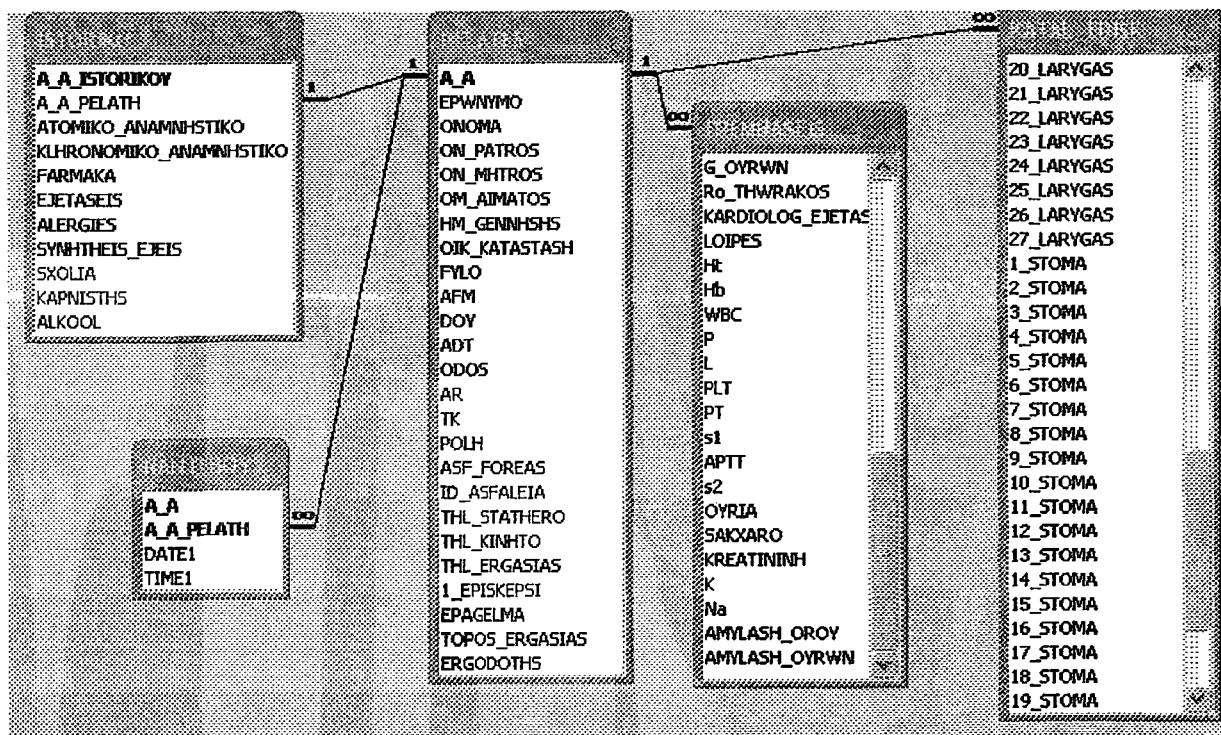
Στον πίνακα των επισκέψεων κρατούνται όλες οι λεπτομέρειες από την εξέταση του πελάτη και είναι αυτός που θα μας βοηθήσει να κάνουμε αναζήτηση για προηγούμενες επισκέψεις και πιθανά ιατρικά προβλήματα του πελάτη.

Ο πίνακας αυτός μπορούσε να διαιρεθεί σε περισσότερους μικρότερους οι οποίοι να συνδεόταν μεταξύ τους αλλά κάτι τέτοιο θα έκανε την διαχείριση με Visual Basic ιδιαίτερα δυσχερή. Αντίθετα η ύπαρξη λίγων πινάκων μας έδωσε την δυνατότητα για απόλυτο έλεγχο των φορμών και των ερωτημάτων ώστε να δημιουργήσουμε μια πιο πλήρη βάση. Άλλωστε τα δεδομένα σε αυτές τις περιπτώσεις δεν είναι ιδιαίτερα πολλά ώστε να προέχουν οι κανόνες κανονικοποίησης των βάσεων δεδομένων, αντίθετα ο έλεγχος των δεδομένων που θα εμφανίζονται και θα αποθηκεύονται παίζει σημαντικό ρόλο .



Επειδή ο πίνακας επισκέψεων δεν εμφανίζεται με όλα τα πεδία του στην παραπάνω εικόνα παραθέτουμε άλλες δύο εικόνες από τις σχέσεις μεταξύ των πινάκων.





Ανάλυση των πινάκων της εφαρμογής

Στον πίνακα πελάτες περιλαμβάνονται τα εξής πληροφορίες:

- A/A Πελάτη (αυτόματη αρίθμηση – πρωτεύον κλειδί του πίνακα)
- Επώνυμο (απαιτείται)
- Όνομα (απαιτείται)
- Όνομα πατρός
- Όνομα μητρός
- Ομάδα αίματος
- Ημερομηνία γέννησης
- ΑΦΜ
- ΑΔΤ (χρησιμοποιήθηκε μάσκα εισαγωγής)
- Οδός
- Ηλικία
- Φύλλο
- Οικογενειακή κατάσταση
- TK (χρησιμοποιήθηκε μάσκα εισαγωγής)

- Ασφαλιστικός Φορέας
- Διάφορα τηλέφωνα
- Εργασία του πελάτη και πληροφορίες γύρω από αυτή
- Ημερομηνία πρώτης επίσκεψης (με προεπιλεγμένη τιμή την τρέχουσα)

Πολύ πιθανό ο πελάτης δεν θα χρειαστεί να γνωρίζει τόσες πληροφορίες για τους ασθενείς του αλλά η συγκεκριμένη εφαρμογή του παρέχει αυτή την δυνατότητα. Επίσης από όλα τα πεδία αυτού του πίνακα απαιτούνται να είναι συμπληρωμένα σε κάποια εγγραφή μόνο τα τρία πρώτα που είναι ο Αύξων Αριθμός (το πρωτεύον κλειδί αυτού του πίνακα που εξασφαλίζει την μοναδικότητα κάθε εγγραφής) το Επώνυμο και το Όνομα. (Είναι πολύ πιθανό κάποιοι πελάτες να μην θελήσουν να μας δώσουν κανένα επιπλέον στοιχείο και να μην είναι σε κάποιον ασφαλιστικό φορέα αλλά να πληρώσουν την εξέταση με μετρητά)

Στον πίνακα **ιστορικό** περιλαμβάνονται τα εξής πληροφορίες:

- A/A Ιστορικού (πρωτεύον κλειδί)
- A/A Πελάτη
- Επώνυμο
- Όνομα
- Ατομικό αναμνηστικό
- Κληρονομικό αναμνηστικό
- Φάρμακα
- Εξετάσεις
- Αλλεργίες
- Σχόλια
- Καπνιστής
- Αλκοόλ

Εδώ το A/A πελάτη είναι το πρωτεύον κλειδί για να συνδεθεί αυτός ο πίνακας με τον πίνακα πελάτες με σχέση ένα προς ένα.

Όλα τα πεδία που αφορούν ιστορικό (Ατομικό αναμνηστικό, Κληρονομικό αναμνηστικό, Φάρμακα, Εξετάσεις, Αλλεργίες, Σχόλια) έχουν οριστεί σαν υπομνήματα ώστε να είναι δυνατόν να καταγραφούν ολόκληρες παράγραφοι από τον θεράποντα ιατρό.

Στον πίνακα επισκέψεις περιλαμβάνονται τα εξής πληροφορίες:

- Α/Α Πελάτη
- Α/Α επίσκεψης
- Επώνυμο
- Όνομα
- Όνομα Πατρός
- Στοιχεία για το όργανο αυτί και πιθανές ασθένειές του
- Στοιχεία για το όργανο μύτη και πιθανές ασθένειές του
- Στοιχεία για το όργανο στόμα και πιθανές ασθένειές του
- Στοιχεία για το όργανο λάρυγγας και πιθανές ασθένειές του
- Στοιχεία για την φαρμακευτική αγωγή
- Στοιχεία για τα ραντεβού του πελάτη μας

Με μια πρόχειρη εξέταση θα μπορούμε να γίνει διαίρεση των στοιχείων που αφορούν το κάθε όργανο που εξετάζει ο ιατρός σε ξεχωριστό πίνακα. Αυτό θα ήταν ιδιαίτερα χρήσιμο στην περίπτωση που η βάση δεδομένων μας είχε πάρα πολλές εγγραφές αλλά κυρίως για την περίπτωση που τα δεδομένα ήταν ανεξάρτητα μεταξύ τους.

Ο συγκεκριμένος πελάτης όμως μας εξήγησε ότι συνήθως υπάρχουν αρκετές αλληλένδετες βλάβες και ασθένειες μεταξύ των παραπάνω οργάνων, ότι συχνά η μία ασθένεια έχει ως συνέπεια την δημιουργία κάποιας άλλης (συχνά σε άλλο όργανο) και ότι κατά την εξέταση κάθε ασθενούς συχνά χρειάζεται να κρατάει στοιχεία και παρατηρήσεις για κάποια όργανα που δεν παρουσιάζουν εκδηλωμένη κάποια ασθένεια αλλά έχουν μια μορφή δυσλειτουργικότητας ως προς το σχήμα, την θέση κ.τ.λ. Σε αυτή την περίπτωση πρέπει να καταγράψει αυτήν την παρατήρηση στο ιστορικό του πελάτη για να του χρησιμεύσει σε πιθανές μελλοντικές διαγνώσεις.

Ανάλυση της λειτουργίας της εφαρμογής

1. Φόρμα εκκίνησης

ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ

ΕΠΩΝΥΜΟ: ΜΑΜΑΣΟΥΛΑΣ
 ΟΝΟΜΑ: ΚΡΗΣΤΟΣ
 ΠΑΤΡΩΝΥΜΟ: ΝΙΚΟΛΑΟΣ ΜΗΤΡΩΝΥΜΟ: ΓΕΩΡΓΙΑ
 ΗΜΕΡ. ΓΕΝΝΗΣΗΣ: 3/2/1974
 ΟΙΚΟΓ. ΚΑΤΑΣΤΑΣΗ: ΕΓΓΑΜΟΣΗ
 ΟΜΑΔΑ ΑΙΜΑΤΟΣ: 0+
 ΦΥΛΟ: ΑΡΡΕΝ
 ΑΦΜ: 065014452
 ΔΟΥ: ΑΓΡΙΝΙΟΥ
 ΑΔΤ: Λ 532456
 ΗΜΕΡΟΜΗΝΙΑ της ΕΠΙΣΚΕΨΗΣ: []

ΔΙΕΥΘΥΝΣΗ: ΤΗΛΕΦΩΝΑ | ΑΛΛΑ |
 ΟΔΟΣ: ΘΥΣΙΑΣ ΑΡ. 1
 ΠΟΛΗ: ΑΓΡΙΝΙΟ
 ΤΚ: 301 00

ΕΥΡΕΣΗ ΑΠΟΘΗΚΕΥΣΗ ΔΙΑΓΡΑΦΗ ΕΚΤΥΠΩΣΗ ΝΕΑ ΕΓΓΡΑΦΗ ΕΞΟΔΟΣ
 ΝΕΑ ΕΠΙΣΚΕΨΗ ΙΣΤΟΡΙΚΟ ΝΕΟ ΧΕΡΟΥΡΓΕΙΟ ΝΕΟ ΡΑΝΤΕΒΟΥ

Εγγραφή: [] από 29
 Παράβολή φόρμας

Στην συγκεκριμένη φόρμα βλέπουμε τα εξής χαρακτηριστικά:

1. τα προσωπικά του στοιχεία του πελάτη
2. το πλήκτρο εύρεσης: που μας δίνει τη δυνατότητα να αναζητήσουμε και να εντοπίσουμε κάποιον ήδη υπάρχοντα πελάτη
3. το πλήκτρο αποθήκευσης: που μας δίνει τη δυνατότητα να αποθηκεύσουμε τα στοιχεία ενός νέου πελάτη
4. το πλήκτρο διαγραφής: που μας δίνει τη δυνατότητα να διαγράψουμε ένα πελάτη
5. το πλήκτρο εκτύπωσης: που μας δίνει τη δυνατότητα να τυπώσουμε τα στοιχεία ενός πελάτη στον εκτυπωτή
6. το πλήκτρο νέας εγγραφής: που μας δίνει τη δυνατότητα να εισάγουμε τα στοιχεία ενός νέου πελάτη
7. το πλήκτρο εξόδου: για να βγούμε από την εφαρμογή

8. το πλήκτρο νέας επίσκεψης: που μας συνδέει με την φόρμα Νέας Επίσκεψης η οποία ανοίγει με τα στοιχεία που βλέπουμε σε αυτή την φόρμα, δηλαδή ανοίγουμε νέα επίσκεψη για τον τρέχοντα πελάτη της αρχικής φόρμας.
9. το πλήκτρο ιστορικό: που μας δίνει το γενικό ιστορικό του ασθενούς και το οποίο συμπληρώνεται, γενικά, στην πρώτη επίσκεψη.
10. το πλήκτρο νέο χειρουργείο: με το οποίο συμπληρώνουμε τα στοιχεία ενός νέου χειρουργείου του ασθενούς.
11. το πλήκτρο νέο ραντεβού: με το οποίο κλείνουμε τα μελλοντικά ραντεβού του πελάτη.

Το τμήμα της Visual Basic που είναι υπεύθυνο για την σύνδεση των δύο φορμών και την σωστή παρουσίαση των χαρακτηριστικών του πελάτη στη φόρμα των επισκέψεων παρατίθεται παρακάτω:

```
Private Sub Εντολή59_Click()
On Error GoTo Err_ Εντολή59_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim strEpwnymo As String
    Dim strOnoma As String
    Dim strA_A As String
    Dim strOn_Patros As String
    strEpwnymo = EPWNYMO.Value
    strOnoma = ONOMA.Value
    strA_A = A_A.Value
    strOn_Patros = ON_PATROS.Value
    stDocName = "KATAL_EPISK_TELIKO"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    Set KATAL_EPISK_TELIKO = Screen.ActiveForm
    DoCmd.GoToRecord , , acNewRec
    EPWNYMO.SetFocus
    KATAL_EPISK_TELIKO.EPWNYMO.Value = strEpwnymo
    ONOMA.SetFocus
    KATAL_EPISK_TELIKO.ONOMA.Value = strOnoma
    KATAL_EPISK_TELIKO.A_A_PELATH.Value = strA_A
```



```

ON_PATROS.SetFocus
KATAL_EPISK_TELIKO.ON_PATROS.Value = strOn_Patros
Exit_ Εντολή59_Click:
Exit Sub
Err_ Εντολή59_Click:
MsgBox Err.Description
Resume Exit_ Εντολή59_Click
End Sub

```

Στα αντίστοιχα string κρατούνται οι τιμές των πεδίων όνομα, επώνυμο, A_A, όνομα πατρός. Η νέα φόρμα ανοίγει και τα αντίστοιχα πεδία συμπληρώνονται αυτόματα με τις τιμές που κρατήσαμε στα strings.

Η εντολή SetFocus εστιάζει στα αντίστοιχα πεδία και κατόπιν αυτά συμπληρώνονται με τις τιμές.

Επιπλέον με εκτεταμένη χρήση του DoCmd μπορούμε να ανοίξουμε την νέα φόρμα και να απαιτήσουμε να αναφέρεται στην εγγραφή της αρχικής φόρμας και να παρουσιάζει δεδομένα που αφορούν στην αρχική φόρμα.

2. Φόρμα επίσκεψης

ΑΡ. ΕΠΙΣΚΕΨΗΣ: 108

ΗΜΕΡΟΜΗΝΙΑ: TODAY

ΑΡ. ΑΣΘΕΝΟΥΣ: 1

ΟΝΟΜΑ: ΧΡΗΣΤΟΣ

ΕΠΩΝΥΜΟ: ΜΑΜΑΣΟΥΛΑΣ

ΟΝ. ΠΑΤΡΟΣ: ΝΙΚΟΛΑΟΣ

ΑΥΤΙ ΜΥΤΗ ΦΑΡΥΓΓΑΣ ΣΤΟΜΑ

ΑΠΙΑ - ΑΥΤΙ

ΙΑΤΡΙΚΕΣ ΠΡΑΞΕΙΣ

ΔΙΑΓΝΩΣΗ

ΣΧΟΛΙΟ

Κλείσιμο Επίσκεψης

Εγγραφή: 1 από 1 (φίλτρο)

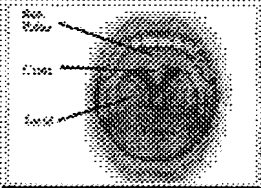
Εδώ εμφανίζονται μόνο τα βασικά στοιχεία των προσωπικών δεδομένων του πελάτη όπως το επώνυμο, το όνομα, ο αύξων αριθμός του ασθενούς και ο αύξων αριθμός της επίσκεψης.

Στη συνέχεια εμφανίζονται τέσσερις καρτέλες, η κάθε μία από τις οποίες περιλαμβάνει τέσσερα πεδία (αιτία, ιατρικές πράξεις, διάγνωση και σχόλιο), δύο απλά κουμπιά (το αριστερά εκ των οποίων χρησιμεύει για αποθήκευση των εγγραφών σε κάποια καρτέλα ενώ το άλλο για ανανέωση των δεδομένων της φόρμας) και τέλος ένα μεγάλο πλήκτρο με το σχήμα του οργάνου που εξετάζουμε και που ο ρόλος του θα αναλυθεί παρακάτω.

Μόνο στην περίπτωση της καρτέλας που αναφέρεται στη μύτη δεν υπάρχει αντίστοιχο πλήκτρο και αυτό γιατί δεν υπάρχει πολυπλοκότητα στις εξετάσεις του συγκεκριμένου οργάνου όπως μας εξήγησε ο γιατρός- πελάτης. Στις τρεις υπόλοιπες καρτέλες εμφανίζονται αντίστοιχα σχήματα που οδηγούν σε αντίστοιχες φόρμες. Στις παρακάτω εικόνες φαίνονται οι αντίστοιχες καρτέλες για το φάρυγγα και το στόμα.

ΑΥΤΗ	ΜΥΤΗ	ΦΑΡΥΓΓΑΣ	ΣΤΟΜΑ
ΑΙΤΙΑ - ΦΑΡΥΓΓΑΣ			
ΙΑΤΡΙΚΕΣ ΠΡΑΞΕΙΣ			
ΔΙΑΓΝΩΣΗ			
ΣΧΟΛΙΟ			

ΑΥΤΙ	ΜΥΤΗ	ΦΑΡΥΓΓΑΣ	ΣΤΟΜΑ
ΑΙΤΙΑ - ΣΤΟΜΑ:			
ΙΑΤΡΙΚΕΣ ΠΡΑΞΕΙΣ			
ΔΙΑΓΝΩΣΗ			
ΣΧΟΛΙΟ:			



Θα αναλύσουμε την λειτουργία του πλήκτρου στην καρτέλα αυτή και τον κώδικα που βρίσκεται πίσω από αυτό. (τα πλήκτρα στις άλλες δύο καρτέλες λειτουργούν με παρόμοιο τρόπο)

```

Private Sub Εντολή66_Click()
On Error GoTo Err_ Εντολή66_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "FORMA_AYTI"
stLinkCriteria = "[A_A_PELATH]=" & Me![A_A_PELATH]
DoCmd.DoMenuItem acFormBar, acRecordsMenu,
acSaveRecord, , acMenuVer70
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_ Εντολή66_Click:
Exit Sub
Err_ Εντολή66_Click:
MsgBox Err.Description
Resume Exit_ Εντολή66_Click
End Sub

```

Το κρίσιμο σημείο σε αυτόν τον κώδικα είναι τα δεύτερο DoCmd: καθορίζει την φόρμα που θα ανοιχτεί (*stDocName = "FORMA_AYTI"*) καθώς και τον τρόπο που θα ανοιχτεί αυτή η φόρμα:

```
(stLinkCriteria = "[A_A_PELATH]=" & Me![A_A_PELATH])
```

Το *A_A_PELATH* της φόρμας που θα ανοιχτεί να είναι ίδιο με το *A_A_PELATH* της φόρμας από την οποία έγινε η κλήση (& Me) .

Με παρόμοιο τρόπο (με την αρχική φόρμα) ανοίγουμε την νέα φόρμα (FORMA_AYTI) με το κριτήριο ότι αναφέρεται στον τρέχοντα πελάτη και θα εισάγουμε δεδομένα για τη συγκεκριμένη επίσκεψη και για το συγκεκριμένο όργανο του πελάτη.

Στο πάνω δεξιά σημείο της φόρμας βλέπουμε το πλήκτρο "TODAY" ο ρόλος του οποίου είναι να τοποθετεί στο text box που βρίσκεται δεξιά του την τρέχουσα ημερομηνία. Αυτό επιτυγχάνεται με την βοήθεια της Visual Basic και συγκεκριμένα με τον παρακάτω κώδικα:

```
Private Sub Εντολή56_Click()  
    HMEROMHΝΙΑ.SetFocus  
    HMEROMHΝΙΑ.Text = Date  
End Sub
```

Το text box που αφορά στην ημερομηνία ονομάζεται HMEROMHΝΙΑ και με την μέθοδο SetFocus γίνεται ενεργό για να δεχτεί την επόμενη διαδικασία. Η επόμενη σειρά στον κώδικα λέει ότι το κείμενο στο text box HMEROMHΝΙΑ θα είναι η τρέχουσα: μέθοδος Date.

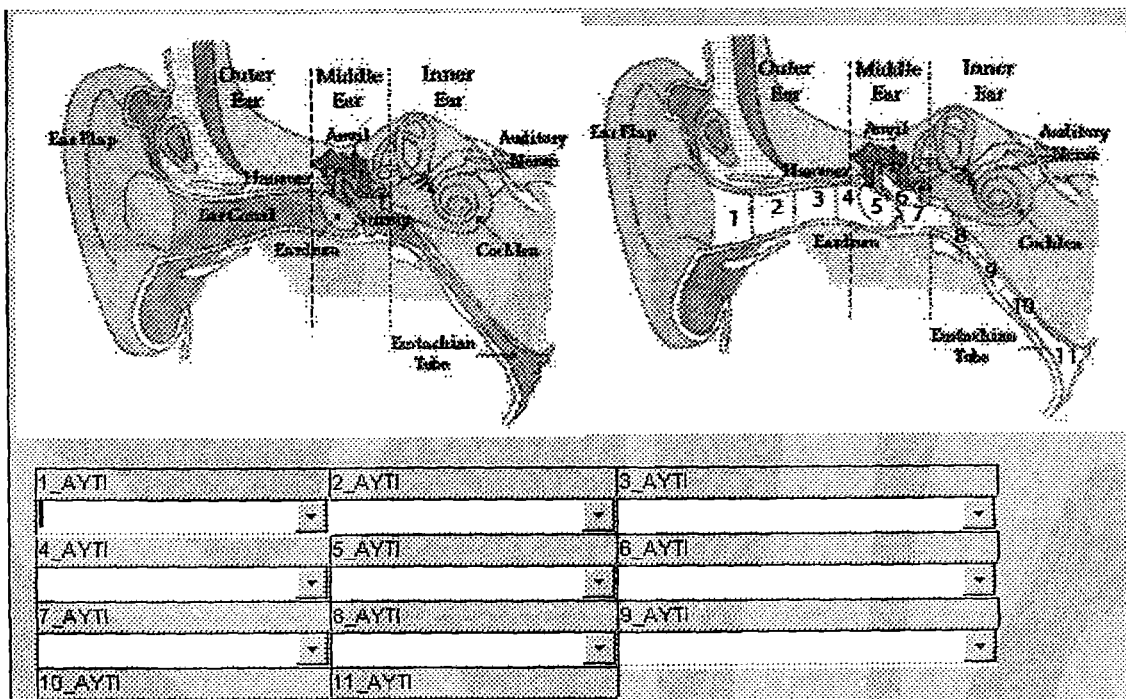
3. Φόρμα αυτή

Στην συγκεκριμένη φόρμα, όπως φαίνεται στην εικόνα που παραθέτουμε παρακάτω, έχουμε την δυνατότητα να δούμε στο σκίτσο το σημείο του αυτιού που

παρουσιάζει κάποιο πρόβλημα (από τα 11 πιθανά σημεία που φαίνονται στο σχήμα) και να επιλέξουμε μία από τις τέσσερις πιθανές εκδοχές που φαίνονται παρακάτω:

1. Εκκριτική ωτίτιδα
2. Δυσλειτουργία Ευσταχιανή
3. Κερί
4. Ωτίτις

Με αυτόν τον τρόπο είναι πολύ εύκολο στον ιατρό να εισάγει στη βάση την βλάβη που παρουσιάζει το συγκεκριμένο όργανο του ασθενούς και το ακριβές σημείο στο οποίο παρουσιάστηκε. Η πληροφορία για το πόσα είναι τα διακριτά σημεία που μπορεί να διαχωριστεί ο έξω ακουστικός πόρος προήλθε από τον ίδιο τον ενδιαφερόμενο και συγκεκριμένα από την βιβλιογραφία του.



Προφανώς με την καταχώρηση των δυσλειτουργιών και το κλείσιμο της συγκεκριμένης φόρμας γίνεται καταχώρηση στους πίνακες που αφορούν τον συγκεκριμένο πελάτη.

4. Φόρμα φάρυγγας

Στην συγκεκριμένη φόρμα, όπως φαίνεται στην εικόνα που υπάρχει παρακάτω, έχουμε την δυνατότητα να δούμε στο σκίτσο το σημείο του λάρυγγα που παρουσιάζει κάποιο πρόβλημα (από τα 27 πιθανά σημεία που φαίνονται στο σχήμα). Εδώ οι προεπιλογές είναι οι ακόλουθες:

1. Λαρυγγίτιδα / Επιγλωτίτιδα
2. Κομβίον / Πολύποδας
3. Καρκίνος.

Προφανώς και εδώ αλλά και στην προηγούμενη καρτέλα, αν ο θεράπων ιατρός το επιθυμεί, μπορεί να επιλέξει την δυσλειτουργία που είναι πιο κοντά σε αυτή που διέγνωσε και κατόπιν στην προηγούμενη φόρμα (από την οποία κλήθηκε η φόρμα αυτή) να αναλύσει εκτενώς τις λεπτομέρειες της διάγνωσης.

The image shows two diagrams of the larynx. The left diagram is a grayscale anatomical illustration with labels: Tongue, Epiglottis, Vocal cord, Pyramiform fossa, Vestibular fold, Trachea, and Esophagus. The right diagram is a similar grayscale illustration but with a 27-point grid overlaid on it, with numbers 1 through 27 indicating specific points of interest. Below the diagrams is a grid of 27 input fields, each labeled 'LARYNGAS' followed by a number from 1 to 27. Each field has a dropdown arrow on the right side.

1. LARYNGAS	2. LARYNGAS	3. LARYNGAS	4. LARYNGAS	5. LARYNGAS
6. LARYNGAS	7. LARYNGAS	8. LARYNGAS	9. LARYNGAS	10. LARYNGAS
11. LARYNGAS	12. LARYNGAS	13. LARYNGAS	14. LARYNGAS	15. LARYNGAS
16. LARYNGAS	17. LARYNGAS	18. LARYNGAS	19. LARYNGAS	20. LARYNGAS
21. LARYNGAS	22. LARYNGAS	23. LARYNGAS	24. LARYNGAS	25. LARYNGAS
26. LARYNGAS	27. LARYNGAS			

Όπως είναι φανερό η επιλογή και αποθήκευση της δυσλειτουργίας με την παραπάνω φόρμα καθιστά ιδιαίτερα εύκολο το έργο του ιατρού ενώ η οπτική επεξεργασία των δεδομένων βοηθάει στον ακριβή εντοπισμό της θέσης της δυσλειτουργίας αλλά είναι και χρήσιμη στην περίπτωση του ιστορικού του ασθενούς.

Τώρα ο θεράπων ιατρός μπορεί να εντοπίσει μεταξύ των άλλων και πιθανή μετατόπιση της ασθένειας κάτι που αν δεν ήταν αποθηκευμένα έτσι τα δεδομένα θα ήταν αδύνατο.

Επίσης είναι ιδιαίτερα εύκολο να βρει σε ποιο σημείο πρέπει, κατά κύριο λόγο, να εξετάσει τον ασθενή του, καθοδηγούμενος από τα σημεία που το ιστορικό δείχνει με ακρίβεια ότι είχε εντοπιστεί δυσλειτουργία.

5. Φόρμα στόμα

Στην συγκεκριμένη φόρμα, όπως φαίνεται στην εικόνα που υπάρχει παρακάτω, έχουμε την δυνατότητα να δούμε σε σκίτσο όλα τα σημεία του στόματος και να εντοπίσουμε με την ίδια ακρίβεια το σημείο ή τα σημεία που παρουσιάζουν κάποια δυσλειτουργία (από τα 27 πιθανά σημεία που έχει διαχωριστεί το στόμα, όπως φαίνονται στο σχήμα). Εδώ οι προεπιλογές είναι οι ακόλουθες:

1. Άφθα
2. Στοματίτιδα
3. Καρκίνος

The diagram shows two views of the mouth. The left view is a grayscale photograph with labels: 'Soft Palate', 'Uvula', and 'Tonsil'. The right view is a schematic diagram of the mouth divided into 27 numbered regions (1-27). Below the diagrams is a form with 27 input fields, each labeled 'X_STOMA' where X is the region number. Each field has a dropdown arrow on the right.

1_STOMA	2_STOMA	3_STOMA	4_STOMA	5_STOMA	6_STOMA
7_STOMA	8_STOMA	9_STOMA	10_STOMA	11_STOMA	12_STOMA
13_STOMA	14_STOMA	15_STOMA	16_STOMA	17_STOMA	18_STOMA
19_STOMA	20_STOMA	21_STOMA	22_STOMA	23_STOMA	24_STOMA
25_STOMA	26_STOMA	27_STOMA			

Ισχύουν και για την συγκεκριμένη φόρμα οι παρατηρήσεις για την αποθήκευση και την ευκολία λειτουργίας που έγιναν στις προηγούμενες φόρμες.

6. Φόρμα ιστορικό

Γενικά η φόρμα αυτή συμπληρώνεται στην πρώτη επίσκεψη ενός νέου ασθενούς, όπως τουλάχιστο συνηθίζεται από το γιατρό. Στη φόρμα αυτή, όπως και τις υπόλοιπες, μεταβαίνουμε από τη φόρμα Πελάτες και συγκεκριμένα από τον συγκεκριμένο πελάτη του οποίου θέλουμε να καταχωρήσουμε το ιστορικό. Τα πεδία Όνομα , Επώνυμο, Όνομα_Πατρός και A_A_Πελάτη συμπληρώνονται αυτόματα με το άνοιγμα της φόρμας, ενώ το πεδίο A_A_Ιστορικού συνεχίζει την αυτόματη αρίθμηση από το τελευταίο νούμερο (Πρωτεύον κλειδί του πίνακα Ιστορικό).

Εδώ κρατούνται τα ατομικά και κληρονομικά αναμνηστικά του κάθε ασθενούς, τα φάρμακα που συνήθως παίρνει για άλλες πιθανές παθήσεις, οι εξετάσεις στις οποίες έχει υποβληθεί, οι αλλεργίες στις διάφορες ουσίες, οι συνήθειες έξεις καθώς και διάφορα σχόλια που δεν περιλαμβάνονται στις παραπάνω περιπτώσεις. Σημαντικό είναι επίσης αν ο ασθενής είναι καπνιστής και η σχέση του με το αλκοόλ, για τα οποία έχει δημιουργηθεί κλιμακωτή λίστα..

ΟΝΟΜΑ:	ΔΗΜΗΤΡΗΣ	ΕΠΩΝΥΜΟ	ΚΟΥΡΤΗΣ
A_A_ISTORIKOU	18	ON PATROS:	ΣΤΕΦΑΝΟΣ
A_A_PELATH	16		
ΑΤΟΜΙΚΟ ΑΝΑΜΗΝΗΣΤΙΚΟ		ΑΛΛΕΡΓΕΣ	
ΚΑΗΡΟΝΟΜΙΚΟ ΑΝΑΜΗΝΗΣΤΙΚΟ		ΣΥΝΗΘΕΙΣ ΕΞΕΙΣ	
ΦΑΡΜΑΚΑ		ΣΧΟΛΙΑ	
ΕΞΕΤΑΣΕΙΣ		ΚΑΠΝΙΣΤΗΣ	<input type="checkbox"/> ΝΑΙ <input checked="" type="checkbox"/> ΟΧΙ <input type="checkbox"/> ΠΟΛΥΚΑΠΝΙΣΤΗΣ
		ΑΛΚΟΟΛ	<input type="checkbox"/> ΣΥΧΝΑ <input type="checkbox"/> ΚΑΙΣΙΜΟ

Εγγραφή: 14 1 από 1 (φίλτρο)
 Προβολή φόρμας ΦΛΤΡ ΑΡ

7. Φόρμα επεμβάσεις

Η φόρμα ανοίγει μέσα από τη φόρμα Πελάτες με το κουμπί Νέο Χειρουργείο. Όπως και στις προηγούμενες καταχωρούνται αυτόματα Όνομα, Επώνυμο, Όνομα Πατρός και A_A_Πελάτη.

A_A	14	ΟΝΟΜΑ	ΔΗΜΗΤΡΗΣ	ΕΠΩΝΥΜΟ	ΚΟΥΡΤΗΣ	ON PATROS	ΣΤΕΦΑΝΟΣ	A_A_PELATH	16
ΗΜΕΡ. ΕΠΕΜΒΑΣΗΣ	3/2/2005	ΗΜΕΡ. ΕΞΟΔΟΥ	3/2/2005	ΗΜΕΡ. ΕΞΟΔΟΥ	6/2/2005	ημέρες νοσηλείας	3		
ΚΑΤΗΓΟΡΙΑ		ΚΛΙΝΙΚΗ	ΟΡΘΟΠΕΔΙΚΗ						
Πάθηση/Εγχειρ. διάγνωση:									
Εγχείρηση:	ΑΡΘΡΟΣΚΟΠΗΣΗ								Κλείσιμο
Αναλυτική περιγραφή:									
ΧΕΙΡΟΥΡΓΟΙ - ΑΝΑΙΣΘΗΣΙΟΛΟΓΟΣ - ΝΑΡΚΩΣΗ : ΠΡΟΕΓΧΕΙΡΗΤΙΚΟΣ ΕΛΕΓΧΟΣ : ΜΕΤΕΓΧΕΙΡΗΤΙΚΗ ΠΟΡΕΙΑ - ΕΠΙΠΛΟΚΕΣ - ΙΣΤΟΛΟΓΙΚΗ : ΟΔΗΓ.									
Χειρουργός 1:	ΔΙΟΝΥΣΟΠΟΥΛΟΣ ΛΕΩΝΙΔΑΣ				Βοηθός 1:	ΠΑΠΠΑ ΓΕΩΡΓΙΑ			
Χειρουργός 2:	ΜΠΟΥΤΑΣ				Βοηθός 2:	ΓΣΟΥΚΑΛΑ ΑΓΓΕΛΙΚΗ			
Ανασθησιολόγος:	ΒΟΥΤΕΝΙΩΤΟΥ				Εργαλειοθήκη:				
Νάρκωση:	ΕΠΙΣΚΛΗΡΙΔΙΟΣ				Χημειοπροφύλαξη:	ΑΜΟΧΙΛ 500			

Εγγραφή: 14 1 από 3 (φίλτρο)
 Προβολή φόρμας ΦΛΤΡ ΚΕΦ ΑΡ

Να σημειωθεί ότι το πεδίο Ημέρες Νοσηλείας συμπληρώνεται αυτόματα από το πρόγραμμα όταν συμπληρωθούν τα πεδία Ημερομηνία Εισόδου και Ημερομηνία Εξόδου. Η φόρμα περιέχει πλήρη στοιχεία για την επέμβαση με τη χρήση της καρτέλας που φαίνεται καλύτερα στις παρακάτω εικόνες:

ΧΗΡΟΥΡΓΟΙ - ΑΝΑΙΣΘΗΣΙΟΛΟΓΟΣ - ΝΑΡΚΩΣΗ - ΠΡΟΕΓΧΕΙΡΗΤΙΚΟΣ ΕΛΕΓΧΟΣ - ΜΕΤΕΓΧΕΙΡΗΤΙΚΗ ΠΟΡΕΙΑ - ΕΠΙΠΛΟΚΕΣ - ΙΣΤΟΛΟΓΙΚΗ - ΟΔΗΓ

Ηt: 0 % Hb: 0 g/dl WBC: 0 (H: 0 % L: 0 %) Γ. Ούρων

PLT: 0 %ML PT: 0 / 0 s APTT: 0 / 0 s Ρα

ΟΥΡΙΑ: 0 mg/dl Σάκχαρο: 0 mg/dl Κρεατινίνη: 0 mg/dl Καρδιολογική Εξέταση

K: 0 Na: 0 Αιμιάση Ορού: 0 Αιμιάση Ούρων: 0 Λοιπές

Εγγραφή: 1 από 3 (φίλτρο)

Προβολή φόρμας ΦΛΤΡ ΑΡ

ΧΗΡΟΥΡΓΟΙ - ΑΝΑΙΣΘΗΣΙΟΛΟΓΟΣ - ΝΑΡΚΩΣΗ - ΠΡΟΕΓΧΕΙΡΗΤΙΚΟΣ ΕΛΕΓΧΟΣ - ΜΕΤΕΓΧΕΙΡΗΤΙΚΗ ΠΟΡΕΙΑ - ΕΠΙΠΛΟΚΕΣ - ΙΣΤΟΛΟΓΙΚΗ - ΟΔΗΓ

ΜΕΤΕΓΧΕΙΡ_ΑΔΑΜΩΧ: ΜΕΤΕΓΧΕΙΡ_ΠΟΡΕΙΑ:

ΕΠΙΠΛΟΚΕΣ ΙΣΤΟΛΟΓ_ΑΠΑΝΤΗΣΗ:

ΕΚΒΑΣΗ ΕΡΑΝΕΛΕΓΧΟΣ:

Εγγραφή: 1 από 3 (φίλτρο)

ΠΡΟΕΓΧΕΙΡΗΤΙΚΟΣ ΕΛΕΓΧΟΣ - ΜΕΤΕΓΧΕΙΡΗΤΙΚΗ ΠΟΡΕΙΑ - ΕΠΙΠΛΟΚΕΣ - ΙΣΤΟΛΟΓΙΚΗ - ΟΔΗΓΕΣ - ΦΑΡΜΑΚΑ ΕΞΟΔΟΥ - ΑΠΟΚΑΤΑΣΤΑΣΗ

Φάρμακα κατά την έξοδο:

Πιθανός χρόνος αποκατάστασης: 0 (ΕΡ6)

Οδηγίες για δραστηριότητες που πρέπει να περιοριστούν. Διαια. Καθαριότητα. Συμπτώματα που πρέπει να αναφερθούν άμεσα στο γιατρό.

Οι οδηγίες δόθηκαν στον/την:

8. Φόρμα ραντεβού

Για να ορίσουμε ένα νέο ραντεβού κάποιου πελάτη εργαζόμαστε ως εξής: στη βασική φόρμα Πελάτες βρίσκουμε τον εν λόγω πελάτη και πατώντας το κουμπί Νέο Ραντεβού ανοίγει η φόρμα Ραντεβού με τα στοιχεία του πελάτη αυτού. Στη συνέχεια

συμπληρώνουμε στο πεδίο DATE1 την ημερομηνία στην οποία θέλουμε το ραντεβού και πατάμε το κουμπί εμφάνιση.

	ημερομηνία	ώρα
Λίστα18	12/3/2005	8:55:00 πμ
	12/3/2005	10:20:00 πμ
	12/3/2005	12:00:00 μμ
	12/3/2005	1:50:00 μμ
	12/3/2005	3:10:00 μμ
	12/3/2005	4:50:00 μμ
	12/3/2005	5:45:00 μμ
	12/3/2005	6:50:00 μμ
	12/3/2005	7:05:00 μμ
	12/3/2005	8:00:00 μμ

Η λίστα 18 θα συμπληρωθεί με τις ώρες των καταχωρημένων ραντεβού της ημερομηνίας αυτής. Έτσι, ο γιατρός έχει τη δυνατότητα να δει ποια ώρα ακριβώς μπορεί να δεχτεί το νέο ασθενή και να συμπληρώσει το πεδίο TIME1.

Πιο κάτω παρατίθεται ο κώδικας που υπάρχει στο κουμπί Εμφάνισε:

```
Private Sub Εντολή20_Click()
```

```
Dim db2 As Database
```

```
Dim ssq1 As String
```

```
Dim rstA As DAO.Recordset
```

```
Dim strA As String
```

```
If Λίστα18.ListCount <> 0 Then
```

```
For i = Λίστα18.ListCount - 1 To 0 Step -1
```

```

        Λίστα18.RemoveItem i
    Next
End If

ssql = "select * from RANTEBOY where RANTEBOY.DATE1=DATE1
order by RANTEBOY.TIME1"

Set db2 = CurrentDb
Set rstA = db2.OpenRecordset(ssql, dbOpenSnapshot)

While rstA.EOF = False
    Λίστα18.AddItem Mid(rstA!DATE1, 1, 10) & "          " &
    Mid(rstA!TIME1, 1, 15)
    rstA.MoveNext
Wend

End Sub

```

Η δομή if – then – endif περιέχει εμφωλευμένη μια επαναληπτική δομή for, η οποία διαγράφει τα τυχόν στοιχεία της λίστας από προηγούμενη εκτέλεση. Στη συνέχεια στο string ssql γράφουμε το SQL ερώτημα με βάση το οποίο θα ανοίξει το recordset rstA, σαν εικόνα (snapshot) της τρέχουσας βάσης που ονομάστηκε db2.

Η δομή While – Wend χρησιμοποιείται για να γεμίσει τη λίστα 18 με τα στοιχεία που θέλουμε. Όσο υπάρχουν εγγραφές στο recordset, η εντολή αναλαμβάνει να τις εμφανίσει μια προς μια στο πλαίσιο λίστας που αδειάσαμε προηγουμένως.

Η εντολή “mid” αναλαμβάνει να εμφανίσει τα πεδία rstA!DATE1, rstA!TIME1 με ένα συγκεκριμένο μέγεθος (προσθέτοντας κενά). Τέλος το recordset είναι ταξινομημένο με βάση την ώρα του κάθε ραντεβού.

Βιβλιογραφία:

1. Εισαγωγή στα συστήματα βάσεων δεδομένων τόμος 1(C.J Date)
2. Εισαγωγή στα συστήματα βάσεων δεδομένων τόμος 2(C.J Date)
3. Εγχειρίδιο προγραμματισμού της Access 2003 με VBA (Allen Taylor & Virginia Andersen)
4. Σημειώσεις του μαθήματος βάσεις δεδομένων II (κ. Σταυρακούδης)
5. Σημειώσεις του μαθήματος ανάλυση πληροφοριακών συστημάτων I (κ. Βώρος)
6. Σημειώσεις του μαθήματος ανάλυση πληροφοριακών συστημάτων II (κ. Μπελληγιάννης)
7. Πλήρης οδηγός της Access 2002 (Andersen)
8. ECDL (εκδόσεις Γκιούρδας)
9. Ανάλυση και σχεδιασμός συστημάτων (Βασίλη Λαοπόδη)