

ΕΦΑΡΜΟΓΗ

ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΧΡΗΣΗ

ΕΡΓΑΛΕΙΩΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Περιεχόμενα

1	Εισαγωγή.....	1
2	Εργαλεία λογισμικού.....	3
2.1	Γιατί Delphi;.....	4
2.2	Γιατί SQL Server;.....	5
2.3	Τι είναι το interface;.....	6
2.4	Τι είναι Σύστημα Βάσης Δεδομένων.....	7
3	Ανάλυση Εφαρμογής.....	9
3.1	Δημιουργία Βάσης Δεδομένων.....	9
3.2	Το μοντέλο E. R.	13
3.3	Σύνδεση Βάσης με Interface.....	14
3.3.1	Δημιουργία ODBC Ψευδωνύμου.....	15
3.3.2	Δημιουργία BDE Ψευδωνύμου.....	16
3.4	Οι φόρμες της Εφαρμογής.....	17
3.5	Τα Στοιχεία Της Εφαρμογής.....	18
3.6	Το Αντικείμενο DataModule.....	23
3.7	Συναρτήσεις.....	24
3.8	Σημάδια Θέσεων (Bookmarks).....	25
4	Περαιτέρω Ανάλυση της Κάθε Φόρμας.....	26
4.1	Φόρμα «Ιατρείο».....	26
4.2	Φόρμα «Ασθενείς».....	28
4.2.1	Φόρμα «Νέα Επίσκεψη».....	34
4.2.2	Φόρμα «Επισκέψεις».....	36
4.2.3	Φόρμα «Εύρεση Ασθενών».....	38
4.2.3.1	Φόρμα «Προσωπικά Στοιχεία Ασθενούς».....	40
4.3	Φόρμα «Ραντεβού».....	45
4.4	Φόρμα «Στατιστικά Στοιχεία Ασθενών».....	48
4.5	Φόρμα «Συνεργάτες».....	54
4.5.1	Φόρμα «Εύρεση Συνεργατών».....	58
	Παράρτημα.....	61

1 Εισαγωγή

Η συγκεκριμένη εφαρμογή έχει αναπτυχθεί με σκοπό να χρησιμοποιηθεί από ιατρούς γενικής ιατρικής καθώς και συγκεκριμένων ειδικοτήτων αφού όμως προηγηθούν προσαρμογές ανάλογες με τις ιδιαίτερες ανάγκες της κάθε ειδικότητας.

Κατά τη διάρκεια μιας μικρής έρευνάς μας και μετά από συνάντησή μας με τρεις ιατρούς και συζήτηση μαζί τους για την ανάπτυξη της εν λόγω εφαρμογής, παρατηρήσαμε και καταγράψαμε τις απαιτήσεις των ιατρών από μια ιατρική εφαρμογή. Μια ιατρική εφαρμογή πρέπει:

- ☛ Να παρέχει στον ιατρό την δυνατότητα καταγραφής και αποθήκευσης των προσωπικών στοιχείων κάθε ασθενούς που έρχεται προς εξέταση, τα οποία είναι: Όνομα ασθενούς, Επώνυμο, Πατρώνυμο, Όνομα συζύγου, Ημερομηνία γέννησης, Επάγγελμα, Διεύθυνση, Τηλέφωνο, Ασφάλιση, Αριθμός ασφάλισης, Αριθμός Μητρώου, Ιστορικό ασθενούς.
- ☛ Να παρέχει επίσης την δυνατότητα καταγραφής και αποθήκευσης των χαρακτηριστικών τις αιτίες της επίσκεψης του ασθενούς στο ιατρείο, δηλαδή τα συμπτώματά του, τις εξετάσεις που πρέπει να κάνει, την διάγνωση του ιατρού και την φαρμακευτική αγωγή που πρέπει να ακολουθήσει για την θεραπεία του.
- ☛ Να παρέχει στον ιατρό, οποιαδήποτε στιγμή πληροφορίες σχετικά με ένα ή περισσότερους

ασθενείς, όσον αφορά τα προσωπικά του στοιχεία αλλά και όλες τις προηγούμενες επισκέψεις του στο ιατρείο.

- ☛ Να παρέχει στον ιατρό την δυνατότητα καταγραφής και αποθήκευσης των προσωπικών στοιχείων των συνεργατών του ιατρού (ακτινολογικών και μικροβιολογικών εργαστηρίων, άλλων ιατρών, προμηθευτών αναλώσιμων υλικών και πάγιου εξοπλισμού και άλλων) τα οποία είναι: Α. Φ. Μ., Ονομα, Επώνυμο, Ειδικότητα, Διεύθυνση, Τηλέφωνο, E-Mail, Web Site, ιδιαιτερότητα.
- ☛ Να παρέχει στον ιατρό, οποιαδήποτε στιγμή, πληροφορίες σχετικά με ένα ή περισσότερους συνεργάτες του, όσον αφορά τα προσωπικά του στοιχεία.
- ☛ Να μην επιτρέπει την καταγραφή και αποθήκευση ενός ασθενούς ή ενός συνεργάτη πάνω από μια φορά.
- ☛ Να παρέχει στον ιατρό την δυνατότητα να κλείνει και να ακυρώνει εύκολα και γρήγορα τηλεφωνικά και άλλα ραντεβού.
- ☛ Να παρέχει ευχάριστο περιβάλλον στο χρήστη όποια σχέση και αν έχει αυτός με την τεχνολογία. Πρέπει να την χαρακτηρίζει η απλότητα, να παρέχει απαραίτητες λειτουργίες μόνο, ώστε να είναι εύχρηστη και ξεκάθαρη.

Με βάση τις παραπάνω επιθυμίες και απαιτήσεις, αλλά και δικών μας πρωτοβουλιών προέκυψε μια εφαρμογή, τα χαρακτηριστικά της οποίας, καθώς και τρόπος ανάπτυξης και λειτουργίας της, αναλύονται στα κεφάλαια που ακολουθούν.

2 Εργαλεία λογισμικού

Πρόκειται για μια εφαρμογή, η οποία απευθύνεται σε ιατρούς, οι περισσότεροι από τους οποίους δεν έχουν ιδιαίτερη σχέση με τη χρήση του υπολογιστή. Οπότε απαιτούν εύχρηστο περιβάλλον με εύκολη μετακίνηση στις διάφορες λειτουργίες της εφαρμογής. Επιπλέον απαιτούν ευχάριστη και γρήγορη πλοήγηση μεταξύ των οθονών που αποτελούν την εφαρμογή.

Για να επιτευχθούν οι παραπάνω απαιτήσεις, κρίνεται απαραίτητη η χρήση μιας γλώσσας προγραμματισμού η οποία συνοδεύεται από ένα εργαλείο γραφικής ανάπτυξης εφαρμογής (για παράδειγμα Visual Basic, Delphi, Visual C++) και επιτρέπει στον προγραμματιστή την ανάπτυξη εφαρμογών για κάθε χρήστη.

Για κάθε εφαρμογή, πλέον, κρίνεται απαραίτητη η χρήση κάποιου εργαλείου Βάσεων Δεδομένων. Η βάση δεδομένων είναι κομμάτι σχεδόν κάθε εφαρμογής. Στην εφαρμογή μας η βάση Δεδομένων είναι απαραίτητη για τη διατήρηση αρχείων όσον αφορά τους ασθενείς και τις επισκέψεις τους, τα ραντεβού, και τους συνεργάτες των ιατρών.

Για να επιτύχουμε το καλύτερο δυνατό αποτέλεσμα θεωρήσαμε καταλληλότερη γλώσσα προγραμματισμού την γλώσσα προγραμματισμού Delphi για την ανάπτυξη της διεπαφής (interface) της εφαρμογής μας, και τον SQL Server για την υποστήριξη της Βάσης Δεδομένων της.

2.1 Γιατί Delphi;

Η Delphi είναι μια ευέλικτη, μεγάλου εύρους γλώσσα αντικειμενοστραφούς προγραμματισμού και έχει σημαντικά χαρακτηριστικά που κάνουν πιο εύκολη την ανάπτυξη εφαρμογών για σύγχρονα αλληλεπιδραστικά δικτυακά περιβάλλοντα που την κάνουν ιδιαίτερα ανταγωνιστική σε σχέση με άλλες αντικειμενοστραφής γλώσσες.

Η Delphi είναι γνωστό ότι αποτελεί έναν όχι και τόσο μακρινό απόγονο της Object Pascal. Έτσι, κατά βάση θυμίζει πολύ τις Object - Oriented (Αντικειμενοστραφείς) γλώσσες όπως τη Visual Basic, όπου υπάρχει ένα γραφικό περιβάλλον ανάπτυξης «κάτω» από το οποίο λειτουργεί η εκάστοτε γλώσσα προγραμματισμού, που στην περίπτωσή μας, όπως ήδη αναφέρθηκε, είναι η Object Pascal.

Ο προγραμματιστής έχει στην διάθεσή του ένα πλήθος εργαλείων τα οποία καθιστούν δυνατή την σχεδίαση ενός interface κατάλληλο για κάθε χρήστη. Παλαιότερα, θα έπρεπε να γράψουμε αρκετό κώδικα μόνο και μόνο για να εμφανίσουμε και να ελέγξουμε ένα απλό και «άδειο» παράθυρο στην οθόνη του υπολογιστή. Σήμερα όμως, ελάχιστα κλικ μας δίνουν την δυνατότητα να εξοικονομήσουμε αρκετό χρόνο και να περάσουμε γρήγορα στην ενασχόληση με το πιο ουσιαστικό τμήμα της εφαρμογής μας.

Πρέπει όμως να τονίσουμε ώστε να γίνει κατανοητό πως όσο εύκολα και να γίνουν τα περιβάλλοντα των γλωσσών προγραμματισμού, όσα εργαλεία και να μας προσφέρονται έτοιμα, αν δεν προσθέσουμε δικό μας κώδικα, εφαρμογή δεν γίνεται!

2.2 Γιατί SQL Server;

Ο SQL Server είναι ένα πολύ δυνατό και αξιόπιστο σύστημα διαχείρισης Σχεσιακών Συστημάτων Βάσεων Δεδομένων, (πράγμα απαραίτητο για εφαρμογές που αφορούν σημαντικούς τομείς όπως την ζωή και την υγεία των ανθρώπων), για κατανεμημένες εργασίες πελάτη / διακομιστή πάνω σε σύγχρονα δικτυακά περιβάλλοντα.

Η ιδέα των Σχεσιακών Συστημάτων Βάσεων Δεδομένων εισήχθη αρχικά από τον E. F. Codd [7], σε ένα άρθρο που έκτοτε αποτελεί σημείο αναφοράς για όλη την έρευνα και ανάπτυξη αυτών των συστημάτων. Σε αντίθεση με τα προηγούμενα συστήματα βάσεων δεδομένων (Δικτύου και ιεραρχικά), τα συστήματα Σχεσιακών Βάσεων Δεδομένων βασίζονται στο σχεσιακό μοντέλο δεδομένων, που έχει ένα δυνατό μαθηματικό υπόβαθρο.

Ένα μοντέλο Δεδομένων είναι μια συλλογή αρχών, σχέσεων και περιορισμών που χρησιμοποιούνται για να αναπαραστήσουν τα δεδομένα ενός πραγματικού προβλήματος.

Η βασική αρχή του σχεσιακού μοντέλου δεδομένων είναι η σχέση, δηλαδή ένας πίνακας. Έτσι, από την άποψη του χρήστη, μια σχεσιακή βάση δεδομένων περιέχει πίνακες και μόνο πίνακες. Μέσα σε ένα πίνακα, υπάρχουν μια οι περισσότερες στήλες και μηδέν ή περισσότερες γραμμές. Σε κάθε θέση γραμμής και στήλης (πεδίο) μέσα σε ένα πίνακα υπάρχει πάντα ακριβώς μια τιμή δεδομένων.

Η σχεσιακή γλώσσα του SQL Server ονομάζεται Transact - SQL. Είναι μια διάλεκτος της πιο σημαντικής γλώσσας Βάσεων Δεδομένων που υπάρχει σήμερα: της SQL (Structured Query Language - Δομημένη Γλώσσα Ερωτημάτων). Η προέλευση της SQL σχετίζεται στενά με το έργο, που καλείται System R, που αναπτύχθηκε και υλοποιήθηκε από





την IBM στις αρχές τις δεκαετίας του 1980. το έργο αυτό έδειξε ότι είναι δυνατό, χρησιμοποιώντας τις θεωρητικές βάσεις του E. F. Codd, να δημιουργηθεί ένα Σύστημα Σχεσιακής Βάσης Δεδομένων. Μετά την επιτυχία του System R, αρκετές τέτοιες εταιρίες δημιούργησαν τα δικά τους τέτοια συστήματα χρησιμοποιώντας σαν γλώσσα την SQL. Για περισσότερα, αναφορά στο «Εισαγωγή Στα Συστήματα Βάσεων δεδομένων» – C. J. Date (Κλειδάριθμος).

2.3 Τι είναι το interface;

Το Interface είναι το τμήμα της εφαρμογής που «βρίσκεται» ανάμεσα στο Γιατρό (χρήστη της εφαρμογής) και τη βάση δεδομένων. Αποτελείται από ``οθόνες``, τις φόρμες (Forms) οι οποίες, με τη σειρά τους αποτελούνται από εργαλεία της Delphi, κατάλληλα τοποθετημένα έτσι ώστε το αποτέλεσμα να είναι ευχάριστο στο χρήστη και αποτελεσματικό, προκειμένου η εφαρμογή να καλύπτει τις ανάγκες του χρήστη.

2.4 Τι είναι Σύστημα Βάσης Δεδομένων

Ένα Σύστημα Βάσεων Δεδομένων δεν είναι παρά ένα σύστημα τήρησης εγγραφών με υπολογιστή. Είναι μία συλλογή συστατικών λογισμικού Βάσης Δεδομένων και Βάσεων Δεδομένων, που περιέχει τα παρακάτω:

-  Προγράμματα εφαρμογών βάσεων δεδομένων
-  Μετωπικά συστατικά (όπως πελάτες)
-  Συστήματα διαχείρισης Βάσεων Δεδομένων (DataBase Management System, DBMS)
-  Βάσεις Δεδομένων

Ένα πρόγραμμα εφαρμογής Βάσεων Δεδομένων είναι ένα λογισμικό ειδικής χρήσης που είναι σχεδιασμένο και υλοποιημένο από χρήστες ή υλοποιημένο από εταιρίες λογισμικού.


Αντίθετα, τα μετωπικά συστατικά είναι λογισμικό γενικής χρήσης για Βάσεις Δεδομένων, σχεδιασμένο και υλοποιημένο από μια εταιρία Βάσεων Δεδομένων ή παραδομένο σε λογισμικό τρίτων κατασκευαστών. Χρησιμοποιώντας προγράμματα εφαρμογών Βάσεων Δεδομένων και μετωπικά συστατικά, οι χρήστες μπορούν να διαχειριστούν και να υποβάλλουν ερωτήματα σε δεδομένα μέσα στη Βάση Δεδομένων.

Η εργασία ενός συστήματος διαχείρισης Βάσεων Δεδομένων (DBMS) να διαχειρίζεται δεδομένα μέσα σε μία Βάση Δεδομένων. Το DBMS διαχειρίζεται όλες της αιτήσεις των χρηστών για προσπέλαση της Βάσης Δεδομένων (Αίτηση για προσθήκη και αφαίρεση πινάκων, ανάκληση και ενημέρωση δεδομένων που είναι αποθηκευμένα σε τέτοιους πίνακες)[2].

Γενικά, μια Βάση Δεδομένων μπορεί να θεωρηθεί ένα είδος ηλεκτρονικής αρχειοθήκης. Με άλλα λόγια, ένας χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων. Τα δεδομένα κατατάσσονται σε πεδία τα οποία συγκεντρώνονται σε εγγραφές. Ένα σύνολο εγγραφών ομογενή δεδομένων αποτελούν ένα πίνακα. Στην εφαρμογή μας ένας πίνακας (Ασθενείς) διατηρεί δεδομένα που περιγράφουν τους ασθενείς του ιατρείου.

Κάθε εγγραφή σε έναν πίνακα πρέπει να είναι μοναδική. Δηλαδή στο παραπάνω παράδειγμα ο πίνακας (Ασθενείς) πρέπει να περιέχει κάποιον ασθενεί μόνο μια φορά. Την μοναδικότητα των εγγραφών την πετυχαίνουμε με την χρήση πεδίων κλειδιών.

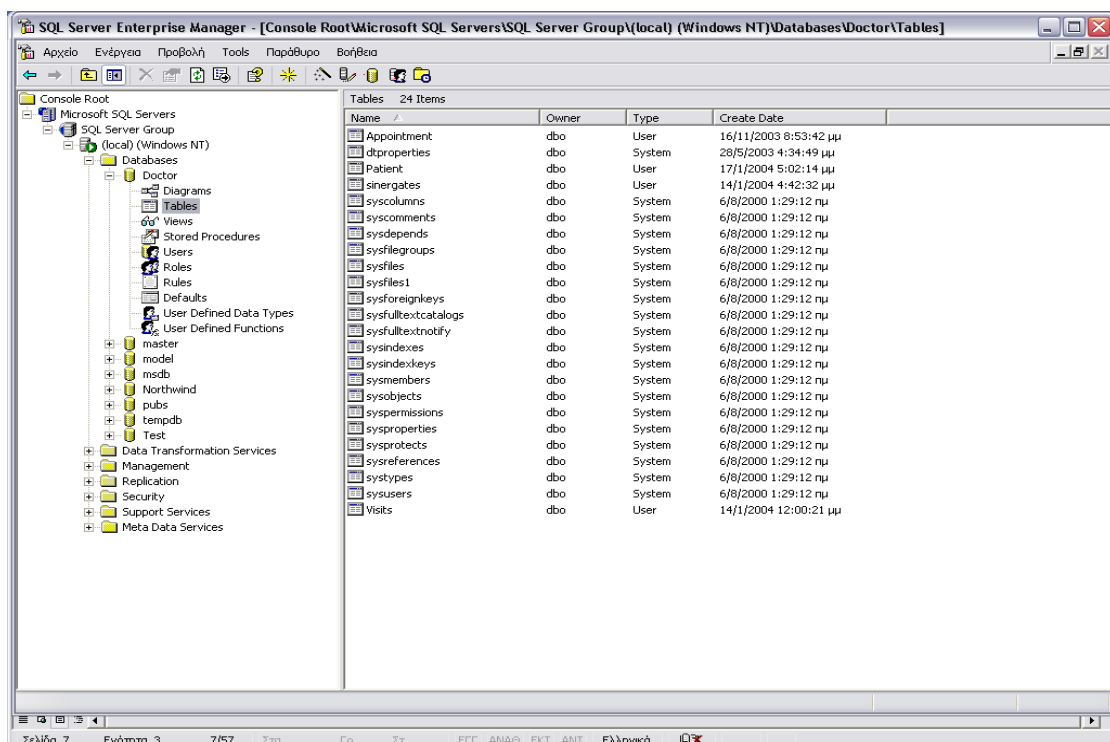
Τα δεδομένα σε μια Βάση Δεδομένων μπορούν να σχεδιαστούν εύκολα, χρησιμοποιώντας μόνο ένα πίνακα που να περιέχει όλα τα δεδομένα. Το κύριο μειονέκτημα μιας τέτοιας σχεδίασης Βάσης Δεδομένων είναι ο μεγάλος πλεονασμός των δεδομένων. Το μοντέλο οντότητας - σχέσης (Entity - Relation, ER) χρησιμοποιείται για να σχεδιάσει σχεσιακές Βάσης Δεδομένων καταργώντας όλο των πλεονασμό μέσα στα δεδομένα [1].



3 Ανάλυση Εφαρμογής

3.1 Δημιουργία Βάσης Δεδομένων

Το πρώτο βήμα στην ανάπτυξη της εφαρμογής μας ήταν να σχεδιάσουμε την βάση δεδομένων μας. Χρησιμοποιήσαμε τον Enterprise manager του SQL Server για την δημιουργία της Βάσης Δεδομένων, σύμφωνα με τις απαιτήσεις και τις ανάγκες των ιατρών που καταγράψαμε στο στάδιο της έρευνάς μας.



Η βάση δεδομένων που καλύπτει τις ανάγκες του ιατρού αποτελείται από τέσσερις πίνακες: Patient, Visits, Appointment, Sinergates.

- Ο πίνακας που θα κρατάει τα προσωπικά στοιχεία των ασθενών (*patient*), ο οποίος έχει τα εξής πεδία:

Περιγραφή	όνομα	τύπος
Κωδικός Ασθενούς	<i>Patient_Kode</i>	<i>bigint8</i>
Όνομα Ασθενούς	<i>Name</i>	<i>char15</i>
Επώνυμο Ασθενούς	<i>L_Name</i>	<i>char 15</i>
Όνομα Πατρός Ασθενούς	<i>F_Name</i>	<i>char 15</i>
Όνομα Συζύγου Ασθενούς	<i>S_Name</i>	<i>char 15</i>
Ημερομηνία Γέννησης	<i>BirthD</i>	<i>datetime 8</i>
Φύλο Ασθενούς	<i>Sex</i>	<i>char 7</i>
Επάγγελμα Ασθενούς	<i>Job</i>	<i>char 20</i>
Διεύθυνση Ασθενούς	<i>Address</i>	<i>char 30</i>
Τηλέφωνο Ασθενούς	<i>Phone</i>	<i>char 15</i>
Ασφάλεια Ασθενούς	<i>Ensurance</i>	<i>char 8</i>
Αριθμός Ασφάλισης	<i>Ensurance_Num</i>	<i>char 10</i>
Ιστορικό Ασθενούς	<i>Historic</i>	<i>char 99</i>
Very Important Case	<i>VIC</i>	<i>char 99</i>
Αριθμός Μητρώου	<i>[A.M]</i>	<i>char 10</i>
Οικογενειακή Κατάσταση	<i>Family</i>	<i>char 8</i>

Επειδή αυτός ο πίνακας περιέχει τα προσωπικά στοιχεία ασθενών, και κάθε ασθενής πρέπει να είναι μοναδικά αποθηκευμένος, έχουμε θέσει σαν πρωτεύον κλειδί το πεδίο *Patient_Kode*.

- Το πεδίο *Sex* μπορεί να πάρει δύο τιμές: Άνδρας, Γυναίκα.
- Το πεδίο *Family* μπορεί να πάρει δύο τιμές: Έγγαμος, Άγαμος.
- Στο πεδίο *VIC* ο ιατρός μπορεί να καταγράψει κάποια ιδιαιτερότητα που μπορεί να έχει κάποιος ασθενείς.

- Ο πίνακας που θα κρατάει τα χαρακτηριστικά των επισκέψεων των ασθενών (Visits), ο οποίος έχει τα εξής πεδία:

Περιγραφή	όνομα	τύπος
Κωδικό ασθενούς	Patient_Kode	bigint8
Ημερομηνία επίσκεψης	V_Date	datetime 8
Αριθμός επίσκεψης	Visit_Num	int 4
Συμπτώματα	Symptoms	char 100
Ιατρικές Εξετάσεις	Tests	char 100
Διάγνωση	Diagnose	char 100
Φαρμακευτική αγωγή	Medication	char 100

Επειδή ο πίνακας αυτός κρατάει όλες τις επισκέψεις όλων των ασθενών, και είναι πολύ πιθανό ένας ασθενής να έχει πάνω από μια επισκέψεις, την μοναδικότητα των εγγραφών την εξασφαλίζουν δύο κλειδιά: Το πεδίο Patient_Kode και το πεδίο Visit_Num.

- Ο πίνακας που θα κρατάει τα ραντεβού του ιατρείου (Appointment), ο οποίος έχει τα εξής πεδία:

Περιγραφή	όνομα	τύπος
Όνομα ασθενούς	Name	char 15
Επώνυμο ασθενούς	L_Name	char 15
Ημερομηνία	A_Date	datetime 8
Ωρα	A_Time	varchar 6
Τηλέφωνο ασθενούς	Phone	char 15

Ο πίνακας αυτός κρατάει τα ραντεβού του ιατρού και τα κλειδιά του πίνακα αυτού είναι η ημερομηνία και η ώρα, αφού για μια συγκεκριμένη ώρα μιας συγκεκριμένης ημέρας πρέπει να μπορεί να δεσμευθεί για ένα μόνο ραντεβού (μια εγγραφή με τις συγκεκριμένες τιμές στα αντίστοιχα πεδία *A_Date* και *A_Time*).

- Ο πίνακας που θα κρατάει τα προσωπικά στοιχεία των συνεργατών (*sinergates*), ο οποίος έχει τα εξής πεδία:

Περιγραφή	όνομα	τύπος
<i>A. Φ. Μ. Συνεργάτη</i>	<i>C_kode</i>	<i>char 9</i>
<i>Επώνυμο συνεργάτη</i>	<i>L_Name</i>	<i>char 15</i>
<i>Όνομα συνεργάτη</i>	<i>Name</i>	<i>char 15</i>
<i>Διεύθυνση συνεργάτη</i>	<i>Address</i>	<i>char 25</i>
<i>Ειδικότητα συνεργάτη</i>	<i>Sector</i>	<i>char 20</i>
<i>Ηλεκτρονική διεύθυνση</i>	<i>E-Mail</i>	<i>char 30</i>
<i>Ιστοσελίδα συνεργάτη</i>	<i>Site</i>	<i>char 30</i>
<i>Ιδιαιτερότητα συνεργάτη</i>	<i>[S.T.]</i>	<i>char 50</i>
<i>Τηλέφωνο συνεργάτη</i>	<i>Phone</i>	<i>Char 15</i>

Πεδίο κλειδί του πίνακα αυτού είναι το *C_Kode* στο οποίο μπαίνει το *A. Φ. Μ.*, το οποίο όπως γνωρίζουμε είναι μοναδικό για κάθε άνθρωπο.

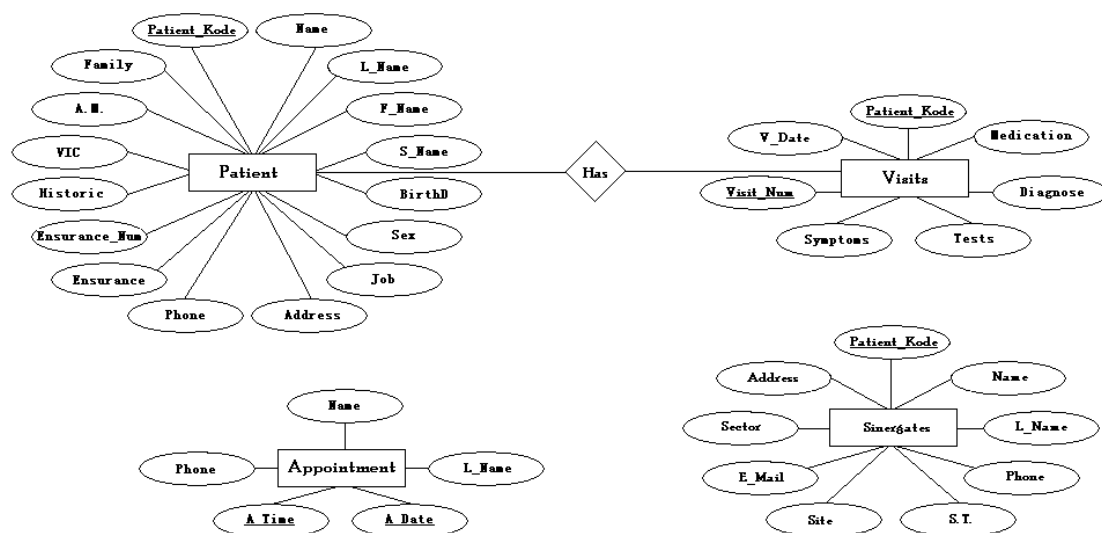
Το πεδίο *S.T.* (*Ιδιαιτερότητα συνεργάτη*) συμπληρώνεται εφόσον ο συγκεκριμένος συνεργάτης πρέπει να αντιμετωπιστεί με ιδιαίτερο τρόπο ή έχει κάποια ιδιαιτερότητα την οποία ο ιατρός θα ήθελε να καταγράψει.

3.2 Το μοντέλο Ε. Ρ.

Στην εφαρμογή μας τα στοιχεία των ασθενών θα μπορούσαν να διατηρούνται σε ένα μόνο πίνακα. Στην περίπτωση αυτή όμως θα είχαμε μεγάλο πλεονασμό δεδομένων. Για παράδειγμα, ο πίνακας «Ασθενείς» θα περιείχε δεδομένα που αφορούν τους ασθενείς και τις επισκέψεις τους. Κάθε ασθενής όμως έχει μια ή περισσότερες επισκέψεις. Για να καταγράψουμε λοιπόν όλες τις επισκέψεις θα επαναλαμβάνονταν τα προσωπικά στοιχεία κάθε ασθενούς, πιθανότατα πάνω από μια φορά (όσες είναι και οι επισκέψεις του).

Το Ε. Ρ. (Entity - Relation) έρχεται να λύσει αυτό το πρόβλημα με την σύνδεση πινάκων. Δημιουργούμε λοιπόν ένα πίνακα «επισκέψεις», στον οποίο κρατούνται οι επισκέψεις και τον συνδέουμε με τον πίνακα ασθενείς με το πεδίο κλειδί Patient_kode. Αυτό σημαίνει πως στον πίνακα «Ασθενείς» κρατούνται τα προσωπικά στοιχεία των ασθενών, και στον πίνακα «επισκέψεις» το μόνο που επαναλαμβάνεται είναι ο κωδικός ασθενούς.

Το μοντέλο Ε. Ρ. της εφαρμογής μας είναι το εξής:



Μοντέλο Ε. Ρ.

3.3 Σύνδεση Βάσης με Interface

Η μέθοδος της σύνδεσης της Βάσης Δεδομένων με το interface της εφαρμογής που διαλέξαμε είναι η χρήση ψευδωνύμων. Τα ψευδώνυμα δημιουργήθηκαν από την ανάγκη των προγραμματιστών να μην αλλάζουν ρυθμίσεις στις εφαρμογές τους κάθε φορά που θα άλλαζε η βάση δεδομένων. Πλέον, κάθε αλλαγή στην Βάση Δεδομένων γίνεται γνωστή στη εφαρμογή μέσω των ψευδωνύμων.

Απαιτήθηκε η δημιουργία δύο ψευδωνύμων, ενός ODBC (Open DataBase Connectivity) και ενός BDE (Borland Database Engine).

Το ODBC έγινε γνωστό στην δεκαετία του 1990 για την εγκατάσταση ενός για τις εφαρμογές προκειμένου να επικοινωνούν με Βάσεις Δεδομένων. Ο στόχος που υπάρχει πίσω από το ODBC είναι ότι οι εφαρμογές είναι γραμμένες στην διεπαφή ODBC και ότι οι προγραμματιστές μπορούν να αλλάξουν τις μηχανές Βάσεις Δεδομένων χωρίς να αλλάζουν κώδικα. Αυτή η δυνατότητα και ευελιξία αποτελούν ένα πολύ καλό λόγο για τη χρήση της συνδετικότητας ODBC [3].

Το BDE είναι μια διεπαφή που παρέχει υποστήριξη εγγενών Βάσεων Δεδομένων για εφαρμογές Inprise, συμπεριλαμβανομένου και του Delphi. Ο Borland Database Administrator παρέχει διαχείριση διάρθρωσης (Configuration) για την BDE.

3.3.1 Δημιουργία ODBC Ψευδωνύμου

- Από τον πίνακα ελέγχου (Control Panel) κάναμε διπλό κλικ στη μίνι εφαρμογή (Applet) Data Sources για να εκτελέσουμε τον ODBC Data Source Administrator (c:/winnt/system32/odbcad32.exe), όπου c:/winnt είναι ο κατάλογος στον οποίο έχουμε εγκαταστήσει τα Windows.
- Για να δημιουργήσουμε ένα ψευδώνυμο ODBC για το δικό μας προφίλ χρήστη (για την δική μας μόνο σύνδεση στα Windows), επιλέγουμε την καρτέλα User DNS. Για να δημιουργήσουμε ένα ψευδώνυμο ODBC για όλους τους χρήστες, επιλέγουμε την καρτέλα System DNS.
- Για την εκκίνηση της δημιουργίας του ψευδωνύμου κάναμε κλικ στο κουμπί Add και ανοίγει το παράθυρο διαλόγου του οδηγού (wizard) Create New Database Source.
- Το παράθυρο διαλόγου Create New Database Source παραθέτει σε λίστα όλα τα προγράμματα οδήγησης (Drivers) που έχουμε εγκαταστήσει. Κάθε Driver αναπαριστά μια διαφορετική Βάση Δεδομένων και έκδοση. Εμείς βρήκαμε και επιλέξαμε τον Microsoft SQL Server driver.
- Κάναμε κλικ στο Finish. Με το βήμα αυτό άνοιξε το παράθυρο διαλόγου ODBC Setup ειδικά για το driver που επιλέξαμε.
- Συμπληρώσαμε τα στοιχεία που μας ζητήθηκαν, βήμα βήμα, σε όλα τα παράθυρα που ακολουθούν και κλείνουμε τον ODBC Administrator.

3.3.2 Δημιουργία BDE Ψευδωνύμου

Ο BDE Administrator βρίσκεται στην μίνι εφαρμογή Control Panels και δίνει την δυνατότητα να καθορίσουμε ψευδώνυμα που χρησιμοποιούν τα Native BDE Databases drivers (προγράμματα οδήγησης) για κάθε βάση δεδομένων για την οποία υπάρχει υποστήριξη καθώς και για τα ψευδώνυμα ODBC.

Ένα ψευδώνυμο BDE δημιουργείται με το μενού Database, Explore της Delphi.

- ✦ Στο παράθυρο που εμφανίζεται (SQL Explorer), από το μενού Object επιλέξαμε New και άνοιξε το παράθυρο διαλόγου New Database Alias.
- ✦ Από το σύνθετο πλαίσιο που εμφανίζεται, επιλέξαμε SQL Server και OK.
- ✦ Στην καρτέλα Definition πληκτρολογήσαμε Doctor (το ψευδώνυμό μας) δίπλα στην προτροπή ODBC DNS.
- ✦ Μετονομάσαμε το ψευδώνυμο ODBC1 σε DoctorODBC (δικό μας ψευδώνυμο).
- ✦ Από το μενού Object επιλέξαμε Apply για να εφαρμόσουμε τις αλλαγές.
- ✦ Από το μενού Object επιλέξαμε Open για να ελέγξουμε εάν το ψευδώνυμο μπορεί να χρησιμοποιηθεί για το άνοιγμα της Βάσης Δεδομένων.
- ✦ Αφού βεβαιωθήκαμε για την κατάλληλη ρύθμιση του ψευδωνύμου κάναμε κλικ στο Object - Close για το κλείσιμο της Βάσης Δεδομένων.

- ✦ Έτσι έχουν ρυθμιστεί και τα δύο ψευδώνυμα, ODBC και BDE.

Μετά από αυτές τις Διαδικασίες ξεκινήσαμε την σχεδίαση και τον προγραμματισμό του *interface* της εφαρμογής.

3.4 Οι Φόρμες της Εφαρμογής

Οι φόρμες της εφαρμογής μας, συνολικά είναι δέκα:

- ⊙ Η αρχική φόρμα, η οποία παρουσιάζεται με την εκκίνηση της εφαρμογής.
- ⊙ Η φόρμα «Ασθενείς», στην οποία αναφέρεται ο χρήστης όταν θέλει να κάνει κάποια ενέργεια που αφορά κάποιον ή κάποιους ασθενείς.
- ⊙ Η φόρμα «Ραντεβού», η οποία αναφέρεται σε ενέργειες που αφορούν τηλεφωνικά και άλλα ραντεβού.
- ⊙ Η φόρμα «Στατιστικά Στοιχεία Ασθενών» στην οποία ο χρήστης μπορεί να αναφερθεί όταν θέλει γενικές πληροφορίες για το σύνολο των ασθενών του.
- ⊙ Η φόρμα «Συνεργάτες» στην οποία αναφέρεται ο χρήστης όταν θέλει να κάνει κάποια ενέργεια που αφορά κάποιον ή κάποιους Συνεργάτες, όπως ακτινολογικά και μικροβιολογικά εργαστήρια, άλλοι γιατροί, προμηθευτές αναλώσιμων και πάγιων υλικών και άλλους.

- Ⓢ Η φόρμα «Εύρεση Συνεργατών» η οποία παρέχει στο χρήστη όλα τα προσωπικά στοιχεία ενός ή περισσότερων συνεργατών.
- Ⓢ Η φόρμα «Επισκέψεις» η οποία προσφέρει λεπτομέρειες για τις επισκέψεις ενός ασθενούς.
- Ⓢ Η φόρμα «Νέα Επίσκεψη», η οποία επιτρέπει την καταγραφή των χαρακτηριστικών της επικείμενης ασθένειας του συγκεκριμένου ασθενούς.
- Ⓢ Η φόρμα «Εύρεση Ασθενών», η οποία επιτρέπει εύρεση ενός ή περισσότερων ασθενών.
- Ⓢ Η φόρμα «Προσωπικά Στοιχεία Ασθενούς» παρέχει στο χρήστη όλα τα προσωπικά στοιχεία ενός ασθενούς.

3.5 Τα Στοιχεία Της Εφαρμογής

Στο περιβάλλον της Delphi υπάρχουν καρτέλες στην παλέτα εξαρτημάτων στις οποίες υπάρχουν εξαρτήματα τα οποία προσφέρονται στον προγραμματιστή για την δημιουργία μιας ευέλικτης εφαρμογής. Κάθε καρτέλα περιέχει και μία διαφορετική κατηγορία στοιχείων.

Στην εφαρμογή μας χρησιμοποιήσαμε:

- ▶ Από την καρτέλα Standard τα στοιχεία: **Label, Button, Edit, RadioButton, RadioGroup, CheckBox.**

Η καρτέλα αυτή περιέχει τα γραφικά στοιχεία ελέγχου που χρησιμοποιούνται περισσότερο στα προγράμματα.

Label

Το εξάρτημα (στοιχείο) `label` είναι ένα στοιχείο το οποίο απεικονίζει κείμενο ή εικόνα. Μας επιτρέπει να εμφανίζουμε προτροπές και μηνύματα στους χρήστες. Με τις ετικέτες εμφανίζονται συμβολοσειρές που δεν μπορεί να αλλάξει ο χρήστης.

Button

Είναι ένα κουμπί ελέγχου το οποίο όταν πιέζεται εκτελεί τον κώδικα που βρίσκεται πίσω του.

Edit

Το στοιχείο αυτό δίνει στον χρήστη της εφαρμογής ένα χώρο όπου μπορεί να πληκτρολογήσει τα δεδομένα του πάνω στην φόρμα. Το εξάρτημα αυτό εμφανίζεται σαν πλαίσιο εισαγωγής.

Radiobutton

Είναι ένα στοιχείο ελέγχου το οποίο παρέχει στον χρήστη την δυνατότητα επιλογής.

RadioGroup

Το στοιχείο αυτό αναπαριστά ένα σύνολο `RadioButtons` τα οποία λειτουργούν σαν ομάδα. Κάθε φορά μόνο ένα `RadioButton` μπορεί να είναι επιλεγμένο.

CheckBox

Παρέχει στον χρήστη την δυνατότητα επιλογής. Το στοιχείο αυτό μπορεί να είναι επιλεγμένο ή όχι επιλεγμένο.

- ▶ Από την καρτέλα win32 τα στοιχεία: **MonthCalendar**, **RichEdit**.

MonthCalendar



Το στοιχείο αυτό είναι ένα ημερολόγιο στο οποίο ο χρήστης μπορεί να επιλέξει μια ή περισσότερες ημερομηνίες.

RichEdit



Αυτό το στοιχείο διαφέρει από το Edit μόνο στο ότι παρέχει χώρο, περισσότερες από μια γραμμές για τα δεδομένα.

- ▶ Από την καρτέλα Data Controls τα στοιχεία: **DBEdit**, **DBRichEdit**, **DBGrid**, **DBComboBox** και **DBNavigator**.

Η διαφορά των στοιχείων που αρχίζουν από DB από τα άλλα είναι ότι επιδρούν άμεσα με τις τιμές τους στη βάση δεδομένων, ανακτώντας ή αλλάζοντας τα δεδομένα, ενώ τα άλλα είναι ανεξάρτητα στη χρήση τους, δηλαδή δεν επικοινωνούν με τη βάση δεδομένων παρά μόνο έμμεσα, αφού πρώτα ο προγραμματιστής διαχειριστεί τις τιμές τους κατάλληλα, αν βέβαια το επιθυμεί.

Π.χ. ένα Label μπορεί να χρησιμοποιείται απλώς για να αναφέρει το όνομα της ενέργειας που κάνουμε («καταχώρηση Ασθενών»), χωρίς να χρειάζεται σύνδεση με κάτι.

DBEdit

Αυτό το στοιχείο έχει άμεση αλληλεπίδραση με τη Βάση δεδομένων. Δίνει καθώς και παίρνει τιμές άμεσα από ένα πεδίο της Βάσης.

DbrichEdit

Αυτό το στοιχείο διαφέρει από το `dbedit` μόνο στο ότι παρέχει χώρο, περισσότερες από μια γραμμές για τα δεδομένα.

DbGrid

Το στοιχείο αυτό εμφανίζει σύνολα δεδομένων από ένα πίνακα ή ερώτημα και μας δίνει την δυνατότητα να τα χειριστούμε (Εισαγωγή, Διαγραφή, Διόρθωση).

DBComboBox

Παρέχει στον χρήστη τον χώρο για να εισάγει δεδομένα πληκτρολογώντας ή επιλέγοντας τα από λίστα.

DBNavigator

Το στοιχείο αυτό παρέχει στον χρήστη την δυνατότητα να ενημερώνει, να εισάγει και να διαγράφει εγγραφές στη Βάση Δεδομένων καθώς και να κινείται μέσα σε αυτές. Τα αποτελέσματα φαίνονται σε ένα `dbgrid` ή σε `dbedit`s, τα οποία είναι συνδεδεμένα με το στοιχείο `navigator`.

- ▶ Από την καρτέλα BDE τα στοιχεία: **Table, Query, UpdatesQL.**

Η καρτέλα αυτή περιέχει στοιχεία που δεν είναι εμφανή στο χρήστη κατά την διάρκεια της εκτέλεσης της εφαρμογής. Είναι εμφανή μόνο κατά τη σχεδίαση της εφαρμογής.

Τα στοιχεία αυτά μας επιτρέπουν να χρησιμοποιούμε τα δεδομένα στην βάση δεδομένων μας.

Table



Αντιπροσωπεύει στο interface της εφαρμογής ένα πίνακα δεδομένων της Βάσης.

Query



Το στοιχείο αυτό δεσμεύει σύνολα εγγραφών από έναν ή περισσότερους πίνακες σύμφωνα με την SQL πρόταση.

UpdatesQL



Το στοιχείο αυτό εφαρμόζει κρυφές ενημερώσεις εκ μέρους ερωτημάτων ή διαδικασιών τα οποία δεν μπορούν να εκτελεστούν απευθείας.

- ▶ Από την καρτέλα Data Access τα στοιχεία: **DataSource**

Η καρτέλα αυτή περιέχει στοιχεία τα οποία επιτρέπουν στα στοιχεία των καρτελών BDE και Data Control να έχουν πρόσβαση στη Βάση Δεδομένων μας.

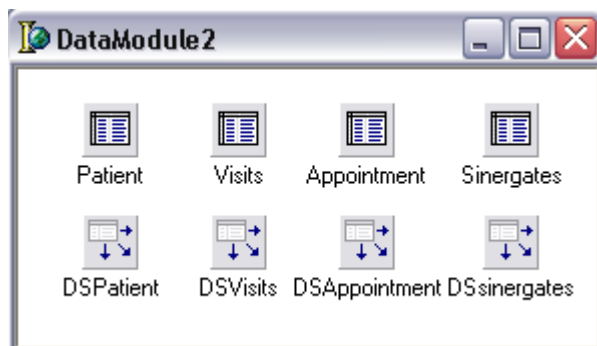
DataSource

Το στοιχείο αυτό είναι ο αγωγός μεταξύ της Βάσης Δεδομένων και των db στοιχείων της φόρμας.

3.6 Το Αντικείμενο DataModule

Στο DataModule μπορούμε να συγκεντρώσουμε στοιχεία σύνδεσης Βάσης Δεδομένων (όπως: Table, Datasource, Query) και να τα χρησιμοποιήσουμε σε όλες τις φόρμες. Το datamodule δημιουργείται από το περιβάλλον της Delphi μέσα από το μενού File - New - Datamodule.

Το DataModule που χρησιμοποιήσαμε στην εφαρμογή μας είναι το παρακάτω:



Όλα τα παραπάνω στοιχεία έχουν μεθόδους και ιδιότητες τις οποίες μπορούμε να διαχειριστούμε είτε από μια λίστα (Object Inspector) στο περιβάλλον της Delphi, είτε με κώδικα.

3.7 Συναρτήσεις

Η Delphi μας παρέχει μια πληθώρα διαδικασιών και συναρτήσεων που μπορούμε να χρησιμοποιήσουμε στις εφαρμογές μας. Στην εφαρμογή μας χρησιμοποιήσαμε τις συναρτήσεις:

✦ **AnsiUpperCase()**

Είναι μια συνάρτηση η οποία μετατρέπει όλους τους Ansi χαρακτήρες σε κεφαλαία. Αυτό είναι απαραίτητο σε ελέγχους.

✦ **DateToStr()**

Είναι μια συνάρτηση η οποία μετατρέπει κάθε ημερομηνία σε δεδομένα τύπου `string`, πράγμα χρήσιμο για συγκρίσεις.

✦ **StrToDate()**

Μοιάζει με την `DateToStr()` μόνο που μετατρέπει ένα `string` σε ημερομηνία (τύπου `Date`).

✦ **Date()**

Η συνάρτηση αυτή επιστρέφει την ημερομηνία του συστήματος.

✦ **Int()**

Η συνάρτηση αυτή μας επιστρέφει το ακέραιο μέρος ενός δεκαδικού αριθμού.

3.8 Σημάδια Θέσεων (Bookmarks)

Χρησιμοποιήσαμε επίσης σημάδια θέσεων (**Bookmarks**). Ένα σημάδι θέσης είναι ένας δείκτης εγγραφής που μας επιτρέπει να επιστρέψουμε σε μία θέση, δηλαδή σε μία συγκεκριμένη εγγραφή.

Τα σημάδια θέσεων τα διαχειριζόμαστε χρησιμοποιώντας τις εξής μεθόδους:

- ✚ Για να δεσμεύσουμε ένα σημάδι θέσης χρησιμοποιούμε τη μέθοδο **GetBookmark**.
- ✚ Για να επιστρέψουμε σε ένα σημάδι θέσης χρησιμοποιούμε τη μέθοδο **GotoBookmark**.
- ✚ Όταν δε χρειαζόμαστε πλέον το σημάδι θέσης, χρησιμοποιούμε τη μέθοδο **FreeBookmark**.

Οι μέθοδοι αυτές είναι μέθοδοι των στοιχείων που διαχειρίζονται εγγραφές, για παράδειγμα ενός πίνακα (Table). Είναι πολύτιμες για την επιστροφή μας σε μια εγγραφή μετά από κάποια έρευνα (έλεγχο).

Παράδειγμα δέσμευσης μιας θέσης:

```
Table1.GotoBookmark(point1);
```

4 Περαιτέρω Ανάλυση της Κάθε Φόρμας

4.1 Φόρμα «Ιατρείο»

Η φόρμα Ιατρείο είναι η κεντρική φόρμα της εφαρμογής μας. Είναι αυτή που εμφανίζεται με την εκκίνηση της εφαρμογής.

The screenshot shows a window titled 'ιατρείο' with a red close button. The main title is 'Ηλεκτρονική Διαχείριση Ιατρείου'. Below it, the section 'Προσωπικά Στοιχεία Ιατρού' (Personal Details of Doctor) is displayed. The details include: 'Φλώρου - Αθανασίου Βασιλική', 'Χειρουργός / Οδοντίατρος', 'Παπαστράτου 8, Αγρίνιο', and '26410 27705 - 6973984678'. At the bottom left, there is an 'Έξοδος' (Exit) button with a left arrow. On the right side, under the heading 'Επιλογές' (Options), there are four buttons with right arrows: 'Ασθενείς' (Patients), 'Ραντεβού' (Appointments), 'Στατιστικά Στοιχεία Ασθενών' (Patient Statistics), and 'Συνεργάτες' (Collaborators).

Φόρμα "Ιατρείο"

Σ' αυτή τη φόρμα παρουσιάζονται τα στοιχεία του ιατρού που χρησιμοποιεί την εφαρμογή, καθώς και οι αρχικές επιλογές που του παρέχονται.

Αποτελείται από:

- ✘ Πέντε κουμπιά (Buttons) τα οποία οδηγούν το χρήστη στις αντίστοιχες φόρμες, προκειμένου να εκτελέσει την ενέργεια που θέλει.
- ✘ Ετικέτες (Labels) οι οποίες υποδεικνύουν τις επιλογές και συμβάλλουν στη γενική εμφάνιση της φόρμας.

Οι επιλογές που παρέχονται στον χρήστη είναι: **Ασθενείς, Ραντεβού, Στατιστικά Στοιχεία Ασθενών, Συνεργάτες και Έξοδος:**

- ◆ Με το πάτημα του κουμπιού που βρίσκεται δίπλα στην ετικέτα "Ασθενείς" εμφανίζεται η φόρμα που περιέχει όλες τις επιλογές και τις λειτουργίες που αφορούν ασθενείς.
- ◆ Το κουμπί "Ραντεβού" οδηγεί στην φόρμα η οποία διαχειρίζεται τα ραντεβού του ιατρού.
- ◆ Το κουμπί "Στατιστικά Στοιχεία Ασθενούς" οδηγεί στη φόρμα η οποία παρέχει πληροφορίες για το σύνολο των ασθενών.
- ◆ Το κουμπί "Συνεργάτες" εμφανίζει την φόρμα η οποία περιέχει όλες τις επιλογές και τις λειτουργίες που αφορούν τους Συνεργάτες του ιατρού.
- ◆ Τέλος το κουμπί "Έξοδος" τερματίζει την εφαρμογή.

4.2 Φόρμα «Ασθενείς»

Η φόρμα "Ασθενείς" εμφανίζεται όταν επιλεγεί το κουμπί "Ασθενείς" στη φόρμα "Ιατρείο" ή όπου αλλού αυτό βρίσκεται.

Φόρμα "Ασθενείς"

Η φόρμα αυτή αποτελείται από:

- ✘ Πέντε Edits, στα οποία ο χρήστης της εφαρμογής καταγράφει το Όνομα, το Επώνυμο, το Πατρώνυμο και το Όνομα Συζύγου. Ο κωδικός του ασθενούς συμπληρώνεται αυτόματα.

- ✘ Έξι DBEdits τα οποία παίρνουν τις τιμές: την Ημερομηνία Γέννησης, το Επάγγελμα, τη Διεύθυνση, το Τηλέφωνο, τον Αριθμό Ασφάλισης και τον Αριθμό Μητρώου.
- ✘ Ένα DBComboBox στο οποίο καταγράφεται η Ασφάλιση του ασθενούς. Ο χρήστης μπορεί να επιλέξει από την λίστα του DBComboBox μία από τις τιμές: Ι. Κ. Α., Δημόσιο, Ο. Γ. Α., Τ. Ε. Β. Ε., καθώς και να πληκτρολογήσει άλλη ιδιωτική ή δημόσια ασφάλιση.
- ✘ Ένα RadioGroup αποτελούμενο από δύο RadioButtons τα οποία ανάλογα με το ποιο επιλέγεται δίνουν τιμή στο πεδίο φύλο (Άνδρας, Γυναίκα).
- ✘ Δύο DBRichEdits στα οποία ο χρήστης καταγράφει το Ιστορικό του ασθενούς (Αλλεργίες, Φυματίωση, Υπέρταση, Σακχαρώδης Διαβήτης, Ηπατίτιδα, Αφροδίσια και ασθένεια ή Εγχείρηση που είχε κάνει στο παρελθόν).
- ✘ Έξι κουμπιά τα οποία κατά την επιλογή τους οδηγούν τον χρήστη σε αντίστοιχες φόρμες: Επισκέψεις, Καταχώρηση, Άκυρο, Ραντεβού, Εύρεση Ασθενών και Αρχικό Μενού.

Πώς και Γιατί;

Όπως προαναφέραμε το πεδίο κλειδί των ασθενών είναι ο κωδικός. Επιλέξαμε αυτό το κλειδί γιατί κανένα άλλο πεδίο από μόνο του δε χαρακτηρίζει μια εγγραφή μοναδική. Επιπλέον ο ιατρός γνωρίζει ανά πάσα στιγμή το πλήθος των ασθενών του, μιας και ο κωδικός είναι αύξων αριθμός, και, αφού ο ιατρός δεν επιθυμεί την διαγραφή των ασθενών του, η μέθοδος αυτή δεν δημιουργεί κανένα λάθος ή πρόβλημα.

Ο κωδικός συμπληρώνεται μέσω κώδικα:

```
begin  
    datamodule2.Patient.Close;  
    datamodule2.Patient.Open;  
    datamodule2.Patient.First;  
    if datamodule2.Patient.Eof=true then  
        begin  
            c:=0;  
            a:=0;  
        end  
    else  
        begin  
            datamodule2.Patient.Last;  
            a:= datamodule2.Patient['patient_kode'];  
        end;  
    c:=1+a;  
    datamodule2.Patient.Insert;  
    edit1.text:=c;  
end;
```


Κάθε φορά που ένας ασθενείς θα εισέρχεται προ εξέταση στο ιατρείο και αφού ο χρήστης (ιατρός) μπει στη φόρμα ασθενείς, ζητά και καταγράφει τα προσωπικά στοιχεία του ασθενούς.

Καταγραφή των Προσωπικών Στοιχείων του Ασθενούς

Προσωπικά Στοιχεία Ασθενούς

Κωδικός:

Όνομα:

Επώνυμο:

Πατρώνυμο:

Όνομα Ισζύγου:

Ημερ. Γέννησης: * HH/MM/XX

Επάγγελμα:

Διεύθυνση:

Τηλέφωνο:

Ασφάλιση:

Αριθμός Ασφάλισης:

Αριθμός Μητρώου:

Φύλο

Γυναίκα

Άνδρας

Ιστορικό

* Αλλεργίες, Φυματίωση, Υπέρταση, Σ.Σ.ακχ, Διαβήτης, Ηπατίτιδα, Αφροδίσια, ...

V. I. C.

* Very Important Case

Διαθέσιμες Επιλογές

>> Επισκέψεις

>> Καταχώρηση

>> Άκυρο

>> Ραντεβού

>> Εύρεση Ασθενών

<< Αρχικό Μενού

Φόρμα "Ασθενείς"

Αφού συμπληρωθούν τα πέντε πρώτα στοιχεία (το πρώτο αυτόματα, και τα υπόλοιπα από τον χρήστη) γίνεται ένας έλεγχος σε όλες τις εγγραφές του πίνακα 'Patient'. Ο έλεγχος αυτός τοποθετήθηκε στο σημείο εισόδου του στοιχείου της ημερομηνίας γέννησης.


```
        d:=true;
        exit;
    end;
    datamodule2.Patient.Next;
end;
if d=true then
    begin
        datamodule2.Patient.GotoBookmark(B);
    end
else
    if d = false then
        Begin
            datamodule2.Patient.Last;
            a:= datamodule2.Patient['patient_kode'];
            datamodule2.Patient.Insert;
            f:=a+1;
            edit1.Text:=f;
            radiobutton1.Checked:=false;
            radiobutton2.Checked:=false;
            button1.Enabled:=true;
            button4.Enabled:=false;
        end;
    end;
end;
```

Συγκρίνονται οι τιμές των Edit στοιχείων: Όνομα, Επώνυμο, Πατρώνυμο, Όνομα συζύγου, με τις τιμές των αντίστοιχων πεδίων του πίνακα. Αν βρεθεί ομοιότητα σημαίνει ότι ο ασθενής είναι πελάτης του ιατρείου.

Ο έλεγχος αυτός γίνεται με σκοπό την αποφυγή εγγραφής του ίδιου ασθενούς πάνω από μια φορά. Διαφορετικά, αν δεν βρεθεί ομοιότητα, σημαίνει ότι ο ασθενής έρχεται για πρώτη φορά στο ιατρείο, ενεργοποιείται το κουμπί "Καταχώρηση" της φόρμας και συνεχίζουμε στην συμπλήρωση των προσωπικών του στοιχείων.

Τώρα ο χρήστης έχει δυο επιλογές:

- Να καταχωρήσει τον ασθενή πατώντας το κουμπί "Καταχώρηση" ή να σβήσει ότι έχει γράψει ως τώρα πατώντας το "Άκυρο".
- Με το πάτημα του κουμπιού καταχώρηση αποθηκεύεται ο ασθενείς στο πίνακα 'Patient' και εμφανίζεται η φόρμα «Νέα Επίσκεψη».

4.2.1 Φόρμα «Νέα Επίσκεψη»

Visits

Η Αιτία Επίσκεψης του Ασθενούς στο Ιατρείο (Ασθένεια)

Πληροφορίες Επίσκεψης	Καταγραφή της Ασθένειας
Πλήθος Επισκέψεων: <input type="text" value="1"/>	Συμπτώματα: <input type="text" value="Καταρροή, Πονοκεφαλούς"/>
Ημερομηνία: <input type="text" value="13/1/2004"/>	Εξετάσεις: <input type="text"/>
	Διάγνωση: <input type="text" value="Γρίπη"/>
	Φαρμ. Αγωγή: <input type="text" value="Contrex"/>

Πίσω <<

Φόρμα "Νέα Επίσκεψη"

Σε αυτή τη φόρμα έχουμε χρησιμοποιήσει:

- ✗ Ένα DBEdit, για το πλήθος των επισκέψεων ενός ασθενούς.
- ✗ Ένα Edit, για την ημερομηνία επίσκεψης, το οποίο παίρνει τιμή αυτόματα μέσω κώδικα χρησιμοποιώντας μια έτοιμη συνάρτηση, που μας

προσφέρει η Delphi, την Date η οποία μας επιστρέφει την ημερομηνία του συστήματος.

- ✘ Τέσσερα DBRichEdits, τα οποία παίρνουν τα συμπτώματα, τις Εξετάσεις, τη Διάγνωση και τη Φαρμακευτική Αγωγή.
- ✘ Ένα κουμπί, που οδηγεί στην φόρμα «Ασθενείς» και συγχρόνως αποθηκεύει την επίσκεψη του ασθενούς στον πίνακα 'Visits'.

Αν κατά την καταχώρηση των προσωπικών στοιχείων του ασθενούς αποδειχθεί ότι ο ασθενής έχει επισκεφθεί στο παρελθόν το ιατρείο, εμφανίζονται όλα τα προσωπικά του στοιχεία στη φόρμα ασθενείς, παραμένει απενεργοποιημένο το κουμπί 'Καταχώρηση' και ενεργοποιείται το κουμπί 'Επισκέψεις'.

Καταγραφή των Προσωπικών Στοιχείων του Ασθενούς

Προσωπικά Στοιχεία Ασθενούς

Κωδικός: 1

Όνομα: Ευθυμία

Επώνυμο: Βλάχου

Πατρώνυμο: Δημητρίου

Όνομα Ισζύγου:

Ημερ. Γέννησης: 6/6/1980 * ΗΗ/ΜΜ/ΧΧ

Επάγγελμα: ΦΟΙΤΗΤΡΙΑ

Διεύθυνση: ΚΟΛΑΚΟΤΡΩΝΙ 12 Δ ΠΑ

Τηλέφωνο: 2641027951

Ασφάλιση: I.K.A.

Αριθμός Ασφάλισης: ΚΟΛΑΚΟΤΡΩΝ

Αριθμός Μητρώου: 2641027951

Φύλο

Γυναίκα

Ανδρας

Ιστορικό

ΑΛΛΕΡΓΙΑ ΣΤΗΝ ΓΥΡΗ

* Αλλεργίες, Φυματίωση, Υπέρταση, Σ.Σ.ακχ, Διαβήτης, Ηπατίτιδα, Αφροδίσια, ...

V. I. C.

* Very Important Case

Διαθέσιμες Επιλογές

>> **Επισκέψεις**

>> **Καταχώρηση**

>> **Άκυρο**

>> **Ραντεβού**

>> **Εύρεση Ασθενών**

<< **Αρχικό Μενού**

Φόρμα "Ασθενείς"

Αυτό που έχει ο χρήστης να κάνει είναι να πατήσει το κουμπί 'Επισκέψεις' και ν' ανοίξει τη φόρμα «Επισκέψεις».

4.2.2 Φόρμα «Επισκέψεις»

Επισκέψεις

Προηγούμενες Επισκέψεις Ασθενούς και Δυνατότητα Διόρθωσής τους

Πληροφορίες Σχετικά με τις επισκέψεις του Ασθενούς: ευθυμια βλαχου Και Διόρθωσή τους

Κωδικός

Πλήθος Επισκέψεων

Ημερ. Επίσκεψης

Τίσω

Νέα Επίσκεψη

Συμπτώματα

Εξετάσεις

Διάγνωση

Φαρμ. Αγωγή

Φόρμα "Επισκέψεις"

Στη Φόρμα αυτή υπάρχουν τα εξής στοιχεία:


- ✘ Ένα DBNavigator, το οποίο επιτρέπει στον χρήστη την πλοήγησή του στις μέχρι σήμερα επισκέψεις του συγκεκριμένου ασθενούς, οι πληροφορίες των οποίων εμφανίζονται στα DBRichEdits τα οποία είναι συνδεδεμένα με το DBNavigator.
- ✘ Τέσσερα DBRichEdits, τα οποία εμφανίζουν τις πληροφορίες της κάθε επίσκεψης του συγκεκριμένου ασθενούς.
- ✘ Τρία DBEdits, τα οποία εμφανίζουν: τον Κωδικό, το Πλήθος επισκέψεων καθώς και την Ημερομηνία μιας συγκεκριμένης επίσκεψης.

- ✘ Δυο κουμπιά: Το κουμπί «Πίσω» κλείνει τη φόρμα «Επισκέψεις» και ανοίγει τη φόρμα «Ασθενείς». Το κουμπί «Νέα Επίσκεψη» κλείνει τη φόρμα «Επισκέψεις» και ανοίγει τη φόρμα «Νέα Επίσκεψη».

Πώς και Γιατί;

Μέσα στην φόρμα αυτή («Επισκέψεις») ο χρήστης έχει τις εξής δυνατότητες:

- Να δει τις πληροφορίες των επισκέψεων του ασθενούς ενημερωτικά, να πατήσει το κουμπί «Πίσω» και να κλείσει τη φόρμα.
- Να δει τις πληροφορίες των επισκέψεων του ασθενούς, να συγκεντρώσει πληροφορίες για το ιατρικό παρελθόν του, και να πατήσει το «Νέα Επίσκεψη».
- Με διπλό κλικ να διορθώσει κάποια ή κάποιες από τις πληροφορίες κάποιας επίσκεψης.

Στην τελευταία περίπτωση απενεργοποιείται το κουμπί «Νέα Επίσκεψη». Αυτό συμβαίνει γιατί είναι πολύ πιθανό κάποιος ασθενής να χρειαστεί να επισκεφθεί ξανά το ιατρείο για την ίδια ασθένεια όπως για επιστροφή αποτελεσμάτων ιατρικών εξετάσεων. Στην περίπτωση αυτή θα ήταν λάθος να δημιουργηθεί επιπλέον επίσκεψη για τον συγκεκριμένο ασθενή. Το μόνο που έχει να κάνει ο χρήστης είναι να κάνει διπλό κλικ στο DBRichEdit 'Εξετάσεις', να προσθέσει τα αποτελέσματα των εξετάσεων, να πατήσει το κουμπί «Καταχώρηση»  (το κουμπί ακριβώς δίπλα στο DBNavigator) και μετά το κουμπί «Πίσω».

Στη φόρμα «Ασθενής» ο χρήστης έχει επίσης την δυνατότητα να επιλέξει το κουμπί «Ραντεβού», το οποίο υπάρχει και στην προηγούμενη φόρμα («Ιατρείο») (γι' αυτό και θα αναλυθεί σε επόμενο κεφάλαιο) και οδηγεί στην φόρμα «Ραντεβού», καθώς και το κουμπί «Εύρεση ασθενών» και το κουμπί «Αρχικό Μενού» το οποίο οδηγεί στη φόρμα «Ιατρείο» όπου υπάρχει το αρχικό μενού της εφαρμογής μας.

Με το πάτημα του κουμπιού «Εύρεση Ασθενών» εμφανίζεται η φόρμα «Εύρεση Ασθενών».

4.2.3 φόρμα «Εύρεση Ασθενών»

Όνομα	Επώνυμο	Πατρώνυμο	Όνομα Συζύγου	Φύλο	Επάγγελμα	Ασφάλιση	Ιστορικό	VIC	Οικ. Κατάσταση
ΑΓΓΕΛΙΚΗ	ΑΒΑΝΑΣΙΟΥ	ΧΡΗΣΤΟΦΟΡΟΣ		ΓΥΝΑΙΚΑ	ΦΟΙΤΗΤΡΙΑ	ΔΗΜΟΣΙΟ	ΑΛΛΕΡΓΙΑ ΣΤΟ ΑΛΚΟΟΛ		ΑΓΑΜΟΣ
ΣΤΑΥΡΟΣ	ΑΝΑΣΤΑΣΙΑΔΗΣ	ΓΕΩΡΓΙΟΣ		ΑΝΔΡΑΣ	ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ	Ι.Κ.Α.			ΑΓΑΜΟΣ
ΕΥΘΥΜΙΑ	ΒΛΑΧΟΥ	ΔΗΜΗΤΡΗΣ		ΓΥΝΑΙΚΑ	ΦΟΙΤΗΤΡΙΑ	Ι.Κ.Α.	ΑΛΛΕΡΓΙΑ ΣΤΗΝ ΓΥΡΗ		ΑΓΑΜΟΣ
ΜΑΡΙΑ	ΔΑΚΑΝΑΛΗ	ΓΕΩΡΓΙΟΣ		ΓΥΝΑΙΚΑ	ΑΠΟΦΟΙΤΟΣ ΤΕΙ	Ι.Κ.Α.	ΥΠΟΤΑΣΗ		ΑΓΑΜΟΣ
ΚΙΚΗ	ΔΗΜΑ	ΧΡΗΣΤΟΦΟΡΟΣ		ΓΥΝΑΙΚΑ	ΚΑΘΗΓΗΤΡΙΑ	Ι.Κ.Α.			ΑΓΑΜΟΣ
ΝΕΦΕΛΗ	ΗΓΟΥΜΕΝΟΥ	ΓΕΩΡΓΙΟΣ		ΓΥΝΑΙΚΑ	ΔΗΜΟΣΙΟΣ ΥΠΑΛΛΗΛΟΣ	Ι.Κ.Α.			ΑΓΑΜΟΣ
ΓΕΩΡΓΙΑ	ΚΑΡΟΥΤΖΟΥ	ΓΕΩΡΓΙΟΣ		ΓΥΝΑΙΚΑ	ΔΗΜΟΣΙΟΣ ΥΠΑΛΛΗΛΟΣ	Ι.Κ.Α.			ΑΓΑΜΟΣ
ΓΕΩΡΓΙΟΣ	ΡΑΠΤΗΣ	ΣΤΑΥΡΟΣ		ΑΝΔΡΑΣ	ΕΠ'ΚΕΙΡΗΜΑΤΙΑΣ	Τ.Ε.Β.	ΕΛΚΟΣ ΣΤΟΜΑΧΟΥ		ΑΓΑΜΟΣ
ΣΑΒΒΑΣ	ΣΠΑΝΟΥΔΗΣ	ΛΟΥΚΑΣ		ΑΝΔΡΑΣ	ΔΗΜΟΣΙΟΓΡΑΦΟΣ	Ι.Κ.Α.			ΑΓΑΜΟΣ

Φόρμα "Εύρεση Ασθενών"

Η φόρμα αυτή αποτελείται από:

- ✘ Ένα Edit στο οποίο γράφουμε το επώνυμο του ασθενούς ή ασθενών που θέλουμε να βρούμε.

- ✘ Τρία κουμπιά. Το «Έρευνα» το οποίο σηματοδοτεί την εκκίνηση της έρευνας, το «Άκυρο» το οποίο αδειάζει το Edit και το DBGrid και το «Πίσω» το οποίο κλείνει τη φόρμα αυτή και ανοίγει την αμέσως προηγούμενη, την «Ασθενής».
- ✘ Ένα DBGrid στο οποίο εμφανίζονται τα αποτελέσματα της έρευνας, δηλαδή όλοι οι ασθενείς του ιατρείου με κοινό επώνυμο, το επώνυμο που δώσαμε στο Edit.

Πώς και Γιατί;

Η εύρεση γίνεται μέσω κώδικα SQL σε συνδυασμό με τα στοιχεία Query και DataSource. Ο κώδικας SQL είναι:

```
select * from Patient where L_name like :last order  
by L_Name
```

Στην παραπάνω πρόταση, η έκφραση ":last" υποδηλώνει ότι το ερώτημα είναι παραμετρικό. Αυτό σημαίνει πως η τιμή της μεταβλητής δίνεται κατά την εκτέλεση της εφαρμογής (Run Time) με τον παρακάτω κώδικα Delphi ο οποίος εκτελείται με το πάτημα του κουμπιού «Εύρεση»:

```
query1.Close;  
query1.Prepare;  
query1.Params[0].Value:=ansiuppercase(edit3.Text);  
query1.Open;
```

Μπορούμε αντί για ολόκληρο το επώνυμο, στο Edit να γράψουμε ένα μέρος του και στο τέλος τον χαρακτήρα "%" (ο χαρακτήρας "%" είναι ο χαρακτήρας 'μπαλαντέρ' της SQL).

Μ' αυτό τον τρόπο θα εμφανιστούν οι ασθενείς των οποίων το επώνυμο αρχίζει με τους χαρακτήρες που γράψαμε στο Edit πριν από τον χαρακτήρα "%". Αν γράψουμε μόνο τον χαρακτήρα "%", στο DBGrid θα εμφανιστούν όλοι οι ασθενείς.

Ο χρήστης μπορεί να επιλέξει κάποιον από αυτούς τους ασθενείς επιλέγοντάς τον από το DBGrid με ένα κλικ, εφόσον επιθυμεί να δει όλα τα προσωπικά του στοιχεία καθώς και της επισκέψεως του.

4.2.3.1 Φόρμα «Προσωπικά Στοιχεία Ασθενούς»

Κωδ. Ασθενούς	Ημ. Επίσκεψης	Αρ. Επίσκεψης	Συμπτώματα	Ιατρικές Εξετάσεις	Διάγνωση	Φαρμακευτική Αγωγή
2	13/1/2004	1	ΡΗΓΗ, ΖΑΛΗ	ΑΙΜΑΤΟΣ	ΓΡΙΠΗ	DEPON
2	14/1/2004	2	ΠΟΝΟΚΕΦΑΛΟΣ, ΠΟΝΟΣ ΣΤ		ΟΤΙΤΗΔΑ	ΑΜΟΧΙΛ, DEPON

Φόρμα "Προσωπικά Στοιχεία Ασθενούς"

Η φόρμα αυτή αποτελείται από:

- ✘ Δώδεκα Edits στα οποία εμφανίζονται τα προσωπικά στοιχεία του ασθενούς.

- ✘ Τέσσερα κουμπιά. Το «Διόρθωση» το οποίο επικυρώνει οποιαδήποτε αλλαγή κάνουμε σε οποιοδήποτε στοιχείο ή και σε οποιοδήποτε πεδίο του DBGrid, το «Νέα Επίσκεψη» το οποίο μας επιτρέπει να δημιουργήσουμε νέα επίσκεψη στον συγκεκριμένο ασθενή με την εμφάνιση της φόρμας «Νέα Επίσκεψη», το «Πίσω» το οποίο κλείνει τη φόρμα αυτή και ανοίγει την αμέσως προηγούμενη, την «Εύρεση ασθενών» και το κουμπί «Ασθενείς» το οποίο επιτρέπει άμεση μεταφορά στη φόρμα «Ασθενείς».
- ✘ Ένα DBGrid στο οποίο εμφανίζονται όλες οι επισκέψεις του ασθενούς που επιλέξαμε προηγουμένως.

Πώς και Γιατί;

Στη φόρμα αυτή παρουσιάζονται όλα τα προσωπικά στοιχεία του ασθενούς που ο χρήστης είχε επιλέξει στην προηγούμενη φόρμα «Εύρεση Ασθενούς». Εδώ ο χρήστης μπορεί να επέμβει (Διορθώσει) τα προσωπικά στοιχεία του ασθενούς καθώς και τις πληροφορίες των επισκέψεων του, οι οποίες εμφανίζονται στο DBGrid με την βοήθεια του στοιχείου Query.

Η πρόταση SQL που κρύβεται πίσω απ' το Query είναι:

```
select * from Visits where Patient_kode=:Patient_kode
```

Η έκφραση `:Patient_Kode` υποδηλώνει παραμετρικό ερώτημα το οποίο εκτελείται με την εμφάνιση της φόρμας, μέσω του κώδικα Delphi:

```
query1.Close;  
query1.Prepare;  
Edit1.Text:=FORM4.Query1.FieldValues['Patient_Kode'];  
Query1.Params[0].Value:=Edit1.Text;  
query1.Open;
```

* Η FORM4 είναι η φόρμα «Εύρεση Ασθενών»

Το κουμπί Διόρθωση επικυρώνει κάθε αλλαγή στα Edits με τον εξής κώδικα:

```
begin  
    datamodule2.Patient.Close;  
    datamodule2.Patient.Open;  
    datamodule2.Patient.First;  
    while not datamodule2.Patient.Eof do  
        begin  
            if datamodule2.Patient['patient_kode']=edit1.Text then  
                begin  
                    k:=datamodule2.Patient.getbookmark;  
                end;  
            datamodule2.Patient.Next;  
        end;  
end;
```



```
datamodule2.Patient.gotobookmark(k);
datamodule2.Patient.Edit;
datamodule2.Patient['name']:=AnsiUppercase(edit2.Text);
datamodule2.Patient['L_Name']:=AnsiUppercase(edit3.Text);
datamodule2.Patient['F_Name']:=AnsiUppercase(edit4.Text);
datamodule2.Patient['S_Name']:=AnsiUppercase(edit5.Text);
datamodule2.Patient['BirthD']:=AnsiUppercase(edit6.Text);
datamodule2.Patient['Job']:=AnsiUppercase(edit7.Text);
datamodule2.Patient['Address']:=AnsiUppercase(edit8.Text);
datamodule2.Patient['Phone']:=AnsiUppercase(edit9.Text);
datamodule2.Patient['Ensurance']:=AnsiUppercase(edit10.Text);
datamodule2.Patient['Ensurance_Num']:=AnsiUppercase
                                     (edit8.Text);
datamodule2.Patient['A.M']:=AnsiUppercase(edit9.Text);
datamodule2.Patient['Historic']:=AnsiUppercase
                                     (Richedit1.Text);
datamodule2.Patient['VIC']:=AnsiUppercase(Richedit2.Text);
datamodule2.Patient.Post;
end;
```

Το κουμπί Διόρθωση επικυρώνει κάθε αλλαγή στο DBGrid, μέσω του εξής Delphi κώδικα:

```
Query1.Database.ApplyUpdates([Query1]);
```

Και στο Event "OnUpdateRecord" του Query γράφουμε τον κώδικα Delphi:

```
procedure TForm3.Query1UpdateRecord(DataSet: TDataSet;
  UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
begin
  if ( UpdateKind = ukModify ) then
    begin
      UpdateSQL1.Apply(UpdateKind);
      UpdateAction := uaApplied;
    end
  else
    if ( UpdateKind = ukDelete ) then
      begin
        UpdateSQL1.Apply(UpdateKind);
        UpdateAction := uaApplied;
      end
    else
      begin
        UpdateAction := uaSkip;
        ShowMessage ('Error');
      end;
end;
```

Ο ακόλουθος κώδικας γράφεται στην ιδιότητα (Properties) ModifySQL του UpdateSQL:

```
update Visits set V_Date=:V_Date,
Visit_Num=:Visit_Num, Symptoms=:Symptoms,
Tests=:Tests, Diagnose=:Diagnose,
medication=:medication where
Patient_kode=:Patient_Kode and Visit_Num=:Visit_num
```

4.3 Φόρμα «Ραντεβού»

Η φόρμα αυτή εμφανίζεται κάθε φορά που ο χρήστης πατάει το κουμπί «Ραντεβού» το οποίο βρίσκεται στη αρχική φόρμα «Ιατρείο» και στην «Ασθενείς». Μέσα από την φόρμα αυτή ο χρήστης μπορεί να κλείσει κάποιο ραντεβού, να δει τα ραντεβού της ημέρας ή και να ακυρώσει κάποιο ραντεβού.

Ραντεβού

Κατάσταση των Ραντεβού του Ιατρείου

Ημερολόγιο

Ιανουάριος 2004

Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Σημερινές συνδέσεις: 14/1/2004

* Επιλογή Ημερομηνίας με Διπλό Κλικ του Ποντικιού Πάνω στην Ημερομηνία

Ασθενής

Όνομα: ΑΓΓΕΛΙΚΗ

Επώνυμο: ΑΒΑΝΑΣΙΟΥ

Τηλέφωνο: 2641025698

Διαθέσιμες Επιλογές

Επικύρωση Ενέργειας >>>

Πίσω <<<

Λίστα των Ενεργών Ραντεβού

<input type="checkbox"/>	09:00	<input type="checkbox"/>	18:00
<input type="checkbox"/>	09:30	<input type="checkbox"/>	18:30
<input type="checkbox"/>	10:00	<input type="checkbox"/>	19:00
<input type="checkbox"/>	10:30	<input type="checkbox"/>	19:30
<input checked="" type="checkbox"/>	11:00	<input type="checkbox"/>	20:00
<input type="checkbox"/>	11:30	<input type="checkbox"/>	20:30
<input type="checkbox"/>	12:00	<input type="checkbox"/>	21:00
<input type="checkbox"/>	12:30		
<input type="checkbox"/>	13:00		
<input type="checkbox"/>	13:30		
<input type="checkbox"/>	14:00		

Φόρμα "Ραντεβού"

Στη φόρμα αυτή χρησιμοποιούμε τα εξής στοιχεία:

- ✘ Ένα MonthCalendar, για την επιλογή ημερομηνίας.
- ✘ Τρία Edits, για την καταγραφή του Ονόματος, του Επωνύμου, του τηλεφώνου του ατόμου που θέλει να κλείσει κάποιο ραντεβού.

- ✘ Δεκαοκτώ CheckBoxes, για την επιλογή ώρας επίσκεψης που επιθυμεί ο ασθενής.
- ✘ Δυο κουμπιά, το «Επικύρωση Ενέργειας» το οποίο αποθηκεύει το ραντεβού, αν πρόκειται για νέο, ή το ακυρώνει δηλαδή το διαγράφει από τον πίνακα 'Appointments'.

Πώς και Γιατί;

Προκειμένου να κλείσει ο χρήστης κάποιο ραντεβού, κάνει διπλό κλικ πάνω στην ημερομηνία που επιθυμεί να κλείσει το ραντεβού και αμέσως εμφανίζονται όλα τα ενεργά ραντεβού εκείνης της ημέρας.

Κατάσταση των Ραντεβού του Ιατρείου

Ημερολόγιο
 < Ιανουάριος 2004 >
 Δευ Τρι Τετ Πεμ Παρ Σαβ Κυρ
 29 30 31 1 2 3 4
 5 6 7 8 9 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25
 26 27 28 29 30 31 1
 2 3 4 5 6 7 8
 Σημερινές συνδέσεις: 14/1/2004
 * Επιλογή Ημερομηνίας με Διπλό Κλικ του Ποντικισού Πάνω στην Ημερομηνία

Ασθενής

Όνομα
 Επώνυμο
 Τηλέφωνο

Διαθέσιμες Επιλογές

Επικύρωση Ενέργειας
 Πίσω

Λίστα των Ενεργών Ραντεβού

Ασθενής	Ωρα	Επιλογή
ΜΑΡΙΑ ΔΑΚΑΝΑΛΗ 6958458753	09:00	<input type="checkbox"/>
	09:30	<input type="checkbox"/>
	10:00	<input checked="" type="checkbox"/>
ΑΓΓΕΛΙΚΗ ΑΒΑΝΑΣΙΟΥ 2641065982	10:30	<input type="checkbox"/>
	11:00	<input checked="" type="checkbox"/>
	11:30	<input type="checkbox"/>
ΓΕΩΡΓΙΟΣ ΡΑΠΤΗΣ 2681024598	12:00	<input type="checkbox"/>
	12:30	<input type="checkbox"/>
	13:00	<input checked="" type="checkbox"/>
	13:30	<input type="checkbox"/>
	14:00	<input type="checkbox"/>
	18:00	<input type="checkbox"/>
	18:30	<input checked="" type="checkbox"/>
	19:00	<input type="checkbox"/>
	19:30	<input type="checkbox"/>
	20:00	<input type="checkbox"/>
	20:30	<input checked="" type="checkbox"/>
	21:00	<input type="checkbox"/>

Φόρμα "Ραντεβού"

Τώρα ο χρήστης μπορεί να επιλέξει μια από τις ελεύθερες ώρες, να συμπληρώσει τα στοιχεία του ασθενούς και να πατήσει το κουμπί «Επικύρωση Ενέργειας».

Η ακύρωση ενός ραντεβού γίνεται βγάζοντας το τικ από κάποιο ραντεβού και πατώντας τον κουμπί «Επικύρωση Ενέργειας».

Για εξοικονόμηση χώρου στη βάση δεδομένου καθώς και μετά από συζήτηση μας με ιατρούς θεωρήσαμε σωστό, όλα τα ραντεβού που έχουν παλιότερη από την τρέχουσα ημερομηνία να διαγράφονται αυτόματα μέσω κώδικα:

```
begin
    datamodule2.Appointment.Close;
    datamodule2.Appointment.Open;
    datamodule2.Appointment.First;
    while not datamodule2.Appointment.Eof do
        begin
            if datamodule2.Appointment['A_date']<date then
                begin
                    if datamodule2.Appointment['A_time']='09:00' then
                        datamodule2.Appointment.Delete;
                    if datamodule2.Appointment['A_time']='09:30' then
                        datamodule2.Appointment.Delete;
                        :
                    if datamodule2.Appointment['A_time']='20:30' then
                        datamodule2.Appointment.Delete;
                    if datamodule2.Appointment['A_time']='21:00' then
                        datamodule2.Appointment.Delete;
                end;
            datamodule2.Appointment.Next;
        end;
    Appointments.Show;
end;
```

4.4 Φόρμα «Στατιστικά Στοιχεία Ασθενών»

Η φόρμα αυτή εμφανίζεται με το πάτημα του κουμπιού «Στατιστικά Στοιχεία Ασθενών» και περιέχει γενικές πληροφορίες (Στατιστικά Στοιχεία) για το σύνολο του Ιατρείου.

Στατιστικά Στοιχεία

Στατιστικά Στοιχεία και Διόρθωση Προσωπικών Στοιχείων του Ασθενούς

Οι Ασθενείς που Έχουν Επισκευθεί το Ιατρείο μέχρι Σήμερα, στο Σύνολό τους είναι: **9**

Ανδρες:	3	33 %	Γυναίκες:	6	66 %
Ανδρες Άνεργοι:	0	0 %	Γυναίκες Άνεργες:	1	16 %
Ανδρες Εργαζόμενοι:	3	100 %	Γυναίκες Εργαζόμενες:	5	83 %

Η Ασφάλεια των Ασθενών:

	ΔΗΜΟΣΙΟ	Ι. Κ. Α.	Τ. Ε. Β. Ε.	Ο. Γ. Α.	ΆΛΛΕΣ
Ασφαλισμένοι:	1	7	1	0	0
Ανασφάλιστοι:	11 %	77 %	11 %	0 %	0 %

Πλήθος Ασθενών με κοινή Ασθένεια:

Ανδρες:	Γυναίκες:	Ασθένεια: <input type="text"/>
Ανδρες Άνεργοι:	Γυναίκες Άνεργες:	<input type="button" value="»»"/>
Ανδρες Εργαζόμενοι:	Γυναίκες Εργαζόμενες:	<input type="button" value="»»"/>

Επιλογές:

Έρευνα

Άκυρο

Αρχικό Μενού

Φόρμα "Στατιστικά Στοιχεία Ασθενών"

Η φόρμα αυτή αποτελείται από:

- ✘ Πολλά Labels, τα οποία απεικονίζουν αριθμούς και ποσοστά τα οποία αντιστοιχούν στα στατιστικά αποτελέσματα.
- ✘ Ένα Edit, στο οποίο καταγράφεται η ασθένεια.
- ✘ Τρία κουμπιά, τα οποία είναι η Εύρεση, το Άκυρο και το Αρχικό Μενού.

Πώς και Γιατί;

Η φόρμα αυτή έγινε με σκοπό να παρέχει στον ιατρό γενικές πληροφορίες για το σύνολο των ασθενών. Μπορεί να αναφερθεί σ' αυτή κάθε φορά που θέλει να κάνει μια στατιστική έρευνα που αφορά τους ασθενείς του. Μπορεί δηλαδή, άμεσα, να δει:

- ✿ Πόσοι είναι οι ασθενείς του, πόσοι από αυτούς είναι άνδρες και πόσοι γυναίκες, πόσοι από αυτούς είναι εργαζόμενοι και πόσοι άνεργοι, καθώς και τα σχετικά ποσοστά.
- ✿ Πόσοι από τους ασθενείς του είναι ασφαλισμένοι και πόσοι ανασφάλιστοι, καθώς και πόσοι είναι ασφαλισμένοι σε κάθε ένα από τα γνωστά ασφαλιστικά ιατρικά ταμεία, και τα ποσοστά τους.
- ✿ Πόσοι από τους ασθενείς του έχουν κοινή ασθένεια (την ασθένεια που καταγράφει στο **Edit**), πόσοι από αυτούς είναι άνδρες και πόσοι από αυτούς είναι γυναίκες, πόσοι από αυτούς είναι εργαζόμενοι και πόσοι όχι, καθώς και τα σχετικά ποσοστά.

Η πρώτη(1) κουκίδα υπολογίζεται μέσω του κώδικα:

```
while not datamodule2.Patient.Eof do
  begin
    if datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ' then
      begin
        m:=m+1;
      end
    else
      begin
        w:=w+1;
      end;
      if (datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ') and
        (datamodule2.Patient['job'] = '') then
        begin
          mu:=mu+1;
        end
      else
        if (datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ') and
          (datamodule2.Patient['job'] <> '') then
          begin
            me:=me+1;
          end;
        if (datamodule2.Patient['sex'] = 'ΓΥΝΑΙΚΑ') and
          (datamodule2.Patient['job'] = '') then
          begin
            wu:=wu+1;
          end
        else
          if (datamodule2.Patient['sex'] = ' ΓΥΝΑΙΚΑ') and
            (datamodule2.Patient['job'] <> '') then
            begin
              we:=we+1;
            end;
          datamodule2.Patient.Next;
        end;
      end;
    end;
```


Η δεύτερη (2) υπολογίζεται μέσω του κώδικα:

```
while not datamodule2.Patient.Eof do
  begin
    if datamodule2.Patient['Ensurance']='' then
      unens:=unens+1
    else
      if datamodule2.Patient['Ensurance']='I. K. A.' then
        ika:=ika+1
      else
        if datamodule2.Patient['Ensurance']='Δημόσιο' then
          dim:=dim+1
        else
          if datamodule2.Patient['Ensurance']='T.E.B.E.' then
            teve:=teve+1
          else
            if datamodule2.Patient['Ensurance']='O.Γ.A.' then
              oga:=oga+1
            else
              begin
                alli:=alli+1;
              end;
            datamodule2.Patient.Next;
          end;
        end;
      end;
    end;
  end;
```

Η τρίτη (3) υπολογίζεται μέσω του κώδικα:

```
while not datamodule2.Patient.Eof do
  begin
    datamodule2.Visits.First;
    while not datamodule2.Visits.Eof do
      begin
        if (datamodule2.Visits['Diagnose'] = edit1.Text) Then
          if (datamodule2.Visits['patient_kode'] =
            (datamodule2.Patient['patient_kode'] ) then
            begin
              if (datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ') and
                (datamodule2.Patient['job'] = ' ') then
                begin
                  muj:=muj+1;
                end;
              if (datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ') and
                (datamodule2.Patient['job'] <> ' ') then
                begin
                  mj:=mj+1;
                end;
              if (datamodule2.Patient['sex'] = 'ΓΥΝΑΙΚΑ') and
                (datamodule2.Patient['job'] = ' ') then
                begin
                  wuj:=wuj+1;
                end;
              if datamodule2.Patient['sex'] = 'ΑΝΔΡΑΣ' then
                begin
                  md:=md+1;
                end
            else
              begin
                wd:=wd+1;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
```

```

if (datamodule2.Patient['sex'] = 'ΓΥΝΑΙΚΑ') and
  (datamodule2.Patient['job'] <> ' ') then
begin
  wj:=wj+1;
end;
datamodule2.Visits.Last;
end;
datamodule2.Visits.Next;
end;
datamodule2.Patient.Next;

```

Η φόρμα «Στατιστικά Στοιχεία Ασθενών» με το πλήθος και τα ποσοστά των ασθενών που έχουν Γρίπη:

Στατιστικά Στοιχεία

Στατιστικά Στοιχεία και Διόρθωση Προσωπικών Στοιχείων του Ασθενούς

Οι Ασθενείς που Έχουν Επισκευθεί το Ιατρείο μέχρι Σήμερα, στο Σύνολό τους είναι: **9**

Ανδρες:	3	33 %	Γυναίκες:	6	66 %
Ανδρες Άνεργοι:	0	0 %	Γυναίκες Άνεργες:	1	16 %
Ανδρες Εργαζόμενοι:	3	100 %	Γυναίκες Εργαζόμενες:	5	83 %

Η Ασφάλεια των Ασθενών:

	ΔΗΜΟΣΙΟ	Ι. Κ. Α.	Τ. Ε. Β. Ε.	Ο. Γ. Α.	ΑΛΛΕΣ
Ασφαλισμένοι:	1	7	1	0	0
Ανασφάλιστοι:	0	0	0	0	0

Πλήθος Ασθενών με κοινή Ασθένεια: **3**

Ανδρες:	1	33 %	Γυναίκες:	2	66 %
Ανδρες Άνεργοι:	0	0 %	Γυναίκες Άνεργες:	0	0 %
Ανδρες Εργαζόμενοι:	1	100 %	Γυναίκες Εργαζόμενες:	2	100 %

Ασθένεια:

Επιλογές:

- Έρευνα
- Άκυρο
- Αρχικό Μενού

Φόρμα "Στατιστικά Στοιχεία"

Τα κουμπιά, όπως και στις άλλες φόρμες, εκτελούν τις αντίστοιχες με το όνομά τους λειτουργίες, και ανοίγουν τις αντίστοιχες φόρμες.

Η έρευνα γίνεται με παραμετρικό ερώτημα.

4.5 Φόρμα «Συνεργάτες»

Η φόρμα αυτή εμφανίζεται με το πάτημα του κουμπιού «Συνεργάτες» κάθε φορά που ο χρήστης θέλει να εισάγει ή να πάρει πληροφορίες για κάποιον ή κάποιους συνεργάτες του ιατρείου οι οποίοι μπορεί να είναι άλλοι ιατροί, ιατρικά εργαστήρια (μικροβιολογικά), προμηθευτές αναλώσιμων και πάγιων υλικών και άλλοι.

The screenshot shows a web form titled "Καταγραφή των Προσωπικών Στοιχείων Συνεργάτη, Διόρθωση, Διαγραφή". The form is divided into two main sections:

- Προσωπικά Στοιχεία Συνεργάτη:** A vertical list of input fields for: Κωδικός, Όνομα, Επώνυμο, Ειδικότητα, Διεύθυνση, Τηλέφωνο, E_Mail, Web Site, and Ιδιαιτερότητα.
- Διαθέσιμες Επιλογές:** A vertical list of buttons with arrows:
 - <<>> Καταχώρηση
 - >> Διόρθωση
 - >> Διαγραφή
 - >> Έρευνα
 - >> Άκυρο
 - << Αρχικό Μενού

Φόρμα "Συνεργάτες"

Η φόρμα αυτή αποτελείται από:

- ✘ Τρία Edits, στα οποία ο χρήστης της εφαρμογής καταγράφει το Όνομα, το Επώνυμο και τον Αριθμό Φορολογικού Μητρώου του (Α. Φ. Μ.).

- ✘ Έξι DBEdits τα οποία παίρνουν τις τιμές: την Ειδικότητα του συνεργάτη, τη Διεύθυνση, το Τηλέφωνο, το E-Mail του, το Web Site και κάποια Ιδιαιτερότητά αν υπάρχει.
- ✘ Έξι κουμπιά τα οποία κατά την επιλογή τους οδηγούν τον χρήστη σε αντίστοιχες λειτουργίες: Καταχώρηση, Διαγραφή, Άκυρο, Εύρεση Συνεργάτη και Αρχικό Μενού.

Πώς και Γιατί;

Κατά την διάρκεια της λειτουργίας της καταχώρησης, αφού συμπληρώσουμε το πρώτο Edit και μπούμε (OnEnter) στο δεύτερο (Όνομα) πραγματοποιείται ένας έλεγχος παρόμοιος μ' αυτόν της φόρμας «Ασθενείς» και γίνεται για τον ίδιο λόγο, δηλαδή να αποφύγουμε εγγραφή του ίδιου συνεργάτη πάνω από μια φορά. Συγκρίνεται δηλαδή η τιμή του πρώτου Edit (Α. Φ. Μ.) με το πεδίο C_Kode του πίνακα `sinergates`.

```
begin  
  datamodule2.Sinergates.Close;  
  datamodule2.Sinergates.Open;  
  datamodule2.Sinergates.First;  
  d:=false;  
  while not datamodule2.Sinergates.Eof do  
    begin  
      if datamodule2.Sinergates['C_Kode'] = edit1.Text then  
        begin  
          c:=datamodule2.Sinergates.GetBookmark;  
          d:=true;  
        end;  
      datamodule2.Sinergates.Next;  
    end;
```

```
if (d=true) then
  begin
    Button1.Enabled:=false;
    label14.Enabled:=false;
    Button2.Enabled:=true;
    label10.Enabled:=true;
    Button3.Enabled:=true;
    label11.Enabled:=true;
    datamodule2.Sinergates.GotoBookmark(c);
    edit1.Text:=datamodule2.Sinergates['C_Kode'];
    edit2.Text:=datamodule2.Sinergates['name'];
    edit3.Text:=datamodule2.Sinergates['L_Name'];
  end
else
  Begin
    datamodule2.Sinergates.Last;
    datamodule2.Sinergates.Insert;
    datamodule2.Sinergates['C_Kode']:=edit1.Text;
    datamodule2.Sinergates['Name']:=edit2.Text;
    datamodule2.Sinergates['L_Name']:=edit3.Text;
    button1.Enabled:=true;
    label14.Enabled:=true;
  end;
end;
```

Αν δεν βρεθεί ομοιότητα ενεργοποιείται το κουμπί «Καταχώρηση» και συνεχίζουμε στην συμπλήρωση των στοιχείων του συνεργάτη.

Αν βρεθεί ομοιότητα σημαίνει ότι ο συνεργάτης έχει ήδη καταχωρηθεί και τα στοιχεία του εμφανίζονται στην DBEdits της φόρμας, ενεργοποιούνται τα κουμπιά «Διόρθωση» και «Διαγραφή» ενώ παραμένει ανενεργό το κουμπί «Καταχώρηση».

Τη στιγμή που πατάμε το κουμπί «Διόρθωση» εκτελείται ο εξής κώδικας:

```
begin
  datamodule2.Sinergates.Open;
  datamodule2.Sinergates.First;
  while not datamodule2.Sinergates.Eof do
    begin
      if datamodule2.Sinergates['C_Kode']= edit1.Text then
        begin
          datamodule2.Sinergates.Edit;
          datamodule2.Sinergates['Name']:=edit2.Text;
          datamodule2.Sinergates['L_Name']:=edit3.Text;
          datamodule2.Sinergates.post;
          exit;
        end;
      datamodule2.Sinergates.Next;
    end;
  end;
```

Τη στιγμή που πατάμε το κουμπί «Διαγραφή» εκτελείται ο εξής κώδικας:

```
begin
  if application.MessageBox('Διαγραφή Τρέχουσας Εγγραφής',
    'Εκτελείται Διαγραφή',
    mb_okcancel + mb_iconquestion)=
    idok then
    begin
      datamodule2.Sinergates.Delete;
      datamodule2.Sinergates.Insert;
      showmessage('1 Εγγραφή Διεγράφη.');
```

4.5.1 Φόρμα «Εύρεση Συνεργατών»

Στη φόρμα αυτή υπάρχει η δυνατότητα έρευνας συνεργατών με βάση το Επώνυμο. Αφού κάνουμε την έρευνα και βρούμε τουλάχιστο έναν, κάνοντας κλικ σε κάποιο πεδίο του DBGrid που υπάρχει εκεί ενεργοποιούνται τα κουμπιά «Διόρθωσης» και «Διαγραφής».

C_code	I_Name	Name	Sector	phone	address	E_Mail	Site	S_T
213548698	ΦΛΩΡΟΥ-ΑΒΑΝΑΣ	ΒΑΣΙΛΙΚΗ	ΟΔΟΝΤΙΑΤΡΟΣ	2641025698	ΠΑΠΑΣΤΡΑΤΟΥ 12			ΙΑΤΡΟΣ

Φόρμα "Εύρεση Συνεργατών"

Εδώ η διόρθωση δεν γίνεται με κώδικα Delphi αλλά με κώδικα SQL. Επίσης χρησιμοποιήσαμε τα στοιχεία UpdateSQL, DataSource και Query. Ο ακόλουθος κώδικας γράφεται στην ιδιότητα (Properties) ModifySQL του UpdateSQL:


```
UPDATE sinergates set Name =:Name,  
L_Name=:L_Name,  
phone =:phone,address=:address,  
E_Mail =:E_Mail, Sector=:sector,  
Site=:Site where C_Kode=:C_Kode
```

Η εκτέλεση του παραπάνω κώδικα γίνεται με την παρακάτω πρόταση η οποία γράφεται στον κώδικα της Delphi:

```
Query1.Database.ApplyUpdates([Query1]);
```

Η Διαγραφή γίνεται με κώδικα Delphi:

```
begin  
    datamodule2.Sinergates.Close;  
    datamodule2.Sinergates.Open;  
    datamodule2.Sinergates.first;  
    while not datamodule2.Sinergates.Eof do  
        begin  
            if datamodule2.Sinergates['C_Kode']=d then  
                if application.MessageBox('Διαγραφή Τρέχουσας  
                    Εγγραφής','Εκτελείται Διαγραφή',  
                    mb_okcancel + mb_iconquestion)=  
                        idok then  
                    begin  
                        datamodule2.Sinergates.Delete;  
                    end;  
                    datamodule2.Sinergates.Next;  
                end;  
            end;  
        end;  
    * d:=query1.FieldValues['C_Kode'];
```

Επίλογος

Μετά την παρουσίαση και ανάλυση της εφαρμογής μας είναι πλέον εμφανές πώς τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής αυτής, *Delphi* και *SQL Server*, μπορούν να χρησιμοποιηθούν από προγραμματιστές για τη δημιουργία εύχρηστων εφαρμογών, πραγματικά αποτελεσματικών, ακόμη και για τον πιο απαιτητικό χρήστη.

Η *Delphi* επιτρέπει στο περιβάλλον της εφαρμογής να είναι πραγματικά φιλικό προς το χρήστη της. Επιπλέον, παρέχει στον προγραμματιστή την δυνατότητα να δώσει την εμφάνιση που εκείνος επιθυμεί. Με λίγα λόγια του επιτρέπει να δώσει στην εφαρμογή προσωπικότητα, σχεδιάζοντας την εμφάνισή της σύμφωνα με το προσωπικό του γούστο.

Ένα επιπλέον πλεονέκτημα που προσφέρει η χρήση του συνδυασμού *Delphi* - *SQL Server* είναι η χρηστικότητα της εφαρμογής (η οποία έχει αναπτυχθεί με τα παραπάνω εργαλεία).

Αξιόλογη επίσης είναι και η επεκτασιμότητα των εν λόγω εφαρμογών. Διατίθεται, δηλαδή, η δυνατότητα ενσωμάτωσης άλλων εφαρμογών πάνω στις υπάρχουσες, εφόσον αυτό είναι επιθυμητό.

Παράρτημα

Ο Κώδικας της Εφαρμογής (Delphi)

unit UnWelcome;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;*

type

TWelcome = class(TForm)

Button1: TButton;

Button2: TButton;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Button3: TButton;

Button4: TButton;

Label9: TLabel;

Label10: TLabel;

```
Button5: TButton;
Label11: TLabel;
Label12: TLabel;
Label85: TLabel;
Label25: TLabel;
Label14: TLabel;
Label116: TLabel;
Label80: TLabel;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Welcome: TWelcome;

implementation

uses UnDataModule1, UnInput, UnAppointment, UnVisits, UnBackup,
  UnStatistics, UnDetail, UnSinergates;

{$R *.dfm}

procedure TWelcome.Button1Click(Sender: TObject);
begin
  Input.Show;
end;
```

```
procedure TWelcome.Button2Click(Sender: TObject);
begin
    datamodule2.Appointment.Close;
    datamodule2.Appointment.Open;
    datamodule2.Appointment.First;
while not datamodule2.Appointment.Eof do
    begin
        if datamodule2.Appointment['A_date']<date then
            begin
                if datamodule2.Appointment['A_time']='09:00' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
                if datamodule2.Appointment['A_time']='09:30' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
                if datamodule2.Appointment['A_time']='10:00' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
                if datamodule2.Appointment['A_time']='10:30' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
                if datamodule2.Appointment['A_time']='11:00' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
                if datamodule2.Appointment['A_time']='11:30' then
                    begin
                        datamodule2.Appointment.Delete;
                    end;
            end;
    end;
```

```
if datamodule2.Appointment['A_time']='12:00' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='12:30' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='13:00' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='13:30' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='14:00' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='18:00' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='18:30' then
  begin
    datamodule2.Appointment.Delete;
  end;
if datamodule2.Appointment['A_time']='19:00' then
  begin
    datamodule2.Appointment.Delete;
  end;
```

```
    if datamodule2.Appointment['A_time']='19:30' then
        begin
            datamodule2.Appointment.Delete;
        end;
    if datamodule2.Appointment['A_time']='20:00' then
        begin
            datamodule2.Appointment.Delete;
        end;
    if datamodule2.Appointment['A_time']='20:30' then
        begin
            datamodule2.Appointment.Delete;
        end;
    if datamodule2.Appointment['A_time']='21:00' then
        begin
            datamodule2.Appointment.Delete;
        end;
    end;
    datamodule2.Appointment.Next;
end;
Appointments.Show;
end;

procedure TWelcome.Button3Click(Sender: TObject);
begin
    form2.Query1.Close;
    sinergates.Show;
end;

procedure TWelcome.Button4Click(Sender: TObject);
begin
    statistiks.Show;
end;
```

```
procedure TWelcome.Button5Click(Sender: TObject);  
begin  
    welcome.Close;  
end;  
  
end.
```


unit UnAppointment;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask, DBCtrls, ComCtrls;*

type

TAppointments = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label5: TLabel;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

CheckBox4: TCheckBox;

CheckBox5: TCheckBox;

CheckBox6: TCheckBox;

CheckBox7: TCheckBox;

CheckBox8: TCheckBox;

CheckBox9: TCheckBox;

CheckBox10: TCheckBox;

CheckBox11: TCheckBox;

CheckBox12: TCheckBox;

CheckBox13: TCheckBox;

CheckBox14: TCheckBox;

CheckBox15: TCheckBox;

CheckBox16: TCheckBox;

CheckBox17: TCheckBox;

CheckBox18: TCheckBox;

MonthCalendar1: TMonthCalendar;

Button1: TButton;

Label3: TLabel;

Label4: TLabel;

Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label33: TLabel;
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
Label37: TLabel;
Label38: TLabel;
Label39: TLabel;

Label40: TLabel;

Label41: TLabel;

Label42: TLabel;

Label43: TLabel;

Label44: TLabel;

Label45: TLabel;

Label46: TLabel;

Label47: TLabel;

Label48: TLabel;

Label49: TLabel;

Label50: TLabel;

Label51: TLabel;

Label52: TLabel;

Label53: TLabel;

Label54: TLabel;

Label55: TLabel;

Label56: TLabel;

Label57: TLabel;

Label58: TLabel;

Label59: TLabel;

Label60: TLabel;

Label61: TLabel;

Label62: TLabel;

Label63: TLabel;

Label64: TLabel;

Label65: TLabel;

Label66: TLabel;

Label67: TLabel;

Label68: TLabel;

Label69: TLabel;

Label70: TLabel;

Label71: TLabel;

Label72: TLabel;

Label73: TLabel;

```
Label74: TLabel;
Label75: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Label76: TLabel;
Label77: TLabel;
Label79: TLabel;
Button2: TButton;
Label81: TLabel;
Label85: TLabel;
Label80: TLabel;
Label84: TLabel;
Label82: TLabel;
Label86: TLabel;
Label104: TLabel;
Label105: TLabel;
Label107: TLabel;
Label108: TLabel;
Label109: TLabel;
Label128: TLabel;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure MonthCalendar1DbClick(Sender: TObject);
procedure MonthCalendar1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Appointments: TAppointments;
```

implementation

uses UnDataModule1, UnInput;

*{ \$R *.dfm }*

procedure TAppointments.Button1Click(Sender: TObject);

begin

datamodule2.Appointment.Insert;

if (edit1.Text <> '') and (edit2.Text <> '') then

if monthcalendar1.Date >= date then

begin

if (checkbox1.Checked = true) and (label3.Caption=' ') then

begin

datamodule2.Appointment['A_time']:= '09:00';

datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);

datamodule2.Appointment['name']:=edit1.Text;

datamodule2.Appointment['L_name']:=edit2.Text;

datamodule2.Appointment['phone']:=edit3.Text;

edit1.Text:= '';

edit2.Text:= '';

edit3.Text:= '';

datamodule2.Appointment.Post ;

end

else

if (checkbox2.Checked = true) and (label8.Caption=' ') then

begin

datamodule2.Appointment['A_time']:= '09:30';

datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);

datamodule2.Appointment['name']:=edit1.Text;

datamodule2.Appointment['L_name']:=edit2.Text;

datamodule2.Appointment['phone']:=edit3.Text;

edit1.Text:= '';

edit2.Text:= '';

edit3.Text:= '';

```
    datamodule2.Appointment.Post;
end
else
if (checkbox3.Checked = true) and (label12.Caption=' ') then
begin
    datamodule2.Appointment['A_time']:= '10:00';
    datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
    datamodule2.Appointment['name']:=edit1.Text;
    datamodule2.Appointment['L_name']:=edit2.Text;
    datamodule2.Appointment['phone']:=edit3.Text;
    edit1.Text:="";
    edit2.Text:="";
    edit3.Text:="";
    datamodule2.Appointment.Post;
end
else
if (checkbox4.Checked = true) and (label16.Caption=' ') then
begin
    datamodule2.Appointment['A_time']:= '10:30';
    datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
    datamodule2.Appointment['name']:=edit1.Text;
    datamodule2.Appointment['L_name']:=edit2.Text;
    datamodule2.Appointment['phone']:=edit3.Text;
    edit1.Text:="";
    edit2.Text:="";
    edit3.Text:="";
    datamodule2.Appointment.Post ;
end
else
if (checkbox5.Checked = true) and (label20.Caption=' ') then
begin
    datamodule2.Appointment['A_time']:= '11:00';
    datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
    datamodule2.Appointment['name']:=edit1.Text;
```

```
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox6.Checked = true) and (label24.Caption=' ') then
begin
datamodule2.Appointment['A_time']:= '11:30';
datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:=edit1.Text;
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox7.Checked = true) and (label28.Caption=' ') then
begin
datamodule2.Appointment['A_time']:= '12:00';
datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:=edit1.Text;
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
```

```
else
  if (checkbox8.Checked = true) and (label32.Caption=' ') then
    begin
      datamodule2.Appointment['A_time']:= '12:30';
      datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
      datamodule2.Appointment['name']:=edit1.Text;
      datamodule2.Appointment['L_name']:=edit2.Text;
      datamodule2.Appointment['phone']:=edit3.Text;
      edit1.Text:="";
      edit2.Text:="";
      edit3.Text:="";
      datamodule2.Appointment.Post ;
    end
  else
    if (checkbox9.Checked = true) and (label36.Caption=' ') then
      begin
        datamodule2.Appointment['A_time']:= '13:00';
        datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
        datamodule2.Appointment['name']:=edit1.Text;
        datamodule2.Appointment['L_name']:=edit2.Text;
        datamodule2.Appointment['phone']:=edit3.Text;
        edit1.Text:="";
        edit2.Text:="";
        edit3.Text:="";
        datamodule2.Appointment.Post ;
      end
    else
      if (checkbox10.Checked = true) and (label40.Caption=' ') then
        begin
          datamodule2.Appointment['A_time']:= '13:30';
          datamodule2.Appointment['A_date']:=DateToStr(monthcalendar1.Date);
          datamodule2.Appointment['name']:=edit1.Text;
          datamodule2.Appointment['L_name']:=edit2.Text;
          datamodule2.Appointment['phone']:=edit3.Text;
```



```
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox11.Checked = true) and (label44.Caption=' ') then
begin
datamodule2.Appointment['A_time']:='14:00';
datamodule2.Appointment['A_date']:=
DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:='edit1.Text';
datamodule2.Appointment['L_name']:='edit2.Text';
datamodule2.Appointment['phone']:='edit3.Text';
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox12.Checked = true) and (label48.Caption=' ') then
begin
datamodule2.Appointment['A_time']:='18:00';
datamodule2.Appointment['A_date']:=
DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:='edit1.Text';
datamodule2.Appointment['L_name']:='edit2.Text';
datamodule2.Appointment['phone']:='edit3.Text';
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
```

```
else
  if (checkbox13.Checked = true) and (label52.Caption=' ') then
    begin
      datamodule2.Appointment['A_time']:='18:30';
      datamodule2.Appointment['A_date']:=
        DateToStr(monthcalendar1.Date);
      datamodule2.Appointment['name']:='edit1.Text';
      datamodule2.Appointment['L_name']:='edit2.Text';
      datamodule2.Appointment['phone']:='edit3.Text';
      edit1.Text:='';
      edit2.Text:='';
      edit3.Text:='';
      datamodule2.Appointment.Post ;
    end
  else
    if (checkbox14.Checked = true) and (label56.Caption=' ') then
      begin
        datamodule2.Appointment['A_time']:='19:00';
        datamodule2.Appointment['A_date']:=
          DateToStr(monthcalendar1.Date);
        datamodule2.Appointment['name']:='edit1.Text';
        datamodule2.Appointment['L_name']:='edit2.Text';
        datamodule2.Appointment['phone']:='edit3.Text';
        edit1.Text:='';
        edit2.Text:='';
        edit3.Text:='';
        datamodule2.Appointment.Post ;
      end
    else
      if (checkbox15.Checked = true) and (label60.Caption=' ') then
        begin
          datamodule2.Appointment['A_time']:='19:30';
          datamodule2.Appointment['A_date']:=
            DateToStr(monthcalendar1.Date);
```

```
datamodule2.Appointment['name']:=edit1.Text;
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox16.Checked = true) and (label64.Caption=' ') then
begin
datamodule2.Appointment['A_time']:= '20:00';
datamodule2.Appointment['A_date']:=
DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:=edit1.Text;
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
datamodule2.Appointment.Post ;
end
else
if (checkbox17.Checked = true) and (label68.Caption=' ') then
begin
datamodule2.Appointment['A_time']:= '20:30';
datamodule2.Appointment['A_date']:=
DateToStr(monthcalendar1.Date);
datamodule2.Appointment['name']:=edit1.Text;
datamodule2.Appointment['L_name']:=edit2.Text;
datamodule2.Appointment['phone']:=edit3.Text;
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
```

```
        datamodule2.Appointment.Post ;
    end
else
    if (checkbox18.Checked = true) and (label72.Caption=' ') then
        begin
            datamodule2.Appointment['A_time']:= '21:00';
            datamodule2.Appointment['A_date']:=
                DateToStr(monthcalendar1.Date);
            datamodule2.Appointment['name']:=edit1.Text;
            datamodule2.Appointment['L_name']:=edit2.Text;
            datamodule2.Appointment['phone']:=edit3.Text;
            edit1.Text:='';
            edit2.Text:='';
            edit3.Text:='';
            datamodule2.Appointment.Post ;
        end
    else
        begin
            showmessage('Ωπα:');
            button1.Enabled:=true;
            label76.Enabled:=true;
        end;
    datamodule2.Appointment.Close;
    datamodule2.Appointment.Open;
    datamodule2.Appointment.First;
    while not datamodule2.Appointment.Eof do
        begin
            if datamodule2.Appointment['A_date']=monthcalendar1.Date then
                if datamodule2.Appointment['A_time']='09:00' then
                    begin
                        label3.Caption:=datamodule2.Appointment['Name'];
                        label4.Caption:=datamodule2.Appointment['L_Name'];
                        label6.Caption:=datamodule2.Appointment['Phone'];
                        checkbox1.Checked:=true;
```

```
end
else
  if datamodule2.Appointment['A_time']='09:30' then
    begin
      label8.Caption:=datamodule2.Appointment['Name'];
      label9.Caption:=datamodule2.Appointment['L_Name'];
      label10.Caption:=datamodule2.Appointment['Phone'];
      checkbox2.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='10:00' then
      begin
        label12.Caption:=datamodule2.Appointment['Name'];
        label13.Caption:=datamodule2.Appointment['L_Name'];
        label14.Caption:=datamodule2.Appointment['Phone'];
        checkbox3.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='10:30' then
        begin
          label16.Caption:=datamodule2.Appointment['Name'];
          label17.Caption:=datamodule2.Appointment['L_Name'];
          label18.Caption:=datamodule2.Appointment['Phone'];
          checkbox4.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='11:00' then
          begin
            label20.Caption:=datamodule2.Appointment['Name'];
            label21.Caption:=datamodule2.Appointment['L_Name'];
            label22.Caption:=datamodule2.Appointment['Phone'];
            checkbox5.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='11:30' then
    begin
      label24.Caption:=datamodule2.Appointment['Name'];
      label25.Caption:=datamodule2.Appointment['L_Name'];
      label26.Caption:=datamodule2.Appointment['Phone'];
      checkbox6.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='12:00' then
      begin
        label28.Caption:=datamodule2.Appointment['Name'];
        label29.Caption:=datamodule2.Appointment['L_Name'];
        label30.Caption:=datamodule2.Appointment['Phone'];
        checkbox7.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='12:30' then
        begin
          label32.Caption:=datamodule2.Appointment['Name'];
          label33.Caption:=datamodule2.Appointment['L_Name'];
          label34.Caption:=datamodule2.Appointment['Phone'];
          checkbox8.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='13:00' then
          begin
            label36.Caption:=datamodule2.Appointment['Name'];
            label37.Caption:=datamodule2.Appointment['L_Name'];
            label38.Caption:=datamodule2.Appointment['Phone'];
            checkbox9.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='13:30' then
    begin
      label40.Caption:=datamodule2.Appointment['Name'];
      label41.Caption:=datamodule2.Appointment['L_Name'];
      label42.Caption:=datamodule2.Appointment['Phone'];
      checkbox10.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='14:00' then
      begin
        label44.Caption:=datamodule2.Appointment['Name'];
        label45.Caption:=datamodule2.Appointment['L_Name'];
        label46.Caption:=datamodule2.Appointment['Phone'];
        checkbox11.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='18:00' then
        begin
          label48.Caption:=datamodule2.Appointment['Name'];
          label49.Caption:=datamodule2.Appointment['L_Name'];
          label50.Caption:=datamodule2.Appointment['Phone'];
          checkbox12.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='18:30' then
          begin
            label52.Caption:=datamodule2.Appointment['Name'];
            label53.Caption:=datamodule2.Appointment['L_Name'];
            label54.Caption:=datamodule2.Appointment['Phone'];
            checkbox13.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='19:00' then
    begin
      label56.Caption:=datamodule2.Appointment['Name'];
      label57.Caption:=datamodule2.Appointment['L_Name'];
      label58.Caption:=datamodule2.Appointment['Phone'];
      checkbox14.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='19:30' then
      begin
        label60.Caption:=datamodule2.Appointment['Name'];
        label61.Caption:=datamodule2.Appointment['L_Name'];
        label62.Caption:=datamodule2.Appointment['Phone'];
        checkbox15.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='20:00' then
        begin
          label64.Caption:=datamodule2.Appointment['Name'];
          label65.Caption:=datamodule2.Appointment['L_Name'];
          label66.Caption:=datamodule2.Appointment['Phone'];
          checkbox16.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='20:30' then
          begin
            label68.Caption:=datamodule2.Appointment['Name'];
            label69.Caption:=datamodule2.Appointment['L_Name'];
            label70.Caption:=datamodule2.Appointment['Phone'];
            checkbox17.Checked:=true;
          end
        end
      end
    end
  end
end
```



```
else
  if datamodule2.Appointment['A_time']='21:00' then
    begin
      label72.Caption:=datamodule2.Appointment['Name'];
      label73.Caption:=datamodule2.Appointment['L_Name'];
      label74.Caption:=datamodule2.Appointment['Phone'];
      checkbox18.Checked:=true;
    end ;
    datamodule2.Appointment.Next;
  end;
end
else
  showmessage('Έάεèò ÇìãñïçíΒά');
datamodule2.Appointment.open;
if (checkbox1.Checked = false) and (label3.Caption<>'') then
  begin
    datamodule2.Appointment.open;
    datamodule2.Appointment.First;
    while not datamodule2.Appointment.Eof do
      begin
        if (datamodule2.Appointment['A_time']='09:00') and
          (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
          begin
            datamodule2.Appointment.Delete;
            label3.Caption:=' ';
            label7.Caption:=' ';
            label6.Caption:=' ';
            label4.Caption:=' ';
          end;
          datamodule2.Appointment.Next;
        end;
      end;
    end;
  end;
  if (checkbox2.Checked = false) and (label8.Caption<>' ') then
    begin
```

```
datamodule2.Appointment.open;
datamodule2.Appointment.First;
while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='09:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
      begin
        datamodule2.Appointment.Delete;
        label8.Caption:=' ';
        label9.Caption:=' ';
        label10.Caption:=' ';
        label11.Caption:=' ';
      end;
      datamodule2.Appointment.Next;
    end;
  end;
if (checkbox3.Checked = false) and (label12.Caption<>' ') then
  begin
    datamodule2.Appointment.First;
    while not datamodule2.Appointment.Eof do
      begin
        if (datamodule2.Appointment['A_time']='10:00') and
          (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
            begin
              datamodule2.Appointment.Delete;
              label12.Caption:=' ';
              label13.Caption:=' ';
              label14.Caption:=' ';
              label15.Caption:=' ';
            end;
            datamodule2.Appointment.Next;
          end;
        end;
      end;
    end;
```

```
if (checkbox4.Checked = false) and (label16.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='10:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label16.Caption:=' ';
      label17.Caption:=' ';
      label18.Caption:=' ';
      label19.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox5.Checked = false) and (label20.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='11:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label20.Caption:=' ';
      label21.Caption:=' ';
      label22.Caption:=' ';
      label23.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox6.Checked = false) and (label24.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='11:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label24.Caption:=' ';
      label25.Caption:=' ';
      label26.Caption:=' ';
      label27.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox7.Checked = false) and (label28.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='12:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label28.Caption:=' ';
      label29.Caption:=' ';
      label30.Caption:=' ';
      label31.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox8.Checked = false) and (label32.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='12:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label32.Caption:=' ';
      label33.Caption:=' ';
      label34.Caption:=' ';
      label35.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox9.Checked = false) and (label36.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='13:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label36.Caption:=' ';
      label37.Caption:=' ';
      label38.Caption:=' ';
      label39.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox10.Checked = false) and (label40.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='13:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label40.Caption:=' ';
      label41.Caption:=' ';
      label42.Caption:=' ';
      label43.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox11.Checked = false) and (label44.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='14:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label44.Caption:=' ';
      label45.Caption:=' ';
      label46.Caption:=' ';
      label47.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox12.Checked = false) and (label48.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='18:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label48.Caption:=' ';
      label49.Caption:=' ';
      label50.Caption:=' ';
      label51.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox13.Checked = false) and (label52.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='18:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label52.Caption:=' ';
      label53.Caption:=' ';
      label54.Caption:=' ';
      label55.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox14.Checked = false) and (label56.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='19:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label56.Caption:=' ';
      label57.Caption:=' ';
      label58.Caption:=' ';
      label59.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox15.Checked = false) and (label60.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='19:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label60.caption:=' ';
      label61.caption:=' ';
      label62.caption:=' ';
      label63.caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```



```
if (checkbox16.Checked = false) and (label64.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='20:00') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label64.Caption:=' ';
      label65.Caption:=' ';
      label66.Caption:=' ';
      label67.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;

if (checkbox17.Checked = false) and (label68.Caption<>' ') then
begin
  datamodule2.Appointment.First;
  while not datamodule2.Appointment.Eof do
  begin
    if (datamodule2.Appointment['A_time']='20:30') and
      (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
    begin
      datamodule2.Appointment.Delete;
      label68.Caption:=' ';
      label69.Caption:=' ';
      label70.Caption:=' ';
      label71.Caption:=' ';
    end;
    datamodule2.Appointment.Next;
  end;
end;
```

```
if (checkbox18.Checked = false) and (label72.Caption<>' ') then
begin
    datamodule2.Appointment.First;
    while not datamodule2.Appointment.Eof do
        begin
            if (datamodule2.Appointment['A_time']='21:00') and
                (datamodule2.Appointment['A_date']=monthcalendar1.Date) then
                begin
                    datamodule2.Appointment.Delete;
                    label72.Caption:=' ';
                    label73.Caption:=' ';
                    label74.Caption:=' ';
                    label75.Caption:=' ';
                end;
            datamodule2.Appointment.Next;
        end;
    end;
end;

procedure TAppointments.FormCreate(Sender: TObject);
begin
    monthcalendar1.Date:=date;
    button1.Enabled:=true;
    label76.Enabled:=true;
    datamodule2.Appointment.Close;
    datamodule2.Appointment.Open;
    datamodule2.Appointment.Insert;
    edit1.Text:="";
    edit2.Text:="";
    edit3.Text:="";
end;
```

```
procedure TAppointments.MonthCalendar1DbClick(Sender: TObject);
begin
  datamodule2.Appointment.Open;
  datamodule2.Appointment.First;
  While not datamodule2.Appointment.Eof do
  begin
    if datamodule2.Appointment['A_date']=monthcalendar1.Date then
    begin
      if datamodule2.Appointment['A_time']='09:00' then
      begin
        label3.Caption:=datamodule2.Appointment['Name'];
        label4.Caption:=datamodule2.Appointment['L_Name'];
        label6.Caption:=datamodule2.Appointment['Phone'];
        checkbox1.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='09:30' then
      begin
        label8.Caption:=datamodule2.Appointment['Name'];
        label9.Caption:=datamodule2.Appointment['L_Name'];
        label10.Caption:=datamodule2.Appointment['Phone'];
        checkbox2.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='10:00' then
      begin
        label12.Caption:=datamodule2.Appointment['Name'];
        label13.Caption:=datamodule2.Appointment['L_Name'];
        label14.Caption:=datamodule2.Appointment['Phone'];
        checkbox3.Checked:=true;
      end
    end
  end
```

```
else
  if datamodule2.Appointment['A_time']='10:30' then
    begin
      label16.Caption:=datamodule2.Appointment['Name'];
      label17.Caption:=datamodule2.Appointment['L_Name'];
      label18.Caption:=datamodule2.Appointment['Phone'];
      checkbox4.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='11:00' then
      begin
        label20.Caption:=datamodule2.Appointment['Name'];
        label21.Caption:=datamodule2.Appointment['L_Name'];
        label22.Caption:=datamodule2.Appointment['Phone'];
        checkbox5.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='11:30' then
        begin
          label24.Caption:=datamodule2.Appointment['Name'];
          label25.Caption:=datamodule2.Appointment['L_Name'];
          label26.Caption:=datamodule2.Appointment['Phone'];
          checkbox6.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='12:00' then
          begin
            label28.Caption:=datamodule2.Appointment['Name'];
            label29.Caption:=datamodule2.Appointment['L_Name'];
            label30.Caption:=datamodule2.Appointment['Phone'];
            checkbox7.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='12:30' then
    begin
      label32.Caption:=datamodule2.Appointment['Name'];
      label33.Caption:=datamodule2.Appointment['L_Name'];
      label34.Caption:=datamodule2.Appointment['Phone'];
      checkbox8.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='13:00' then
      begin
        label36.Caption:=datamodule2.Appointment['Name'];
        label37.Caption:=datamodule2.Appointment['L_Name'];
        label38.Caption:=datamodule2.Appointment['Phone'];
        checkbox9.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='13:30' then
        begin
          label40.Caption:=datamodule2.Appointment['Name'];
          label41.Caption:=datamodule2.Appointment['L_Name'];
          label42.Caption:=datamodule2.Appointment['Phone'];
          checkbox10.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='14:00' then
          begin
            label44.Caption:=datamodule2.Appointment['Name'];
            label45.Caption:=datamodule2.Appointment['L_Name'];
            label46.Caption:=datamodule2.Appointment['Phone'];
            checkbox11.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='18:00' then
    begin
      label48.Caption:=datamodule2.Appointment['Name'];
      label49.Caption:=datamodule2.Appointment['L_Name'];
      label50.Caption:=datamodule2.Appointment['Phone'];
      checkbox12.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='18:30' then
      begin
        label52.Caption:=datamodule2.Appointment['Name'];
        label53.Caption:=datamodule2.Appointment['L_Name'];
        label54.Caption:=datamodule2.Appointment['Phone'];
        checkbox13.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='19:00' then
        begin
          label56.Caption:=datamodule2.Appointment['Name'];
          label57.Caption:=datamodule2.Appointment['L_Name'];
          label58.Caption:=datamodule2.Appointment['Phone'];
          checkbox14.Checked:=true;
        end
      else
        if datamodule2.Appointment['A_time']='19:30' then
          begin
            label60.Caption:=datamodule2.Appointment['Name'];
            label61.Caption:=datamodule2.Appointment['L_Name'];
            label62.Caption:=datamodule2.Appointment['Phone'];
            checkbox15.Checked:=true;
          end
        end
      end
    end
  end
end
```

```
else
  if datamodule2.Appointment['A_time']='20:00' then
    begin
      label64.Caption:=datamodule2.Appointment['Name'];
      label65.Caption:=datamodule2.Appointment['L_Name'];
      label66.Caption:=datamodule2.Appointment['Phone'];
      checkbox16.Checked:=true;
    end
  else
    if datamodule2.Appointment['A_time']='20:30' then
      begin
        label68.Caption:=datamodule2.Appointment['Name'];
        label69.Caption:=datamodule2.Appointment['L_Name'];
        label70.Caption:=datamodule2.Appointment['Phone'];
        checkbox17.Checked:=true;
      end
    else
      if datamodule2.Appointment['A_time']='21:00' then
        begin
          label72.Caption:=datamodule2.Appointment['Name'];
          label73.Caption:=datamodule2.Appointment['L_Name'];
          label74.Caption:=datamodule2.Appointment['Phone'];
          checkbox18.Checked:=true;
        end;
      end;
    end;
  datamodule2.Appointment.Next;
end;
end;

procedure TAppointments.MonthCalendar1Click(Sender: TObject);
begin
  label3.Caption:=' ';
  label4.Caption:=' ';
  label6.Caption:=' ';
```

```
checkbox1.Checked:=false;
label8.Caption:=' ';
label9.Caption:=' ';
label10.Caption:=' ';
checkbox2.Checked:=false;
label12.Caption:=' ';
label13.Caption:=' ';
label14.Caption:=' ';
checkbox3.Checked:=false;
label16.Caption:=' ';
label17.Caption:=' ';
label18.Caption:=' ';
checkbox4.Checked:=false;
label20.Caption:=' ';
label21.Caption:=' ';
label22.Caption:=' ';
checkbox5.Checked:=false;
label24.Caption:=' ';
label25.Caption:=' ';
label26.Caption:=' ';
checkbox6.Checked:=false;
label28.Caption:=' ';
label29.Caption:=' ';
label30.Caption:=' ';
checkbox7.Checked:=false;
label32.Caption:=' ';
label33.Caption:=' ';
label34.Caption:=' ';
checkbox8.Checked:=false;
label36.Caption:=' ';
label37.Caption:=' ';
label38.Caption:=' ';
```



```
checkbox9.Checked:=false;
label40.Caption:=' ';
label41.Caption:=' ';
label42.Caption:=' ';
checkbox10.Checked:=false;
label44.Caption:=' ';
label45.Caption:=' ';
label46.Caption:=' ';
checkbox11.Checked:=false;
label48.Caption:=' ';
label49.Caption:=' ';
label50.Caption:=' ';
checkbox12.Checked:=false;
label52.Caption:=' ';
label53.Caption:=' ';
label54.Caption:=' ';
checkbox13.Checked:=false;
label56.Caption:=' ';
label57.Caption:=' ';
label58.Caption:=' ';
checkbox14.Checked:=false;
label60.Caption:=' ';
label61.Caption:=' ';
label62.Caption:=' ';
checkbox15.Checked:=false;
label64.Caption:=' ';
label65.Caption:=' ';
label66.Caption:=' ';
checkbox16.Checked:=false;
label68.Caption:=' ';
label69.Caption:=' ';
label70.Caption:=' ';
```

```
checkbox17.Checked:=false;
label72.Caption:=' ';
label73.Caption:=' ';
label74.Caption:=' ';
checkbox18.Checked:=false;
end ;

procedure TAppointments.Button2Click(Sender: TObject);
begin
    appointments.Close;
end;

end.
```

```
unit UnDataModule1;

interface

uses
  SysUtils, Classes, DB, DBTables;

type
  TDataModule2 = class(TDataModule)
    Visits: TTable;
    DSPatient: TDataSource;
    DSVisits: TDataSource;
    Appointment: TTable;
    DSAppointment: TDataSource;
    Patient: TTable;
    AppointmentName: TStringField;
    AppointmentL_Name: TStringField;
    AppointmentA_Date: TDateTimeField;
    AppointmentA_Time: TStringField;
    AppointmentPhone: TStringField;
    Sinergates: TTable;
    DSSinergates: TDataSource;
    SinergatesC_kode: TStringField;
    Sinergatesaddress: TStringField;
    SinergatesSector: TStringField;
    SinergatesE_Mail: TStringField;
    SinergatesSite: TStringField;
    SinergatesST: TStringField;
    Sinergatesphone: TStringField;
    SinergatesL_Name: TStringField;
    SinergatesName: TStringField;
    PatientPatient_Kode: TFloatField;
    PatientName: TStringField;
    PatientL_Name: TStringField;
    PatientF_Name: TStringField;
```

```
PatientS_Name: TStringField;
PatientBirthD: TDateTimeField;
PatientSex: TStringField;
PatientJob: TStringField;
PatientAddress: TStringField;
PatientPhone: TStringField;
PatientEnsurance: TStringField;
PatientEnsurance_Num: TStringField;
PatientHistoric: TStringField;
PatientVIC: TStringField;
PatientAM: TStringField;
PatientFamily: TStringField;
VisitsPatient_Kode: TFloatField;
VisitsV_Date: TDateTimeField;
VisitsVisit_Num: TIntegerField;
VisitsSymptoms: TStringField;
VisitsTests: TStringField;
VisitsDiagnose: TStringField;
VisitsMedication: TStringField;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  DataModule2: TDataModule2;

implementation

uses UnInput, UnWelcome, UnVisits, UnAppointment;

{$R *.dfm}

end.
```

unit UnDetail;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ComCtrls, DBCtrls, Mask, DB, DBTables, ExtCtrls,
Buttons;*

type

TForm1 = class(TForm)

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit1: TDBEdit;

DBNavigator1: TDBNavigator;

Query1: TQuery;

DataSource1: TDataSource;

Button1: TButton;

UpdateSQL1: TUpdateSQL;

Button2: TButton;

Label77: TLabel;

Label18: TLabel;

Label11: TLabel;

Label3: TLabel;

Label2: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Button3: TButton;

Label25: TLabel;

Label26: TLabel;

Label120: TLabel;

Label28: TLabel;

dbrichedit31: TDBRichEdit;

```
dbrichedit2: TDBRichEdit;
dbrichedit4: TDBRichEdit;
dbrichedit3: TDBRichEdit;
Label111: TLabel;
Label38: TLabel;
Label39: TLabel;
Query1Patient_Kode: TFloatField;
Query1V_Date: TDateTimeField;
Query1Visit_Num: TIntegerField;
Query1Symptoms: TStringField;
Query1Tests: TStringField;
Query1Diagnose: TStringField;
Query1Medication: TStringField;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure DBRichEdit3DbClick(Sender: TObject);
procedure DBRichEdit2DbClick(Sender: TObject);
procedure DBRichEdit4DbClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

uses UnDataModule1, UnVisits, UnInput;

{$R *.dfm}
```

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
begin
  form1.Close;
  visits.Show;
  datamodule2.Visits.Close;
  datamodule2.Visits.Open;
  datamodule2.Visits.First;
  while not datamodule2.Visits.Eof do
  begin
    if datamodule2.Visits['patient_kode']=DBEdit1.Text then
      a:= datamodule2.Visits['visit_num']+ 1;
      datamodule2.Visits.Next;
    end;
    datamodule2.Visits.Insert;
    datamodule2.Visits['patient_kode']:=form1.DBEdit1.Text;
    visits.DBEdit2.Text:=a;
    input.Close;
  end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  datamodule2.Visits.Close;
  datamodule2.Visits.Open;
  datamodule2.Visits.First;
  while not datamodule2.Visits.Eof do
  begin
    if (datamodule2.Visits['patient_kode']=dbedit1.Text) and
      (datamodule2.Visits['visit_num']=dbedit2.Text) then
      begin
        datamodule2.Visits.Edit;
        datamodule2.Visits['diagnose']:=ansiuppercase(dbrichedit3.Text);
        datamodule2.Visits['tests']:=ansiuppercase(dbrichedit2.Text);
        datamodule2.Visits['medication']:=ansiuppercase(dbrichedit4.Text);
```

```
        datamodule2.Visits.Post;
        query1.CachedUpdates:=false;
    end;
    datamodule2.Visits.Next;
end;
button2.Enabled:=false;
end;

procedure TForm1.DBRichEdit3DbClick(Sender: TObject);
begin
    query1.CachedUpdates:=true;
    button2.Enabled:=true;
    button1.Enabled:=false;
    Label18.Enabled:=false;
end;

procedure TForm1.DBRichEdit2DbClick(Sender: TObject);
begin
    query1.CachedUpdates:=true;
    button2.Enabled:=true;
    button1.Enabled:=false;
    Label18.Enabled:=false;
end;

procedure TForm1.DBRichEdit4DbClick(Sender: TObject);
begin
    query1.CachedUpdates:=true;
    button2.Enabled:=true;
    button1.Enabled:=false;
    Label18.Enabled:=false;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    button2.Enabled:=false;
end;
```



```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
```

```
begin
```

```
    datamodule2.Patient.Close;
```

```
    datamodule2.Patient.Open;
```

```
    datamodule2.Patient.last;
```

```
    input.Edit1.Text:=datamodule2.Patient['patient_kode']+ 1;
```

```
    input.RadioButton1.Checked:=false;
```

```
    input.RadioButton2.Checked:=false;
```

```
    input.Edit2.Text:='';
```

```
    input.Edit3.Text:='';
```

```
    input.Edit4.Text:='';
```

```
    input.Edit5.Text:='';
```

```
    input.DBEdit6.Text:='';
```

```
    input.DBEdit7.Text:=' ';
```

```
    input.DBEdit8.Text:=' ';
```

```
    input.DBEdit9.Text:=' ';
```

```
    input.DBComboBox1.Text:=' ';
```

```
    input.DBEdit11.Text:=' ';
```

```
    input.DBEdit12.Text:=' ';
```

```
    input.DBRichEdit1.Text:=' ';
```

```
    input.DBRichEdit2.Text:=' ';
```

```
    input.button4.Enabled:=false;
```

```
    input.button1.Enabled:=false;
```

```
    input.label18.Enabled:=false;
```

```
    input.label17.Enabled:=false;
```

```
    input.Show;
```

```
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
```

```
    Form1.Close;
```

```
end;
```

```
end.
```

unit UnSinergates;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, Grids, DBGrids, DBTables, StdCtrls;*

type

*TDBGridOption = (dgEditing, dgAlwaysShowEditor, dgTitles, dgIndicator,
dgColumnResize, dgColLines, dgRowLines, dgTabs, dgRowSelect,
dgAlwaysShowSelection, dgConfirmDelete, dgCancelOnExit, dgMultiSelect);*

TDBGridOptions = set of TDBGridOption;

TForm2 = class(TForm)

Query1: TQuery;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

UpdateSQL1: TUpdateSQL;

Label19: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Label18: TLabel;

Label1: TLabel;

Label2: TLabel;

Query1C_kode: TStringField;

Query1address: TStringField;

Query1Sector: TStringField;

Query1E_Mail: TStringField;

Query1Site: TStringField;

Query1ST: TStringField;

Query1phone: TStringField;

Query1L_Name: TStringField;

Query1Name: TStringField;

```
Label79: TLabel;
Label78: TLabel;
Label111: TLabel;
Label5: TLabel;
Edit1: TEdit;
Button5: TButton;
Label9: TLabel;
Label16: TLabel;
Label10: TLabel;
Label8: TLabel;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Query1UpdateRecord(DataSet: TDataSet;
    UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);
procedure FormShow(Sender: TObject);
procedure Button5Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form2: TForm2;
    VAR d:variant;

implementation

uses UnDataModule1, UnBackup, UnAppointment, UnDetail, UnInput, UnVisits,
    UnWelcome;

{$R *.dfm}
```

```
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    query1.Close;
end;
```

```
procedure TForm2.Query1UpdateRecord(DataSet: TDataSet;
UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
begin
    if ( UpdateKind = ukModify ) then
        begin
            UpdateSQL1.Apply(UpdateKind);
            UpdateAction := uaApplied;
        end
    else
        if ( UpdateKind = ukDelete ) then
            begin
                UpdateSQL1.Apply(UpdateKind);
                UpdateAction := uaApplied;
            end
        else
            begin
                UpdateAction := uaSkip;
                ShowMessage ('Error');
            end;
end;
```

```
procedure TForm2.Button1Click(Sender: TObject);
begin
    Query1.Database.ApplyUpdates([Query1]);
    button1.Enabled:=false;
    label18.Enabled:=false;
    label1.Enabled:=false;
    button2.Enabled:=false;
end;
```

```
procedure TForm2.Button3Click(Sender: TObject);
```

```
begin
```

```
  Query1.Close;
```

```
  edit1.Text:="";
```

```
end;
```

```
procedure TForm2.Button4Click(Sender: TObject);
```

```
begin
```

```
  sinergates.DBEdit4.Text:=' ';
```

```
  sinergates.DBEdit5.Text:=' ';
```

```
  sinergates.DBEdit6.Text:=' ';
```

```
  sinergates.DBEdit7.Text:=' ';
```

```
  sinergates.DBEdit8.Text:=' ';
```

```
  sinergates.DBEdit9.Text:=' ';
```

```
  form2.Close;
```

```
end;
```

```
procedure TForm2.Button2Click(Sender: TObject);
```

```
begin
```

```
  datamodule2.Sinergates.Close;
```

```
  datamodule2.Sinergates.Open;
```

```
  datamodule2.Sinergates.first;
```

```
  while not datamodule2.Sinergates.Eof do
```

```
    begin
```

```
      if datamodule2.Sinergates['C_Kode']=d then
```

```
        if application.MessageBox('ÄéáñáöÐ ÆñÝ÷ïóáò ÄãñáöÐ', 'ÄêäääBôáé
```

```
ÄéáñáöÐ',mb_okcancel + mb_iconquestion)=idok then
```

```
          begin
```

```
            datamodule2.Sinergates.Delete;
```

```
          end;
```

```
          datamodule2.Sinergates.Next;
```

```
        end;
```

```
end;
```

```
procedure TForm2.DBGrid1CellClick(Column: TColumn);
```

```
begin
```

```
  d:=query1.FieldValues['C_Kode'];
```

```
  button2.Enabled:=true;
```

```
  label1.Enabled:=true;
```

```
  button1.Enabled:=true;
```

```
  label18.Enabled:=true;
```

```
end;
```

```
procedure TForm2.FormShow(Sender: TObject);
```

```
begin
```

```
  button2.Enabled:=false;
```

```
  label1.Enabled:=false;
```

```
  button1.Enabled:=false;
```

```
  label18.Enabled:=false;
```

```
end;
```

```
procedure TForm2.Button5Click(Sender: TObject);
```

```
begin
```

```
  query1.Close;
```

```
  query1.Prepare;
```

```
  query1.Params[0].Value:=edit1.Text;
```

```
  query1.Open;
```

```
end;
```

```
end.
```

```
unit UnPatient;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, StdCtrls, ExtCtrls, ComCtrls, DBCtrls, Mask, DB,  
DBTables;
```

```
type
```

```
TForm3 = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
Label10: TLabel;
```

```
Label11: TLabel;
```

```
Label12: TLabel;
```

```
Label13: TLabel;
```

```
Label15: TLabel;
```

```
Label16: TLabel;
```

```
Edit2: TEdit;
```

```
Edit3: TEdit;
```

```
Edit4: TEdit;
```

```
Edit5: TEdit;
```

```
DBGrid1: TDBGrid;
```

```
Query1: TQuery;
```

```
DataSource1: TDataSource;
```

```
Edit1: TEdit;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Label14: TLabel;  
Label17: TLabel;  
UpdateSQL1: TUpdateSQL;  
Edit6: TEdit;  
Edit7: TEdit;  
Edit8: TEdit;  
Edit9: TEdit;  
Edit10: TEdit;  
Edit11: TEdit;  
Edit12: TEdit;  
RichEdit1: TRichEdit;  
Button3: TButton;  
Button4: TButton;  
Label6: TLabel;  
Label18: TLabel;  
Richedit2: TRichEdit;  
Label19: TLabel;  
Label120: TLabel;  
Label110: TLabel;  
Label24: TLabel;  
Label26: TLabel;  
Label29: TLabel;  
Query1Patient_Kode: TFloatField;  
Query1V_Date: TDateTimeField;  
Query1Visit_Num: TIntegerField;  
Query1Symptoms: TStringField;  
Query1Tests: TStringField;  
Query1Diagnose: TStringField;  
Query1Medication: TStringField;  
procedure FormShow(Sender: TObject);  
procedure Button1Click(Sender: TObject);  
procedure Query1UpdateRecord(DataSet: TDataSet;  
    UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
```



```
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form3: TForm3;
  k:tbookmark;

implementation

uses UnAppointment, UnBackup, UnDataModule1, UnDetail, UnInput,
  UnSinergates, UnStatistics, UnVisits, UnWelcome, UnSearch;

{$R *.dfm}

procedure TForm3.FormShow(Sender: TObject);
begin
  form3.query1.Close;
  form3.query1.Prepare;
  form3.Edit1.Text:=FORM4.Query1.FieldValues['Patient_Kode'];
  form3.Query1.Params[0].Value:=Edit1.Text;
  form3.query1.Open;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  datamodule2.Patient.Close;
  datamodule2.Patient.Open;
  datamodule2.Patient.First;
```

```
while not datamodule2.Patient.Eof do
  begin
    if datamodule2.Patient['patient_kode']=edit1.Text then
      begin
        k:=datamodule2.Patient.getbookmark;
      end;
      datamodule2.Patient.Next;
    end;
    datamodule2.Patient.gotobookmark(k);
    datamodule2.Patient.Edit;
    datamodule2.Patient['name']:=AnsiUppercase(edit2.Text);
    datamodule2.Patient['L_Name']:=AnsiUppercase(edit3.Text);
    datamodule2.Patient['F_Name']:=AnsiUppercase(edit4.Text);
    datamodule2.Patient['S_Name']:=AnsiUppercase(edit5.Text);
    datamodule2.Patient['BirthD']:=AnsiUppercase(edit6.Text);
    datamodule2.Patient['Job']:=AnsiUppercase(edit7.Text);
    datamodule2.Patient['Address']:=AnsiUppercase(edit8.Text);
    datamodule2.Patient['Phone']:=AnsiUppercase(edit9.Text);
    datamodule2.Patient['Ensurance']:=AnsiUppercase(edit10.Text);
    datamodule2.Patient['Ensurance_Num']:=AnsiUppercase(edit8.Text);
    datamodule2.Patient['A.M']:=AnsiUppercase(edit9.Text);
    datamodule2.Patient['Historic']:=AnsiUppercase(Richedit1.Text);
    datamodule2.Patient['VIC']:=AnsiUppercase(Richedit2.Text);
    datamodule2.Patient.Post;
    Query1.Database.ApplyUpdates([Query1]);
  end;
```

```
procedure TForm3.Query1UpdateRecord(DataSet: TDataSet;
  UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
begin
  if ( UpdateKind = ukModify ) then
    begin
      UpdateSQL1.Apply(UpdateKind);
      UpdateAction := uaApplied;
    end
  else
    if ( UpdateKind = ukDelete ) then
      begin
        UpdateSQL1.Apply(UpdateKind);
        UpdateAction := uaApplied;
      end
    else
      begin
        UpdateAction := uaSkip;
        ShowMessage ('Error');
      end;
  end;
```

```
procedure TForm3.Button3Click(Sender: TObject);
begin
  form3.Close;
end;
```

```
procedure TForm3.Button4Click(Sender: TObject);
begin
  form4.Close;
  form3.Close;
  input.Show;
end;
```

```
procedure TForm3.Button2Click(Sender: TObject);
var a:string;
begin
  form4.Close;
  form3.Close;
  visits.Show;
  datamodule2.Visits.Close;
  datamodule2.Visits.Open;
  datamodule2.Visits.First;
  while not datamodule2.Visits.Eof do
    begin
      if datamodule2.Visits['patient_kode']=Edit1.Text then
        a:= datamodule2.Visits['visit_num']+ 1;
        datamodule2.Visits.Next;
      end;
    datamodule2.Visits.Insert;
    datamodule2.Visits['patient_kode']:=Edit1.Text;
    visits.DBEdit2.Text:=a;
    input.Close;
  end;
end.
```

```
unit UnSearch;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, Grids, DBGrids, StdCtrls;
```

```
type
```

```
TForm4 = class(TForm)  
Label14: TLabel;  
Label10: TLabel;  
Label1: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Edit3: TEdit;  
Button1: TButton;  
Button3: TButton;  
Button4: TButton;  
Query1: TQuery;  
DataSource1: TDataSource;  
Label19: TLabel;  
Label5: TLabel;  
Label16: TLabel;  
Label2: TLabel;  
DBGrid1: TDBGrid;  
Query1Patient_Kode: TFloatField;  
Query1Name: TStringField;  
Query1L_Name: TStringField;  
Query1F_Name: TStringField;  
Query1S_Name: TStringField;  
Query1BirthD: TDateTimeField;  
Query1Sex: TStringField;  
Query1Job: TStringField;  
Query1Address: TStringField;
```

```
Query1Phone: TStringField;
Query1Ensurance: TStringField;
Query1Ensurance_Num: TStringField;
Query1Historic: TStringField;
Query1VIC: TStringField;
Query1AM: TStringField;
Query1Family: TStringField;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;

implementation

uses UnDataModule1, UnPatient, UnInput;

{$R *.dfm}

procedure TForm4.FormCreate(Sender: TObject);
begin
  edit3.Text:="";
end;
```

```
procedure TForm4.Button1Click(Sender: TObject);
begin
    query1.Close;
    query1.Prepare;
    query1.Params[0].Value:=ansiuppercase(edit3.Text);
    query1.Open;
end;

procedure TForm4.DBGrid1CellClick(Column: TColumn);
begin
    form3.Show;
    form3.Edit2.Text:= query1.FieldValues['Name'];
    form3.Edit3.Text:= query1.FieldValues['L_Name'];
    form3.Edit4.Text:= query1.FieldValues['F_Name'];
    form3.Edit5.Text:= query1.FieldValues['S_Name'];
    form3.Edit1.Text:= query1.FieldValues['Patient_Kode'];
    form3.Edit6.Text:= query1.FieldValues['BirthD'];
    form3.Edit7.Text:= query1.FieldValues['Job'];
    form3.Edit8.Text:= query1.FieldValues['Address'];
    form3.Edit9.Text:= query1.FieldValues['Phone'];
    form3.Edit10.Text:= query1.FieldValues['Ensurance'];
    form3.Edit11.Text:= query1.FieldValues['Ensurance_Num'];
    form3.Edit12.Text:= query1.FieldValues['A.M'];
    form3.RichEdit1.Text:= query1.FieldValues['Historic'];
    form3.RichEdit2.Text:= query1.FieldValues['VIC'];
end;

procedure TForm4.Button3Click(Sender: TObject);
begin
    edit3.Text:="";
    query1.Close;
end;
```

```
procedure TForm4.Button4Click(Sender: TObject);
begin
    form4.Close;
end;

procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    input.Close;
    edit3.Text:="";
    query1.Close;
end;

end.
```


unit UnInput;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, DBCtrls, Mask, ExtCtrls;*

type

TBOOKMARK=POINTER;

TInput = class(TForm)

DBEdit7: TDBEdit;

DBEdit8: TDBEdit;

DBEdit9: TDBEdit;

DBEdit12: TDBEdit;

DBRichEdit1: TDBRichEdit;

DBRichEdit2: TDBRichEdit;

DBEdit6: TDBEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label15: TLabel;

Button1: TButton;

Button2: TButton;

Button4: TButton;

DBEdit11: TDBEdit;
Label16: TLabel;
RadioGroup1: TRadioGroup;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Label14: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Button3: TButton;
Button5: TButton;
Label20: TLabel;
Label21: TLabel;
Button6: TButton;
Label109: TLabel;
Label110: TLabel;
Label111: TLabel;
Label113: TLabel;
Edit1: TEdit;
Label107: TLabel;
Label87: TLabel;
Label88: TLabel;
Label89: TLabel;
Label90: TLabel;
Label91: TLabel;
Label92: TLabel;
Label93: TLabel;
DBComboBox1: TDBComboBox;

```
procedure Button4Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure DBEdit6Enter(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure DBEdit7Exit(Sender: TObject);
procedure DBEdit8Exit(Sender: TObject);
procedure DBEdit9Exit(Sender: TObject);
procedure DBEdit10Exit(Sender: TObject);
procedure DBEdit11Exit(Sender: TObject);
procedure DBEdit12Exit(Sender: TObject);
procedure DBRichEdit1Exit(Sender: TObject);
procedure DBRichEdit2Exit(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Input: TInput;
  B:TBOOKMARK;
  a:variant;
  c,f:variant;

implementation

uses UnWelcome, UnDataModule1, UnVisits, UnAppointment, UnDetail,
  UnStatistics, UnSearch;

{$R *.dfm}
```

```
procedure TInput.Button4Click(Sender: TObject);
```

```
begin
```

```
  form1.Show;
```

```
  datamodule2.Visits.Close;
```

```
  datamodule2.Visits.Open;
```

```
  datamodule2.Visits.First;
```

```
  form1.QUERY1.Close;
```

```
  form1.query1.Prepare;
```

```
  form1.query1.Params[0].Value:=edit1.Text;
```

```
  form1.query1.Open;
```

```
  form1.button1.Enabled:=true;
```

```
  Form1.Label18.Enabled:=true;
```

```
  Form1.Label111.Caption:=edit2.Text;
```

```
  Form1.Label38.Caption:=edit3.Text;
```

```
end;
```

```
procedure TInput.Button1Click(Sender: TObject);
```

```
begin
```

```
  if edit5.Text <> '' then
```

```
    datamodule2.Patient['Family']:= 'ΆΆΆΆΪΪΌ';
```

```
  if edit5.Text = '' then
```

```
    datamodule2.Patient['Family']:= 'ΑΆΑΪΪΌ';
```

```
  if radiobutton1.Checked = true then
```

```
    datamodule2.Patient['sex']:= 'ΆΌΪΆΈΈΆ';
```

```
  if radiobutton2.Checked=true then
```

```
    datamodule2.Patient['sex']:= 'ΆΪΆÑΆΌ';
```

```
  datamodule2.Patient['Patient_Kode']:= edit1.Text;
```

```
  datamodule2.Patient['name']:= ansiuppercase(edit2.Text);
```

```
  datamodule2.Patient['L_name']:=ansiuppercase(edit3.Text);
```

```
  datamodule2.Patient['F_name']:= ansiuppercase(edit4.text);
```

```
  datamodule2.Patient['S_name']:= ansiuppercase(edit5.Text);
```

```
  datamodule2.Patient['address']:= ansiuppercase(dbedit8.Text);
```

```
  datamodule2.Patient['job']:=ansiuppercase(dbedit7.Text);
```

```
  datamodule2.Patient['ensurance']:= ansiuppercase(dbcombobox1.text);
```

```
datamodule2.Patient['ensurance_Num']:= ansiuppercase(dbedit11.Text);
datamodule2.Patient['A.M']:=ansiuppercase(dbedit12.Text);
datamodule2.Patient['historic']:= ansiuppercase(dbrichedit1.text);
datamodule2.Patient['VIC']:= ansiuppercase(dbrichedit2.Text);
datamodule2.Patient.Post;
Visits.Show;
input.Close;
datamodule2.Visits.Close;
datamodule2.Visits.Open;
datamodule2.Visits.Insert;
datamodule2.Visits['visit_num']:=1;
datamodule2.Visits['patient_kode']:=a+ 1;
end;

procedure TInput.Button2Click(Sender: TObject);
begin
  datamodule2.Patient.close;
  datamodule2.Patient.Open;
  datamodule2.Patient.Last;
  a:= datamodule2.Patient['patient_kode'];
  c:=1+ a;
  edit1.text:=c;
  button4.Enabled:=false;
  button1.Enabled:=false;
  label18.Enabled:=false;
  label17.Enabled:=false;
  input.radiobutton1.Checked:=false;
  input.radiobutton2.Checked:=false;
  input.button1.Enabled:=FALSE;
  input.button4.Enabled:=false;
  input.label18.Enabled:=false;
  input.label17.Enabled:=FALSE;
  input.DBEdit6.Text:='';
  input.DBEdit7.Text:=' ';
```

```
input.DBEdit8.Text:=' ';
input.DBEdit9.Text:=' ';
input.DBComboBox1.Text:=' ';
input.DBEdit11.Text:=' ';
input.DBEdit12.Text:=' ';
input.DBRichEdit1.Text:=' ';
input.DBRichEdit2.Text:=' ';
input.Edit2.Text:='';
input.Edit3.Text:='';
input.Edit4.Text:='';
input.Edit5.Text:='';
end;

procedure TInput.Button3Click(Sender: TObject);
begin
    input.Close;
end;

procedure TInput.Button5Click(Sender: TObject);
begin
    form4.Show;
end;

procedure TInput.Button6Click(Sender: TObject);
begin
    Appointments.Show;
end;

procedure TInput.DBEdit6Enter(Sender: TObject);
var d:boolean;
begin
    datamodule2.Patient.Close;
    datamodule2.Patient.Open;
    datamodule2.Patient.First;
```

```
IF datamodule2.Patient.Eof = true then
  begin
    d:=false;
    button1.Enabled:=true;
    label17.Enabled:=true;
    dbrichedit2.Enabled:=true;
  end
else
  begin
    d:=false;
    while not datamodule2.Patient.Eof do
      begin
        if (datamodule2.Patient['name']=ansiuppercase(edit2.Text)) and
          (datamodule2.Patient['L_name']=ansiuppercase(edit3.Text) )and
          (datamodule2.Patient['F_name']=ansiuppercase(edit4.Text)) and
          (datamodule2.Patient['S_name']=ansiuppercase(edit5.Text)) then
          begin
            B:= datamodule2.Patient.GetBookmark;
            edit1.Text:=datamodule2.Patient['patient_Kode'];
            button4.Enabled:=true;
            button1.Enabled:=false;
            dbrichedit2.Enabled:=false;
            label18.Enabled:=true;
            label17.Enabled:=false;
            if datamodule2.Patient['sex']='♂' then
              radiobutton1.Checked:=true;
            if datamodule2.Patient['sex']='♀' then
              radiobutton2.Checked:=true;
            form1.DBEdit1.Text:=datamodule2.Patient['patient_kode'];
            d:=true;
            exit;
          end;
        datamodule2.Patient.Next;
      end;
    end;
  end;
```

```
if d=true then
  begin
    datamodule2.Patient.GotoBookmark(B);
  end
else
  if d = false then
    Begin
      datamodule2.Patient.Last;
      a:= datamodule2.Patient['patient_kode'];
      datamodule2.Patient.Insert;
      input.DBEdit6.Text:='';
      input.DBEdit7.Text:=' ';
      input.DBEdit8.Text:=' ';
      input.DBEdit9.Text:=' ';
      input.DBComboBox1.Text:=' ';
      input.DBEdit11.Text:=' ';
      input.DBEdit12.Text:=' ';
      input.DBRichEdit1.Text:=' ';
      input.DBRichEdit2.Text:=' ';
      f:=a+ 1;
      edit1.Text:=f;
      radiobutton1.Checked:=false;
      radiobutton2.Checked:=false;
      button1.Enabled:=true;
      button4.Enabled:=false;
      label18.Enabled:=false;
      label17.Enabled:=true;
    end;
  end;
end;
```



```
procedure TInput.FormShow(Sender: TObject);
begin
  datamodule2.Patient.Close;
  datamodule2.Patient.Open;
  datamodule2.Patient.First;
  if datamodule2.Patient.Eof=true then
    begin
      c:=0;
      a:=0;
    end
  else
    begin
      datamodule2.Patient.Last;
      a:= datamodule2.Patient['patient_kode'];
    end;
  c:=1+ a;
  datamodule2.Patient.Insert;
  edit1.text:=c;
  button4.Enabled:=false;
  button1.Enabled:=false;
  label18.Enabled:=false;
  label17.Enabled:=false;
  input.radiobutton1.Checked:=false;
  input.radiobutton2.Checked:=false;
  input.DBEdit6.Text:='';
  input.DBEdit7.Text:=' ';
  input.DBEdit8.Text:=' ';
  input.DBEdit9.Text:=' ';
  input.DBComboBox1.Text:=' ';
  input.DBEdit11.Text:=' ';
  input.DBEdit12.Text:=' ';
  input.DBRichEdit1.Text:=' ';
  input.DBRichEdit2.Text:=' ';
  input.Edit2.Text:='';
```

```
input.Edit3.Text:="";
input.Edit4.Text:=" ";
input.Edit5.Text:="";
end;

procedure TInput.DBEdit7Exit(Sender: TObject);
begin
  if DBEdit7.Text = " then
    input.DBEdit7.Text:=' ';
end;

procedure TInput.DBEdit8Exit(Sender: TObject);
begin
  if DBEdit8.Text = " then
    input.DBEdit8.Text:=' ';
end;

procedure TInput.DBEdit9Exit(Sender: TObject);
begin
  if DBEdit9.Text = " then
    input.DBEdit9.Text:=' ';
end;

procedure TInput.DBEdit10Exit(Sender: TObject);
begin
  if DBComboBox1.Text = " then
    input.DBComboBox1.Text:=' ';
end;

procedure TInput.DBEdit11Exit(Sender: TObject);
begin
  if DBEdit11.Text = " then
    input.DBEdit11.Text:=' ';
end;
```

```
procedure TInput.DBEdit12Exit(Sender: TObject);
begin
  if DBEdit12.Text = '' then
    input.DBEdit12.Text:=' ';
end;

procedure TInput.DBRichEdit1Exit(Sender: TObject);
begin
  if DBRichEdit1.Text = '' then
    input.DBRichEdit1.Text:=' ';
end;

procedure TInput.DBRichEdit2Exit(Sender: TObject);
begin
  if DBRichEdit2.Text = '' then
    input.DBRichEdit2.Text:=' ';
end;

end.
```

```
unit UnBackup;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Mask, DBCtrls;
```

```
type
```

```
TBOOKMARK=POINTER;
```

```
Tsinergates = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
DBEdit4: TDBEdit;
```

```
DBEdit5: TDBEdit;
```

```
DBEdit6: TDBEdit;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
DBEdit7: TDBEdit;
```

```
DBEdit8: TDBEdit;
```

```
Label6: TLabel;
```

```
DBEdit9: TDBEdit;
```

```
Button1: TButton;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
Edit3: TEdit;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Button4: TButton;
```

```
Label19: TLabel;  
Label14: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Button5: TButton;  
Label13: TLabel;  
Label79: TLabel;  
Label110: TLabel;  
Label111: TLabel;  
Label108: TLabel;  
Label122: TLabel;  
Button6: TButton;  
Label40: TLabel;  
procedure Button1Click(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure Button2Click(Sender: TObject);  
procedure Button3Click(Sender: TObject);  
procedure Button4Click(Sender: TObject);  
procedure Button5Click(Sender: TObject);  
procedure Button6Click(Sender: TObject);  
procedure Edit2Enter(Sender: TObject);  
  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```

```
var
  sinergates: Tsinergates;
  c:tbookmark;
implementation

uses UnDataModule1, UnSinergates, UnAppointment, UnDetail, UnInput,
  UnVisits, UnWelcome, UnStatistics, UnSearch, UnPatient ;

{$R *.dfm}

procedure Tsinergates.Button1Click(Sender: TObject);
begin
  datamodule2.Sinergates['Name']:=ansiuppercase(edit2.Text);
  datamodule2.Sinergates['L_Name']:=ansiuppercase(edit3.Text);
  datamodule2.Sinergates['sector']:=ansiuppercase(dbedit4.Text);
  datamodule2.Sinergates['address']:=ansiuppercase(dbedit5.Text);
  datamodule2.Sinergates['S.T. ']:=ansiuppercase(dbedit9.Text);
  datamodule2.Sinergates.Post;
  edit1.Text:="";
  edit2.Text:="";
  edit3.Text:="";
  Button1.Enabled:=false;
  datamodule2.Sinergates.Insert;
end;

procedure Tsinergates.FormCreate(Sender: TObject);
begin
  edit1.Text:="";
  edit2.Text:="";
  edit3.Text:="";
  Button1.Enabled:=false;
  label14.Enabled:=false;
  Button2.Enabled:=false;
  label10.Enabled:=false;
```

```

Button3.Enabled:=false;
label11.Enabled:=false;
datamodule2.Sinergates.Close;
datamodule2.Sinergates.Open;
datamodule2.Sinergates.Insert;
datamodule2.Sinergates['C_Kode']:=edit1.Text;
end;

procedure Tsinergates.Button2Click(Sender: TObject);
begin
  datamodule2.Sinergates.Open;
  datamodule2.Sinergates.First;
  while not datamodule2.Sinergates.Eof do
    begin
      if datamodule2.Sinergates['C_Kode']= edit1.Text then
        begin
          datamodule2.Sinergates.Edit;
          datamodule2.Sinergates['Name']:=edit2.Text;
          datamodule2.Sinergates['L_Name']:=edit3.Text;
          datamodule2.Sinergates.post;
          exit;
        end;
      datamodule2.Sinergates.Next;
    end;
  end;

procedure Tsinergates.Button3Click(Sender: TObject);
begin
  if application.MessageBox('ÄéáñáöÐ      ÔñÝ÷ïóóò      ÄãñáöÐ', 'Äêâëåßôé
ÄéáñáöÐ', mb_okcancel + mb_iconquestion)=idok then
    begin
      datamodule2.Sinergates.Delete;
      datamodule2.Sinergates.Insert;
    end;
end;

```

```
edit1.Text:="";  
edit2.Text:="";  
edit3.Text:="";  
showmessage('1 ΆãñáöÐ ÄéãñÛöç.');
```

```
end;  
end;  
  
procedure Tsinergates.Button4Click(Sender: TObject);
```

```
begin  
edit1.Text:="";  
edit2.Text:="";  
edit3.Text:="";  
dbedit4.Text:="";  
dbedit5.Text:="";  
dbedit6.Text:="";  
dbedit7.Text:="";  
dbedit8.Text:="";  
dbedit9.Text:="";  
Button1.Enabled:=false;  
label14.Enabled:=false;  
Button2.Enabled:=false;  
label10.Enabled:=false;  
Button3.Enabled:=false;  
label11.Enabled:=false;  
end;
```

```
procedure Tsinergates.Button5Click(Sender: TObject);
```

```
begin  
sinergates.Close;  
end;
```



```
procedure Tsinergates.Button6Click(Sender: TObject);
begin
    form2.show;
end;

procedure Tsinergates.Edit2Enter(Sender: TObject);
var d:boolean;

begin
    datamodule2.Sinergates.Close;
    datamodule2.Sinergates.Open;
    datamodule2.Sinergates.First;
    d:=false;
    while not datamodule2.Sinergates.Eof do
        begin
            if datamodule2.Sinergates['C_Kode'] = edit1.Text then
                begin
                    c:=datamodule2.Sinergates.GetBookmark;
                    d:=true;
                end;
            datamodule2.Sinergates.Next;
        end;
    if (d=true) then
        begin
            Button1.Enabled:=false;
            label14.Enabled:=false;
            Button2.Enabled:=true;
            label10.Enabled:=true;
            Button3.Enabled:=true;
            label11.Enabled:=true;
            datamodule2.Sinergates.GotoBookmark(c);
            edit1.Text:=datamodule2.Sinergates['C_Kode'];
            edit2.Text:=datamodule2.Sinergates['name'];
            edit3.Text:=datamodule2.Sinergates['L_Name'];
        end
    end
```

```
else
  Begin
    datamodule2.Sinergates.Last;
    datamodule2.Sinergates.Insert;
    datamodule2.Sinergates['C_Kode']:=edit1.Text;
    datamodule2.Sinergates['Name']:=edit2.Text;
    datamodule2.Sinergates['L_Name']:=edit3.Text;
    button1.Enabled:=true;
    label14.Enabled:=true;
  end;
end;

end.
```

unit UnStatistics;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, DBTables;*

type

Tstatistiks = class(TForm)

Button1: TButton;

Button3: TButton;

Label19: TLabel;

Label14: TLabel;

Label10: TLabel;

Button4: TButton;

Label1: TLabel;

Label3: TLabel;

Label79: TLabel;

Label111: TLabel;

Label4: TLabel;

Edit1: TEdit;

Label6: TLabel;

Label13: TLabel;

Label15: TLabel;

Label17: TLabel;

Label18: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

Label33: TLabel;

Label35: TLabel;

Label36: TLabel;

Label37: TLabel;

Label38: TLabel;

Label5: TLabel;

Label11: TLabel;

Label20: TLabel;

Label28: TLabel;

Label34: TLabel;

Label40: TLabel;

Label41: TLabel;

Label42: TLabel;

Label43: TLabel;

Label44: TLabel;

Label45: TLabel;

Label46: TLabel;

Label47: TLabel;

Label48: TLabel;

Label7: TLabel;

Label8: TLabel;

Label12: TLabel;

Label39: TLabel;

Label49: TLabel;

Label50: TLabel;

Label51: TLabel;

Label2: TLabel;

Label52: TLabel;

Label53: TLabel;

Label54: TLabel;

Label55: TLabel;

Label56: TLabel;
Label82: TLabel;
Label83: TLabel;
Label84: TLabel;
Label85: TLabel;
Label87: TLabel;
Label88: TLabel;
Label89: TLabel;
Label86: TLabel;
Label90: TLabel;
Label91: TLabel;
Label92: TLabel;
Label93: TLabel;
Label94: TLabel;
Label95: TLabel;
Label96: TLabel;
Label97: TLabel;
Label103: TLabel;
Label104: TLabel;
Label108: TLabel;
Label109: TLabel;
Label110: TLabel;
Label112: TLabel;
Label113: TLabel;
Label114: TLabel;
Label106: TLabel;
Label107: TLabel;
Label115: TLabel;
Label116: TLabel;
Label117: TLabel;
Label125: TLabel;
Label126: TLabel;
Label127: TLabel;
Label128: TLabel;

```
Label105: TLabel;  
Label129: TLabel;  
Label58: TLabel;  
Label59: TLabel;  
Label133: TLabel;  
Label134: TLabel;  
Label135: TLabel;  
Label60: TLabel;  
procedure Button4Click(Sender: TObject);  
procedure FormShow(Sender: TObject);  
procedure Button1Click(Sender: TObject);  
procedure Button3Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;  
  
var  
    statistiks: Tstatistiks;  
  
implementation  
  
uses UnDataModule1, UnBackup, UnInput;  
  
{$R *.dfm}  
  
procedure Tstatistiks.Button4Click(Sender: TObject);  
begin  
    statistiks.Close;  
    Input.Close;  
end;
```

```
procedure Tstatistiks.FormShow(Sender: TObject);
var m,w,mu,me,wu,we,ika,dim,teve,oga,alli,unens,sum: variant;
begin
  datamodule2.Patient.Close;
  datamodule2.Patient.Open;
  datamodule2.Patient.Last;
  label38.Caption:=datamodule2.Patient['Patient_kode'];
  datamodule2.Patient.First;
  m:=0; mu:=0; me:=0; w:=0; wu:=0; we:=0;
  ika:=0; dim:=0; teve:=0; oga:=0; alli:=0; unens:=0;
  while not datamodule2.Patient.Eof do
    begin
      if datamodule2.Patient['Ensurance']='' then
        unens:=unens+ 1
      else
        if datamodule2.Patient['Ensurance']='É. Ê. Á.' then
          ika:=ika+ 1
        else
          if datamodule2.Patient['Ensurance']='ÄÇÏŒÉĬ' then
            dim:=dim+ 1
          else
            if datamodule2.Patient['Ensurance']='Ô. Å. Â. Ã.' then
              teve:=teve+ 1
            else
              if datamodule2.Patient['Ensurance']='İ. Ñ. Á.' then
                oga:=oga+ 1
              else
                begin
                  alli:=alli+ 1;
                end;
            end;
          end;
        end;
      end;
    end;
```

```

if datamodule2.Patient['sex'] = 'ΑΙΆÑÁÓ' then
  begin
    m:=m+ 1;
  end
else
  begin
    w:=w+ 1;
  end;
if (datamodule2.Patient['sex'] = 'ΑΙΆÑÁÓ') and (datamodule2.Patient['job'] = '') then
  begin
    mu:=mu+ 1;
  end
else
if (datamodule2.Patient['sex'] = 'ΑΙΆÑÁÓ') and (datamodule2.Patient['job'] <> '') then
  begin
    me:=me+ 1;
  end;
if (datamodule2.Patient['sex'] = 'ÃŒÍAÉÊÁ') and (datamodule2.Patient['job'] = '') then
  begin
    wu:=wu+ 1;
  end
else
if (datamodule2.Patient['sex'] = 'ÃŒÍAÉÊÁ') and (datamodule2.Patient['job'] <> '') then
  begin
    we:=we+ 1;
  end;
datamodule2.Patient.Next;
end;

sum:=ika+ dim+ teve+ oga+ alli;
label26.Caption:=m;
label27.Caption:=mu;
label32.Caption:=me;
label31.Caption:=w;

```



```
label30.Caption:=wu;
label29.Caption:=we;
label42.Caption:=dim;
label43.Caption:=ika;
label44.Caption:=teve;
label45.Caption:=oga;
label46.Caption:=alli;
label47.Caption:=sum;
label48.Caption:=unens;
if (w+ m<>0) then
  Begin
    sum:=int(sum*100/(w+ m));
    label86.Caption:=sum;
    sum:=int(unens*100/(w+ m));
    label90.Caption:=sum;
  end
else
  Begin
    label86.Caption:='0';
    label90.Caption:='0';
  end;
if ((ika+ dim+ teve+ oga+ alli)<>0) then
  Begin
    sum:=int(dim*100/(ika+ dim+ teve+ oga+ alli));
    label93.Caption:=sum;
    sum:=int(ika*100/(ika+ dim+ teve+ oga+ alli));
    label94.Caption:=sum;
    sum:=int(teve*100/(ika+ dim+ teve+ oga+ alli));
    label95.Caption:=sum;
    sum:=int(oga*100/(ika+ dim+ teve+ oga+ alli));
    label96.Caption:=sum;
    sum:=int(alli*100/(ika+ dim+ teve+ oga+ alli));
    label97.Caption:=sum;
  end
end
```

else

Begin

label93.Caption:='0';

label94.Caption:='0';

label95.Caption:='0';

label96.Caption:='0';

label97.Caption:='0';

end;

if (w<>0) or (m<>0) then

Begin

*sum:=int(m*100/(m+ w));*

label85.Caption:=sum;

*sum:=int(w*100/(m+ w));*

label88.Caption:=sum;

if m<>0 then

Begin

*sum:=int(mu*100/m);*

label52.Caption:=sum;

*sum:=int(me*100/m);*

label53.Caption:=sum;

end

else

Begin

label52.Caption:='0';

label53.Caption:='0';

end;

if w<>0 then

Begin

*sum:=int(wu*100/w);*

label54.Caption:=sum;

*sum:=int(we*100/w);*

label55.Caption:=sum;

end

```
else
  Begin
    label54.Caption:=sum;
    label55.Caption:=sum;
  end;
end
else
  Begin
    label85.Caption:='0';
    label88.Caption:='0';
  end;
label36.Caption:='';
Label134.Caption:='';
Label135.Caption:='';
label35.Caption:='';
label33.Caption:='';
label37.Caption:='';
label50.Caption:='';
label49.Caption:='';
label103.Caption:='';
label104.Caption:='';
Label108.Caption:='';
Label109.Caption:='';
Label110.Caption:='';
Label112.Caption:='';
Label113.Caption:='';
Label114.Caption:='';
Label106.Caption:='';
Label107.Caption:='';
Label115.Caption:='';
Label116.Caption:='';
Label121.Caption:='';
Label123.Caption:='';
Label124.Caption:='';
```

```
Label122.Caption:="";
Label58.Caption:="";
end;

procedure Tstatistiks.Button1Click(Sender: TObject);
var sum,mj,wj,muj,wuj,md,wd:variant;
d:boolean;
begin
datamodule2.Visits.Close;
datamodule2.Patient.Close;
datamodule2.Patient.Open;
datamodule2.Visits.Open;
datamodule2.Patient.First;
datamodule2.Visits.First;
mj:=0;
wj:=0;
muj:=0;
wuj:=0;
md:=0;
wd:=0;
while not datamodule2.Patient.Eof do
begin
datamodule2.Visits.First;
while not datamodule2.Visits.Eof do
begin
if (datamodule2.Visits['Diagnose'] = ansiuppercase(edit1.Text)) Then
if (datamodule2.Visits['patient_kode'] = (datamodule2.Patient['patient_kode'] ) then
begin
if (datamodule2.Patient['sex'] = 'ΑΉÑÁÓ') and
(datamodule2.Patient['job'] = '') then
begin
muj:=muj+ 1;
end;
end;
end;
end;
end;
end;
```

```
if (datamodule2.Patient['sex'] = 'ΑΙΆÑÁÓ') and
  (datamodule2.Patient['job'] <> '') then
  begin
    mj:=mj+ 1;
  end;
if (datamodule2.Patient['sex'] = 'ĂŒÍAÉÊÁ') and
  (datamodule2.Patient['job'] = '') then
  begin
    wuj:=wuj+ 1;
  end;
if datamodule2.Patient['sex'] = 'ΑΙΆÑÁÓ' then
  begin
    md:=md+ 1;
  end
else
  begin
    wd:=wd+ 1;
  end;
if (datamodule2.Patient['sex'] = 'ĂŒÍAÉÊÁ') and
  (datamodule2.Patient['job'] <> '') then
  begin
    wj:=wj+ 1;
  end;
  datamodule2.Visits.Last;
end;
  datamodule2.Visits.Next;
end;
  datamodule2.Patient.Next;
end;
label58.Caption:=md+ wd;
label36.Caption:=muj;
label35.Caption:=mj;
label33.Caption:=wuj;
label37.Caption:=wj;
```

```
Label134.Caption:='------';
label50.Caption:=md;
label49.Caption:=wd;
Label135.Caption:='------';
if (md+ wd<>0) then
  Begin
    sum:=int(md*100/(md+ wd));
    label106.Caption:=sum;
    label107.Caption:='%';
  end
else
  Begin
    label106.Caption:='0';
    label107.Caption:='%';
  end;
if (wd+ md<>0) then
  Begin
    sum:=int(wd*100/(md+ wd));
    label115.Caption:=sum;
    label116.Caption:='%';
  end
else
  Begin
    label115.Caption:='0';
    label116.Caption:='%';
  end;
if (muj+ mj<>0) then
  Begin
    sum:=int(muj*100/(muj+ mj));
    label108.Caption:=sum;
    label104.Caption:='%';
    sum:=int(mj*100/(muj+ mj));
    label110.Caption:=sum;
    label103.Caption:='%';
```

```
end
else
  Begin
    label108.Caption:='0';
    label104.Caption:='%';
    label110.Caption:='0';
    label103.Caption:='%';
  end;
if (wuj+ wj<>0) then
  Begin
    sum:=int(wuj*100/(wuj+ wj));
    label109.Caption:=sum;
    label114.Caption:='%';
    sum:=int(wj*100/(wuj+ wj));
    label112.Caption:=sum;
    label113.Caption:='%';
  end
else
  Begin
    label109.Caption:='0';
    label114.Caption:='%';
    label112.Caption:='0';
    label113.Caption:='%';
  end;
end;

procedure Tstatistiks.Button3Click(Sender: TObject);
begin
  edit1.Text:='';
  label36.Caption:='';
  label35.Caption:='';
  label33.Caption:='';
  label37.Caption:='';
  label50.Caption:='';
```

```
label49.Caption:="";  
label103.Caption:="";  
label104.Caption:="";  
Label108.Caption:="";  
Label109.Caption:="";  
Label110.Caption:="";  
Label112.Caption:="";  
Label113.Caption:="";  
Label114.Caption:="";  
Label106.Caption:="";  
Label107.Caption:="";  
Label115.Caption:="";  
Label116.Caption:="";  
Label121.Caption:="";  
Label123.Caption:="";  
Label124.Caption:="";  
Label122.Caption:="";  
Label134.Caption:="";  
Label135.Caption:="";  
Label58.Caption:="";  
end;  
end.
```



```
unit UnVisits;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ComCtrls, StdCtrls, Mask, DBCtrls;
```

```
type
```

```
tBOOKMARK=POINTER;
```

```
TVisits = class(TForm)
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
Label7: TLabel;
```

```
DBEdit2: TDBEdit;
```

```
DBRichEdit1: TDBRichEdit;
```

```
DBRichEdit2: TDBRichEdit;
```

```
DBRichEdit3: TDBRichEdit;
```

```
DBRichEdit4: TDBRichEdit;
```

```
Button1: TButton;
```

```
Label14: TLabel;
```

```
Edit1: TEdit;
```

```
Label1: TLabel;
```

```
Label19: TLabel;
```

```
Label111: TLabel;
```

```
Label22: TLabel;
```

```
Label23: TLabel;
```

```
Label80: TLabel;
```

```
Label27: TLabel;
```

```
procedure FormCreate(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Visits: TVisits;
  A:TBOOKMARK;

implementation

uses UnDataModule1, UnWelcome, UnInput, UnDetail;

{$R *.dfm}

procedure TVisits.FormCreate(Sender: TObject);
begin
  datamodule2.Visits.Close;
  datamodule2.Visits.Open;
  datamodule2.Visits.Insert;
  edit1.Text:=DateToStr(Date);
end;

procedure TVisits.Button1Click(Sender: TObject);
begin
  Visits.Close;
end;

procedure TVisits.FormClose(Sender: TObject; var Action: TCloseAction);
var a:integer;
begin
  datamodule2.Visits['v_date']:=edit1.Text;
  input.RadioButton1.Checked:=false;
  input.RadioButton2.Checked:=false;
```

```
input.Edit2.Text:="";
input.Edit3.Text:="";
input.Edit4.Text:="";
input.Edit5.Text:="";
input.button4.Enabled:=false;
input.button1.Enabled:=false;
input.label18.Enabled:=false;
input.label17.Enabled:=false;
datamodule2.Visits['symptoms']:=ansiuppercase(dbrichedit1.text);
datamodule2.Visits['tests']:=ansiuppercase(dbrichedit2.text);
datamodule2.Visits['diagnose']:=ansiuppercase(dbrichedit3.text);
datamodule2.Visits['medication']:=ansiuppercase(dbrichedit4.text);
datamodule2.Visits.Post;
INPUT.Show;
end;

end.
```

Βιβλιογραφία και Αναφορές

Για τη δημιουργία της εργασίας αυτής, προκειμένου να συλλέξουμε γνώσεις καθώς και να διατυπώσουμε κάποιους ορισμούς και έννοιες, αναφερθήκαμε στις παρακάτω πηγές στις οποίες μπορείτε να αναφερθείτε για περισσότερες πληροφορίες:

☛ Βιβλία

- ☛ [1] Οδηγός του SQL Server 2000
- ☛ [2] Εισαγωγή Στα Συστήματα Βάσεων Δεδομένων (C. J. Date, Κλειδάριθμος)
- ☛ [3] Πλήρης Οδηγός του Delphi 6 (Paul Kimmel, Β. Γκιούρδας)
- ☛ [4] Οδηγός του Delphi 3 (Frank Engo, Β. Γκιούρδας Εκδοτική)

☛ Help File Delphi [5]

☛ On Line: Delphi.about.com (Zarko Galic) [6]

☛ Άρθρα

- ☛ [7] "A Relational Model Of Data For Large Shared Data Banks" το 1970