



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μελέτη - Σχεδίαση - Ανάπτυξη μιας Desktop
Εφαρμογής για την Διαχείριση Πελατών ε-
νός Ιατρείου**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΟΥΣΟΥΛΗΣ ΘΩΜΑΣ

(ΑΕΜ: 1093)

Επιβλέπων:

Τασκασαπλίδης Γεώργιος

ΚΑΣΤΟΡΙΑ 06/2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μελέτη - Σχεδίαση - Ανάπτυξη μιας Desktop
Εφαρμογής για την Διαχείριση Πελατών ενός
Ιατρείου**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΟΥΣΟΥΛΗΣ ΘΩΜΑΣ
(ΑΕΜ: 1093)

Επιβλέπων:
Τασκασαπλίδης Γεώργιος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....

Καστοριά 06 - 2022

[Copyright © 2022 – ΜΟΥΣΟΥΛΗΣ ΘΩΜΑΣ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.]

Περίληψη

Η χρήση του ηλεκτρονικού υπολογιστή για την διεκπεραίωση καθημερινών απλών και σύνθετων διεργασιών είναι απολύτως αναγκαία. Ειδικότερα, σήμερα, που βρισκόμαστε στην εποχή της 4^{ης} βιομηχανικής επανάστασης με τα δεδομένα να έχουν κατακλύσει την καθημερινότητα μας σε όλους τους τομείς, η ύπαρξη βάσεων δεδομένων όπου τα δεδομένα αποθηκεύονται και οργανώνονται και η ανάπτυξη κατάλληλων εφαρμογών που τα διαχειρίζονται αποτελούν τα αναγκαία εργαλεία διαχείρισης δεδομένων για μια επιχείρηση ή έναν οργανισμό.

Γι' αυτό στην παρούσα πτυχιακή εργασία παρουσιάζω μια απλή εφαρμογή διαχείρισης ραντεβού ενός ιατρείου που έχω υλοποιήσει με την εφαρμογή Microsoft Access 2016. Συγκεκριμένα, περιγράφεται αναλυτικά ο σχεδιασμός της βάσης δεδομένων, οι φόρμες και οι λειτουργίες της εφαρμογής γραμμένες με την γλώσσα προγραμματισμού Visual Basic for Applications (VBA). Σκοπός της εργασίας είναι η ανάδειξη του πλεονεκτήματος της χρήσης της πλατφόρμας Microsoft Access για την ανάπτυξη μικρών, ευέλικτων και πρακτικών εφαρμογών που καλύπτουν τις ανάγκες μιας μικρής επιχείρησης, όπως στην προκείμενη περίπτωση ενός ιατρείου. Τέλος, παρουσιάζεται σύντομα μια μελλοντική επέκταση της εφαρμογής σε διαδικτυακό περιβάλλον μέσω σύγχρονων εργαλείων ανάπτυξης εφαρμογών (ASP.NET, SQL Server, κτλ).

Λέξεις κλειδιά: *Microsoft Access, SQL, VBA, reports, frontend, backend, βάση δεδομένων, δεδομένα, φόρμες, ΣΔΒΔ*

Abstract

The use of the computer to perform daily simple and complex processes is absolutely necessary. In particular, today, in the era of the 4th industrial revolution where data has flooded our daily lives in all areas, the existence of databases where data is stored and organized and the development of appropriate applications that manage them are the necessary management data tools for a small business or organization.

Hence, in this dissertation I present a simple appointment management application for a doctor's office which I have implemented with the Microsoft Access 2016 application. Specifically, it describes in detail the design of the database, the forms and functions of the application written in the programming language Visual Basic for Applications (VBA). The purpose of this work is to highlight the advantage of using the Microsoft Access platform to develop small, flexible and practical applications that meet the needs of a small business, as in the case of a doctor's office. Finally, a future extension of the application to the web environment through modern application development tools (ASP.NET, SQL Server, etc.) will be presented.

Keywords: *Microsoft Access, SQL, VBA, Reports, frontend, backend, database, data, forms, DBMS*

Περιεχόμενα

Περίληψη	v
Abstract.....	vi
Περιεχόμενα.....	vii
Λίστα Εικόνων – Διαγραμμάτων	xi
Λίστα Πινάκων	xv
Λεξικό Αγγλικών και Ελληνικών Όρων.....	xvii
Κεφάλαιο 1: Εισαγωγή.....	19
1.1 Γενικά	19
1.2 Στοχοθεσία.....	20
1.2.1 Τίτλος – Θέμα	20
1.2.2 Σκοπός και Στόχος	20
1.2.3 Άμεσοι στόχοι.....	21
1.3 Μελέτη Περίπτωσης	21
1.4 Δομή εργασίας	23
Κεφάλαιο 2: Θεωρητικό Πλαίσιο.....	25
2.1 Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ).....	25
2.1.1 Οι βασικές λειτουργίες ενός ΣΔΒΔ	26
2.1.2 Η αρχιτεκτονική ενός ΣΔΒΔ	27
2.1.3 Τα ΣΔΒΔ σήμερα.....	29
2.2 Η γλώσσα SQL	30
2.2.1 Κύρια Χαρακτηριστικά	30
2.2.2 Επεκτάσεις.....	32
2.2.3 Η επέκταση Jet SQL της Microsoft	32
2.3 Η Microsoft Access	33
2.3.1 Λειτουργίες.....	33
2.3.2 Περιβάλλον χρήσης.....	35

2.3.2 Πλεονεκτήματα – Μειονεκτήματα.....	38
2.3.3 Το πακέτο εκτέλεσης Microsoft Access Runtime	42
2.4 Η γλώσσα προγραμματισμού Visual Basic for Applications (VBA)	42
2.4.1 Χαρακτηριστικά.....	43
2.4.2 Τύποι δεδομένων	44
2.4.3 Διαδικασίες (Procedures).....	45
2.4.4 Συναρτήσεις (Functions).....	46
2.4.5 Το περιβάλλον ανάπτυξης VBA.....	47
Κεφάλαιο 3: Σχεδιασμός της Βάσης Δεδομένων	50
3.1 Βήματα υλοποίησης της ΒΔ	50
3.2 Καταγραφή Απαιτήσεων	51
3.3 Εννοιολογικός Σχεδιασμός.....	53
3.3.1 Εννοιολογικό Μοντέλο	53
3.3.2 Διάγραμμα Οντοτήτων – Συσχετίσεων	56
3.4 Λογικός Σχεδιασμός.....	57
3.4.1 Το Σχεσιακό Μοντέλο.....	57
3.4.2 Διάγραμμα Σχεσιακού Μοντέλου	59
3.5 Φυσικός Σχεδιασμός.....	62
3.5.1 Το αρχείο της Βάσης Δεδομένων	62
3.5.2 Οι Πίνακες της ΒΔ.....	64
Κεφάλαιο 4: Σχεδιασμός της εφαρμογής.....	68
4.1 Βασικές αρχές σχεδίασης ενός γραφικού περιβάλλοντος.....	68
4.1.1 Σχεδίαση φορμών και αναφορών με την MS Access.....	68
4.1.2 Δημιουργία του αρχείου front-end - Σύνδεση με ΒΔ.....	70
4.1.3 Διάγραμμα των διαλόγων/λειτουργιών της εφαρμογής.....	73
4.2 Τα Ερωτήματα.....	74
4.2.1 Το ερώτημα «qryTodayAppointments»	74
4.2.2 Το ερώτημα «qryNextDayAppointments»	75

4.2.3 Το ερώτημα «qryAppointments»	75
4.2.4 Το ερώτημα «qryPatientGender»	76
4.2.5 Το ερώτημα «qryAgeGroups»	76
4.3 Οι Φόρμες	77
4.3.1 Η φόρμα «frmLogin» - Σύνδεση Χρήστη	79
4.4.2 Η φόρμα «frmMain» - Κεντρική Φόρμα Επιλογών	81
4.4.3 Η φόρμα «frmPatientArpts» - Επισκέψεις Ασθενών.....	84
4.4.4 Η φόρμα «frmAddPatient» - Νέος Ασθενής.....	90
4.4.5 Η φόρμα «frmEditPatient» - Επεξεργασία Ασθενή.....	92
4.4.6 Η φόρμα «frmAddAppointment» - Νέο Ραντεβού	94
4.4.7 Η φόρμα «frmEditAppointment» - Επεξεργασία Ραντεβού	97
4.4.8 Η φόρμα «frmSubPatient» - Προβολή Ασθενών (Κύρια).....	99
4.4.9 Η φόρμα «frmSubAppointment» - Προβολή Επισκέψεων (Θυγατρική)	100
4.4.10 Η φόρμα «sfmTodayArpts» - Σημερινά Ραντεβού	100
4.4.11 Η φόρμα «sfmNextdayArpts» - Επόμενα Ραντεβού.....	101
4.5 Οι Εκθέσεις (Αναφορές)	102
4.5.1 Η έκθεση «rptTodayArpts» - Σημερινά Ραντεβού.....	102
4.5.2 Η έκθεση «rptComingArpts» - Επόμενα Ραντεβού	103
4.5.3 Η έκθεση «rptPatientGender» - Ασθενείς ανά Φύλο	103
4.5.4 Η έκθεση «rptPatientAgeGroups» - Ηλικιακές ομάδες ασθενών	104
4.6 Οι Λειτουργικές Μονάδες	105
4.6.1 Η μονάδα «mdlPatient»	105
4.6.2 Η μονάδα «mdlAppointment»	106
4.6.3 Η μονάδα «gblVariables»	107
4.7 Παραγωγή του εκτελέσιμου αρχείου	107
Κεφάλαιο 5: Τρόπος λειτουργίας της εφαρμογής	112
5.1 Σύνδεση στην εφαρμογή	112
5.2 Καταχώρηση ασθενή	115

5.3 Καταχώρηση νέου ραντεβού	117
5.4 Αλλαγή στοιχείων ασθενή - επίσκεψης	119
5.4 Εκτυπώσεις εκθέσεων	123
5.5 Πρόσθετες Λειτουργίες	125
5.5.1 Προσθήκη Ημερών Εργασίας	125
5.5.2 Διαχείριση Χρηστών	128
Κεφάλαιο 6: Συμπεράσματα – Επεκτάσεις	132
Βιβλιογραφία	136
Παράρτημα	138
A. Κώδικας των φορμών	138
Φόρμα «frmAddAppointment».....	138
Φόρμα «frmAddPatient»	140
Φόρμα «frmEditAppointment».....	143
Φόρμα «frmEditPatient».....	145
Φόρμα «frmLogin»	148
Φόρμα «frmMain»	150
Φόρμα «frmPatientAppts».....	152
Φόρμα «frmSubAppointment»	155
Φόρμα «frmSubPatient»	155
Φόρμα «frmUserManager».....	156
Φόρμα «frmWorkdays»	159
B. Κώδικας των λειτουργικών μονάδων.....	160
Λειτουργική μονάδα «mdlAppt».....	160
Λειτουργική μονάδα «mdlPatient».....	161

Λίστα Εικόνων – Διαγραμμάτων

Εικόνα 1: Διάγραμμα της ροής εκτέλεσης των κύριων λειτουργιών της εφαρμογής.....	22
Εικόνα 2: Τα δομικά στοιχεία ενός ΣΔΒΔ.....	25
Εικόνα 3: Διαγραμματική αναπαράσταση του ρόλου ενός ΣΔΒΔ.....	26
Εικόνα 4: Δείγμα του περιβάλλοντος χρήσης της MS Access.....	35
Εικόνα 5: Η κορδέλα των καρτέλων. Διακρίνονται οι κύριες καρτέλες και η πρόσθετες καρτέλες που «Σχεδίαση φόρμας», «Τακτοποίηση» και «Μορφοποίηση».....	36
Εικόνα 6: Ο Πίνακας Αντικειμένων της Access.....	36
Εικόνα 7: Το Φύλλο Ιδιοτήτων ενός αντικειμένου της MS Access.....	37
Εικόνα 8: Η περιοχή των αντικειμένων επιτρέπει στο χρήστη να ανοίγει περισσότερα από ένα αντικείμενα της Access, είτε για προβολή είτε για επεξεργασία. Οι λειτουργικές μονάδες ανοίγουν μόνο στον Επεξεργαστή VBA.....	37
Εικόνα 9: Το IDE της VBA όπως είναι ενσωματωμένο στην MS Access από την έκδοση 2010 και μετά. Διακρίνεται το παράθυρο του κώδικα, το παράθυρο άμεσων μηνυμάτων και το παράθυρο με τη λίστα των αντικειμένων.....	48
Εικόνα 10: Το διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣ) της βάσης δεδομένων.....	56
Εικόνα 11: Το διάγραμμα του Σχισιακού Μοντέλου της ΒΔ, σχεδιασμένο στο λογισμικό Microsoft Visio Professional 2019.....	60
Εικόνα 12: Το παράθυρο διαλόγου «Επεξεργασία Σχέσεων».....	61
Εικόνα 13: Το διάγραμμα του Σχισιακού Μοντέλου όπως προκύπτει μέσα από το περιβάλλον της MS Access.....	61
Εικόνα 14: Ο φάκελος της εφαρμογής με τους καταλόγους DATA και APP που περιέχουν τα αρχεία της εφαρμογής.....	63
Εικόνα 15: Το εικονίδιο-αρχείο της εφαρμογής της ΒΔ.....	63
Εικόνα 16: Το αρχείο back-end που περιέχει τους πίνακες της ΒΔ.....	64
Εικόνα 17: Το μενού επιλογών της MS Access 2019 για τον σχεδιασμό φορμών. Διακρίνονται όλες οι διαθέσιμες επιλογές.....	69
Εικόνα 18: Το αρχείο της εφαρμογής (front-end), medApp.accdb, στο φάκελο APP.....	70
Εικόνα 19: Το μενού "Νέα Προέλευση δεδομένων" από όπου επιλέγουμε την Access ως πηγή των εξωτερικών δεδομένων.....	71
Εικόνα 20: Η εισαγωγή του αρχείου της βάσης δεδομένων.....	71
Εικόνα 21: Οι πίνακες της βάσης δεδομένων θα συνδεθούν μέσω του οδηγού στο νέο αρχείο.....	72

Εικόνα 22: Οι πίνακες του αρχείου της βάσης (back-end) έχουν συνδεθεί στο αρχείο εφαρμογής (front-end).....	72
Εικόνα 23: Το διάγραμμα διαλόγων της εφαρμογής. Περιέχει τις βασικές λειτουργίες της εφαρμογής που είναι διαθέσιμες στο χρήστη.....	73
Εικόνα 24: Η φόρμα «Σύνδεση Χρήστη» (frmLogin)	79
Εικόνα 25: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ραντεβού».....	81
Εικόνα 26: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ασθενείς».....	82
Εικόνα 27: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Εκτυπώσεις»	83
Εικόνα 28: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ρυθμίσεις»	83
Εικόνα 29: Η φόρμα «Ασθενείς» (frmPatient). Διακρίνονται στο πάνω μέρος τα στοιχεία του ασθενή και στο κάτω μέρος τα ραντεβού του ασθενή.....	84
Εικόνα 30: Η φόρμα «Νέος Ασθενής» (frmAddPatient).....	90
Εικόνα 31: Η φόρμα «Επεξ/σία στοιχείων ασθενή» (frmEditPatient)	92
Εικόνα 32: Η φόρμα «Νέο Ραντεβού» (frmAddAppointment).....	94
Εικόνα 33: Οι τιμές του πεδίου «Ωρα» με την ένδειξη εάν είναι διαθέσιμη ή όχι στον ασθενή.....	95
Εικόνα 34: Η φόρμα «Επεξ/σία Ραντεβού» (frmEditAppointment).....	97
Εικόνα 35: Η υπο-φόρμα «Προβολή Ασθενών» (frmSubPatient)	99
Εικόνα 36: Η υπο-φόρμα «Προβολή Ραντεβού Ασθενών» (frmSubAppointment).....	100
Εικόνα 37: Η υπο-φόρμα «Σημερινά Ραντεβού» (sfmTodayArpts).....	100
Εικόνα 38: Η υπο-φόρμα «Επόμενα Ραντεβού» (sfmNextdayArpts)	101
Εικόνα 39: Η έκθεση «Σημερινά Ραντεβού» (rptTodayArpts)	102
Εικόνα 40: Η αναφορά «Επόμενα Ραντεβού» (rptNextdayArpt).....	103
Εικόνα 41: Η αναφορά «Ασθενείς ανά Φύλο» (rptPatientperGender).....	103
Εικόνα 42: Η αναφορά «Ηλιακές ομάδες ασθενών» (rptPatientAgeGroups).....	104
Εικόνα 43: Οι ρυθμίσεις επιλογών ενός αρχείου MS Access	108
Εικόνα 44: Οι ρυθμίσεις επιλογών για τις ενότητες «Περιήγηση» και «Επιλογές γραμμής εργαλείων και κορδέλας»	109
Εικόνα 45: Η δημιουργία του εκτελέσιμου αρχείου γίνεται μέσω της επιλογής «Αποθήκευση Ως».....	109

Εικόνα 46: Η φόρμα Σύνδεση Χρήστη είναι η πρώτη φόρμα που εμφανίζεται κατά την εκτέλεση της εφαρμογής.	112
Εικόνα 47: Μήνυμα προειδοποίησης για κενή τιμή στο πεδίο «Χρήστης»	112
Εικόνα 48: Μήνυμα προειδοποίησης για κενή τιμή στο πεδίο «Κωδ. Πρόσβασης»	113
Εικόνα 49: Μήνυμα ελέγχου ταυτότητας χρήστη (username) στη ΒΔ.....	113
Εικόνα 50: Μήνυμα προειδοποίησης λάθους κωδικού πρόσβασης.	113
Εικόνα 51: Η Αρχική οθόνη επιλογών.....	114
Εικόνα 52: Η καρτέλα «Ασθενείς» της αρχικής φόρμας.	115
Εικόνα 53: Η φόρμα καταχώρησης ενός νέου ασθενή στη ΒΔ.....	115
Εικόνα 54: Μήνυμα προειδοποίησης για την ορθότητα του ΑΜΚΑ.....	116
Εικόνα 55: Μήνυμα επιτυχούς καταχώρησης του ασθενή στη ΒΔ	116
Εικόνα 56: Μήνυμα σφάλματος για μη καταχώρηση ΑΜΚΑ στο αντίστοιχο πεδίο	116
Εικόνα 57: Η φόρμα Επισκέψεις (Ραντεβού) Ασθενών	117
Εικόνα 58: Μήνυμα προειδοποίησης πως δεν βρέθηκε ασθενής με το ΑΜΚΑ που δόθηκε στο πεδίο.....	117
Εικόνα 59: Η φόρμα καταχώρησης ενός νέου ραντεβού ασθενή.....	118
Εικόνα 60: Μήνυμα σφάλματος για κενές τιμές στα πεδία Ημερομηνία και Ώρα	118
Εικόνα 61: Μήνυμα σφάλματος για δεσμευμένο ραντεβού	119
Εικόνα 62: Μήνυμα ενημέρωσης για επιτυχή καταχώρηση του ραντεβού ασθενή	119
Εικόνα 63: Η φόρμα «Επεξ/σία στοιχείων ασθενή» όπως εμφανίζεται μετά το πάτημα του κουμπιού	120
Εικόνα 64: Μήνυμα επιβεβαίωσης αλλαγών στον ασθενή	120
Εικόνα 65: Μήνυμα ενημέρωσης αλλαγών στον ασθενή	120
Εικόνα 66: Η φόρμα «Επεξ/σία Ραντεβού» όταν καλείται από τον χρήστη	121
Εικόνα 67: Μήνυμα επιβεβαίωσης αλλαγών στο ραντεβού του ασθενή.....	121
Εικόνα 68: Μήνυμα ενημέρωσης αλλαγών στο ραντεβού του ασθενή.....	122
Εικόνα 69: Μήνυμα επιβεβαίωσης διαγραφής του ραντεβού ενός ασθενή.....	122
Εικόνα 70: Μήνυμα ενημέρωσης για τη διαγραφή του ραντεβού ενός ασθενή.....	122
Εικόνα 71: Η έκθεση «Ημερήσιες επισκέψεις».....	123
Εικόνα 72: Η έκθεση «Ποσοστά Ασθενών ανά Φύλο».....	124
Εικόνα 73: Η έκθεση «Ηλικιακές ομάδες».....	124
Εικόνα 74: Η φόρμα «Ημέρες εργασίας»	126
Εικόνα 75: Μήνυμα προειδοποίησης για ήδη καταχωρημένες ημέρες του έτους που αναγράφεται στο πεδίο «Τρέχον έτος»	126

Εικόνα 76: Μήνυμα ενημέρωσης για επιτυχή καταχώρηση των ημερών στη ΒΔ	126
Εικόνα 77: Η φόρμα «Διαχείριση Χρηστών»	128
Εικόνα 78: Μήνυμα σφάλματος για μη συμπληρωμένα πεδία της φόρμας	128
Εικόνα 79: Μήνυμα προειδοποίησης για ύπαρξη ονόματος χρήστη στη ΒΔ	129
Εικόνα 80: Μήνυμα σφάλματος για το μέγεθος του κωδικού πρόσβασης του χρήστη	129
Εικόνα 81: Μήνυμα ελέγχου λάθους κωδικού πρόσβασης κατά την επιβεβαίωση της τιμής του στο αντίστοιχο πεδίο	129
Εικόνα 82: Μήνυμα ενημέρωσης για την επιτυχή καταχώρηση του χρήστη στη ΒΔ	130

Λίστα Πινάκων

Πίνακας 1: Τα δέκα πιο δημοφιλή ΣΔΒΔ στον κόσμο (Απρίλιος 2022). Πηγή: https://db-engines.com/en/ranking	29
Πίνακας 2: Δημοφιλείς επεκτάσεις της γλώσσας SQL. Πηγή: https://en.wikipedia.org/wiki/SQL#Procedural_extensions	32
Πίνακας 3: Οι υποστηριζόμενοι τύποι δεδομένων της VBA.....	44
Πίνακας 4: Οι οντότητες με τους αντίστοιχους πίνακες της ΒΔ.....	53
Πίνακας 5: Τα χαρακτηριστικά της οντότητας «Ασθενής».....	53
Πίνακας 6: Τα χαρακτηριστικά της οντότητας «Ραντεβού».....	54
Πίνακας 7: Το χαρακτηριστικό της οντότητας «Ημέρες Εργασίας».....	54
Πίνακας 8: Το χαρακτηριστικό της οντότητας «Ώρες Λειτουργίας».....	55
Πίνακας 9: Τα χαρακτηριστικά της οντότητας «Χρήστης».....	55
Πίνακας 10: Τα χαρακτηριστικά της οντότητας «Τύπος Χρήστη».....	55
Πίνακας 11: Ο πίνακας-οντότητα «Patient».....	57
Πίνακας 12: Ο πίνακας-οντότητα «Appointment».....	58
Πίνακας 13: Ο πίνακας-οντότητα «AppDays».....	58
Πίνακας 14: Ο πίνακας-οντότητα «AppHours».....	58
Πίνακας 15: Ο πίνακας-οντότητα «AppHours» που αποτελεί σχεσιακή οντότητα	58
Πίνακας 16: Ο πίνακας-οντότητα «Users».....	59
Πίνακας 17: Ο πίνακας-οντότητα «UserTypes».....	59
Πίνακας 18: Τα πεδία του πίνακα «Patient».....	64
Πίνακας 19: Τα πεδία του πίνακα «Appointment».....	65
Πίνακας 20: Τα πεδία του πίνακα «AppDays»	65
Πίνακας 21: Τα πεδία του πίνακα «AppHours».....	65
Πίνακας 22: Τα πεδία του πίνακα «AppDetails».....	65
Πίνακας 23: Τα πεδία του πίνακα «Users».....	66
Πίνακας 24: Τα πεδία του πίνακα «UserTypes».....	66
Πίνακας 25: Τα στοιχεία ελέγχου της φόρμας «Σύνδεση Χρήστη»	79
Πίνακας 26: Τα στοιχεία ελέγχου της φόρμας «Αρχική»	84
Πίνακας 27: Τα κουμπιά των βασικών λειτουργιών επί των ασθενών και των ραντεβού	85
Πίνακας 28: Τα στοιχεία ελέγχου της φόρμας «Ραντεβού Ασθενή»	86
Πίνακας 29: Οι απαραίτητες τιμές των ιδιοτήτων της θυγατρικής φόρμας frmSubAppointment.....	88
Πίνακας 30: Τα στοιχεία ελέγχου της φόρμας «Νέος Ασθενής»	91

Πίνακας 31: Τα στοιχεία ελέγχου της φόρμας «Νέο Ραντεβού»	95
Πίνακας 32: Οι ρυθμίσεις των πεδίων «Επιλογές εφαρμογής»	108
Πίνακας 33: Οι ρυθμίσεις που αφορούν την Περιήγηση και τις επιλογές των γραμμών εργαλείων.....	108

Λεξικό Αγγλικών και Ελληνικών Όρων

ACDE	Access Database Engine
ADO	ActiveX Data Object
API	Application Programming Interface
DAO	Data Access Object
DBA	Database Administrator
DBMS	Database Management System
DDL	Data Definition Language
DML	Data Manipulation Language
ER	Entity-Relationship model
ERD	Entity-Relationship Diagram
GUI	Graphical User Interface
IDE	Integrated Development Environment
ODBC	Open Database Connectivity
RDBMS	Relational Database Management System
SQL	Structured Query Language
UDF	User-defined Function
UML	Unified Modelling Language
VB	Visual Basic
VBA	Visual Basic for Applications
ΒΔ	Βάση Δεδομένων
ΛΣ	Λειτουργικό Σύστημα
ΠΣ	Πληροφοριακά Συστήματα
ΣΔΒΔ	Σύστημα Διαχείρισης Βάσεων Δεδομένων
ΣΣΔΒΔ	Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων

1 Εισαγωγή

Κεφάλαιο 1: Εισαγωγή

1.1 Γενικά

Με την αυγή του 21^{ου} αιώνα ο κόσμος μπήκε στην εποχή της Πληροφορίας με τους ανθρώπους να έχουν την δυνατότητα να δημιουργούν, να ανταλλάσσουν και να μεταφέρουν πληροφορίες γρήγορα έχοντας παράλληλα πρόσβαση σε γνώσεις που στο παρελθόν ήταν πάρα πολύ δύσκολο. Ο κάθε άνθρωπος πλέον μπορεί και διαχειρίζεται πλήθος από πληροφορίες στην καθημερινότητα του βοηθώντας τον να αντιμετωπίζει τα καθημερινά προβλήματα του, λαμβάνοντας αποφάσεις που τον οδηγούν σε λύσεις.

Αυτός είναι και ο κύριος λόγος που η ανάθεση των σύνθετων προβλημάτων γίνεται με την βοήθεια των προγραμμάτων-εφαρμογών που διαχειρίζονται, επεξεργάζονται και αναλύουν τα δεδομένα. Ειδικότερα, οι μεγάλες ποσότητες δεδομένων που διακινούνται από μια μικρή επιχείρηση ως έναν μεγάλο οργανισμό, επιβάλλουν την εύρεση αποτελεσματικών μεθόδων αποθήκευσης. Αυτά με την σειρά τους πρέπει να είναι αποθηκευμένα και οργανωμένα έτσι ώστε να διευκολύνεται η διαχείριση τους καθώς ο μεγάλος όγκος τους επιβάλλουν αποτελεσματικές λύσεις αποθήκευσης. Γι' αυτό το λόγο αναπτύχθηκαν οι βάσεις δεδομένων και τα συστήματα διαχείρισης τους.

Μια βάση δεδομένων αποτελεί τον κυριότερο και σημαντικότερο παράγοντα της σωστής λειτουργίας και οργάνωσης μιας εταιρίας και ενός οργανισμού, καθώς συλλέγει, αποθηκεύει, οργανώνει και επεξεργάζεται τα δεδομένα ώστε να παραχθούν τα κατάλληλα αποτελέσματα σχετικά με κάποιο ζήτημα. Έτσι, αναπτύχθηκαν συστήματα όπου τα δεδομένα αποθηκεύονται και επεξεργάζονται σε κατάλληλα ψηφιακά μέσα με τη χρήση κατάλληλων εφαρμογών λογισμικού. Οι εφαρμογές αυτές, εκτός από τις βασικές λειτουργίες της δημιουργίας και της αποθήκευσης των δεδομένων, παρέχουν στον χρήστη τη δυνατότητα να επεξεργάζεται και να αναλύει αυτά τα δεδομένα προκειμένου να λάβει κάποιες αποφάσεις σχετικά με τα προβλήματα που αντιμετωπίζει.

Είτε πρόκειται για ένα εμπορικό κατάστημα, είτε για μια βιομηχανία, είτε για έναν οργανισμό ή εταιρεία, στο εμπόριο του λογισμικού υπάρχουν δεκάδες τέτοιες εφαρμογές που διαχειρίζονται δεδομένα και προσαρμόζονται ανάλογα με τις ανάγκες και τις απαιτήσεις της κάθε εταιρίας ή οργανισμού. Μάλιστα, η πρόσφατη πανδημία που έπληξε τον κόσμο ενεργοποίησε πολύ το ενδιαφέρον των υπεύθυνων των εταιριών να ενσωματώσουν στην επιχείρησή τους νέες ψηφιακές λειτουργίες και λύσεις που προσφέρουν αυτές οι εφαρμογές. Ωστόσο, τις περισσότερες φορές το κόστος των εφαρμογών αυτών είναι μερικές φορές α-

παγορευτικό για μια επιχείρηση, ενώ ορισμένες φορές τα εργαλεία που προσφέρουν στο χρήστη δεν επαρκούν για όλες τις λειτουργίες που διατελεί.

Η περίπτωση περίπτωσης που θα εξετάσουμε με αυτήν την εργασία, αφορά τον σχεδιασμό και την υλοποίηση μιας πρακτικής και οικονομικής εφαρμογής διαχείρισης ραντεβού σε ένα ιατρείο με ένα από τα παλαιότερα συστήματα διαχείρισης δεδομένων, την Microsoft Access. Ειδικότερα, εξετάζουμε την απλή περίπτωση κράτησης ραντεβού ενός ασθενούς σε ένα οποιοδήποτε ιατρείο για μια εξέταση ενώ παράλληλα θα αποθηκεύονται στοιχεία της επισκέψεως όπως η αιτία της επίσκεψης του ασθενούς, η πάθηση με την οποία διαγνώστηκε, το φύλο και η ηλικία του ασθενούς, έτσι ώστε ο χρήστης της εφαρμογής να μπορεί ανά πάσα στιγμή να αναζητήσει χρήσιμες πληροφορίες του ασθενούς σε κάθε επίσκεψή του στο ιατρείο.

1.2 Στοχοθεσία

1.2.1 Τίτλος – Θέμα

Ο τίτλος της πτυχιακής εργασίας είναι «*Μελέτη - Σχεδίαση - Ανάπτυξη μιας Desktop Εφαρμογής για την Διαχείριση Πελατών ενός Ιατρείου*».

Το **θέμα** της εργασίας αφορά την υλοποίηση μιας εφαρμογής γραφείου (desktop application) που καταγράφει και διαχειρίζεται τα ραντεβού ενός ιατρείου παρέχοντας παράλληλα χρήσιμα στατιστικά στοιχεία που αφορούν τους πελάτες του ιατρείου, μέσα από ένα ευέλικτο, πρακτικό και φιλικό προς τον χρήστη περιβάλλον εργασίας, όπως αυτό της Microsoft Access.

1.2.2 Σκοπός και Στόχος

Ο **σκοπός** της εργασίας είναι η ανάδειξη των πλεονεκτημάτων της MS Access σε συνδυασμό με την γλώσσα προγραμματισμού Visual Basic for Applications (VBA), που αφορούν το σχεδιασμό και την ανάπτυξη μιας μικρής και ευέλικτης εφαρμογής επιτραπέζιας εφαρμογής για της ανάγκες μιας μικρής επιχείρησης ή ενός οργανισμού, χωρίς το μεγάλος κόστος που τις περισσότερες φορές είναι απαγορευτικό για πολλούς επαγγελματίες.

Ο κύριος **στόχος** της εργασίας είναι η καταχώρηση και η διαχείριση των ημερήσιων ραντεβού ενός ιατρείου, μέσω μιας ευέλικτης και πρακτικής εφαρμογής όπου θα καταχωρούνται τα στοιχεία των ασθενών. Μέσω της εφαρμογής ο χρήστης (ο ιατρός ή ο γραμματέας) θα μπορεί να ελέγχει την ημερήσια κίνηση του ιατρείου, εξυπηρετώντας εύκολα και άμεσα

τους ασθενείς και μέσω κατάλληλων αναφορών (εκθέσεων) να λαμβάνει αποφάσεις σχετικά με αυτούς, όπως για παράδειγμα τα ποσοστά των ηλικιακών ομάδων των ασθενών.

1.2.3 Άμεσοι στόχοι

Όπως αναφερθήκαμε προηγουμένως, ο κύριος στόχος της εργασίας είναι η δημιουργία μιας εύχρηστης εφαρμογής για την διαχείριση των ραντεβού ενός ιατρείου. Ωστόσο, σε γενικό πλαίσιο, οι άμεσοι στόχοι της πτυχιακής εργασίας συνοψίζονται παρακάτω:

- α. Δημιουργία της βάσης δεδομένων με τους κατάλληλους πίνακες και τις συσχετίσεις που έχουν μεταξύ τους, στους οποίους καταχωρούνται οι βασικές πληροφορίες της εφαρμογής.
- β. Δημιουργία των φορμών της εφαρμογής που θα επιτρέπουν την καταχώρηση, την ενημέρωση και την αναζήτηση των στοιχείων των ασθενών.
- γ. Σύνταξη απαραίτητων ερωτημάτων που διευκολύνουν τη σωστή διαχείριση των δεδομένων της ΒΔ.
- δ. Δυνατότητα εκτύπωσης των ημερησίων ραντεβού και του ιστορικού των επισκέψεων του ασθενή.
- ε. Δυνατότητα προβολής στατιστικών αναφορών που σχετίζονται με τις επισκέψεις των ασθενών.

1.3 Μελέτη Περίπτωσης

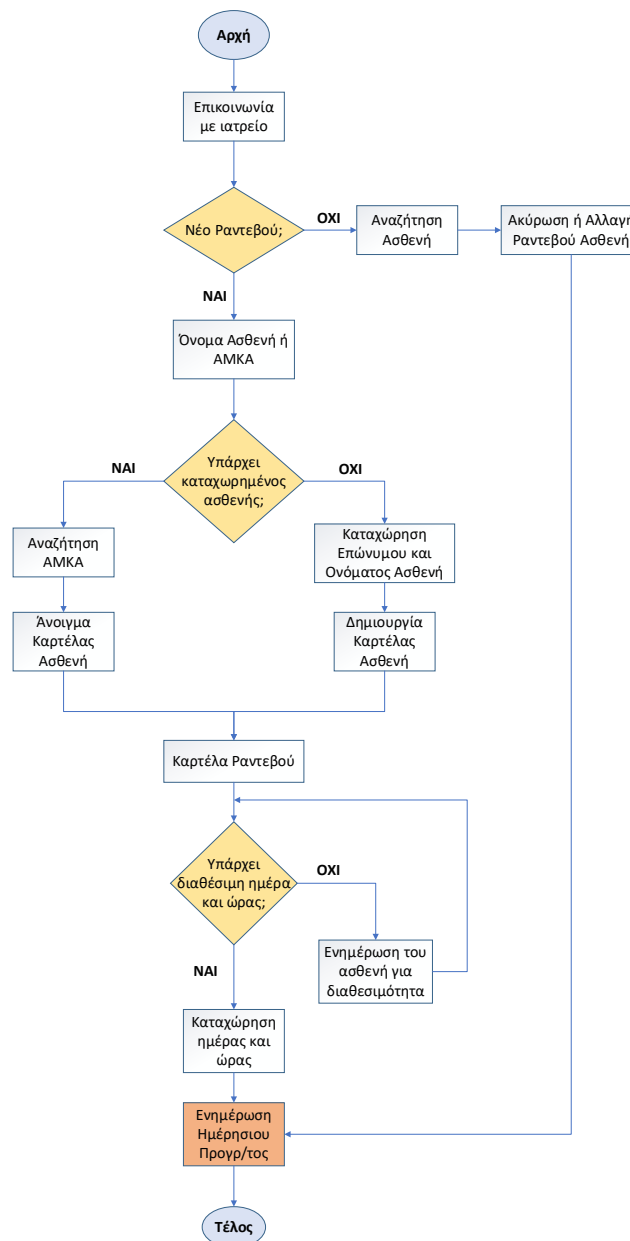
Η διαδικασία καταχώρησης ενός ραντεβού από τον χρήστη (γιατρό) περιγράφεται από τα παρακάτω βήματα:

1. Ο ασθενής επικοινωνεί με τον γιατρό και ζητά μια επίσκεψη στο ιατρείο του. Ο γιατρός ελέγχει το ημερήσιο πρόγραμμα του για διαθέσιμη ώρα επίσκεψης και ενημερώνει τον ασθενή. Εάν ο ασθενής συμφωνεί για την ημέρα και την ώρα, ο γιατρός προχωράει στην κράτηση. Διαφορετικά ενημερώνει τον ασθενή για διαθεσιμότητα σε διαφορετική ημέρα και ώρα.
2. Εάν πρόκειται για παλιό ασθενή, ο γιατρός αναζητεί τον ασθενή με το επίθετο του ή το ΑΜΚΑ του και στην συνέχεια καταχωρεί την ημέρα, την ώρα επισκέψεως και τον σκοπό της επισκέψεως του. Σε περίπτωση που ο ασθενής επισκέπτεται πρώτη φορά

το ιατρείο, ο γιατρός καταχωρεί την ημέρα, την ώρα, τον σκοπό της επισκέψεως και το ονοματεπώνυμο του ασθενή.

3. Την ημέρα της προσέλευσης στο ιατρείο ο γιατρός καταχωρεί και τα υπόλοιπα στοιχεία του ασθενή, όπως το ΑΜΚΑ, η ημερομηνία γέννησης και τα στοιχεία επικοινωνίας του.
4. Σε περίπτωση ακύρωσης της επισκέψεως στο ιατρείο, ο γιατρός ανοίγει την καρτέλα των ραντεβού ή την καρτέλα του ασθενή και ακυρώνει το ραντεβού, αφήνοντας την ώρα του ραντεβού διαθέσιμη για άλλον ασθενή.

Τα παραπάνω βήματα μπορούν να αναπαρασταθούν διαγραμματικά:



Εικόνα 1: Διάγραμμα της ροής εκτέλεσης των κύριων λειτουργιών της εφαρμογής.

1.4 Δομή εργασίας

Στο επόμενο «**Κεφάλαιο 2: Θεωρητικό Πλαίσιο**» καλύπτονται οι βασικές έννοιες που σχετίζονται με τις βάσεις δεδομένων και τα σύγχρονα συστήματα διαχείρισης βάσεων δεδομένων, την σχεσιακή γλώσσα δεδομένων SQL, το περιβάλλον της MS Access και τη γλώσσα προγραμματισμού Visual Basic for Applications (VBA).

Στο «**Κεφάλαιο 3: Σχεδιασμός της βάσης δεδομένων**» δίνεται αναλυτική περιγραφή στην ανάλυση και τον σχεδιασμό της βάσης δεδομένων δηλαδή, στις οντότητες, στους πίνακες, στους τύπους δεδομένων που καταχωρούνται στους πίνακες, στις σχέσεις μεταξύ των πινάκων και στο διάγραμμα οντοτήτων-συσχετίσεων (Entity-Relationship Diagram - ER).

Στο «**Κεφάλαιο 4: Σχεδιασμός της εφαρμογής**» περιγράφονται οι φόρμες και οι εκτυπώσεις (αναφορές) της εφαρμογής με πλήρη περιγραφή των βασικών λειτουργιών τους (κουμπιά, πεδία, κτλ.)

Στο «**Κεφάλαιο 5: Τρόπος λειτουργίας της εφαρμογής**» περιγράφονται τα βήματα εκτέλεσης των βασικών λειτουργιών της εφαρμογής, όπως η καταχώρηση ενός ασθενή, η δημιουργία ενός νέου ραντεβού, κτλ.

Στο «**Κεφάλαιο 6: Συμπεράσματα - Επεκτάσεις**» καταγράφονται τα συμπεράσματα που προέκυψαν από την υλοποίηση της εφαρμογής και οι μελλοντικές προσθήκες και επεκτάσεις της. Συγκεκριμένα, γίνεται μια σύντομη αναφορά στον τρόπο με τον οποίο μπορεί να υλοποιηθεί η εφαρμογή για φορητές συσκευές (κινητά, tablet) μέσω της γλώσσας προγραμματισμού ASP.Net και της χρήσης του ΣΔΒΔ SQL-Server.

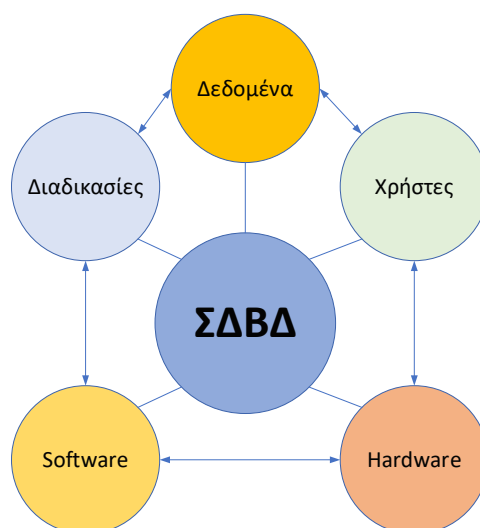
Τέλος, στο κεφάλαιο «**Παραρτήματα**» υπάρχει ο κώδικας της εφαρμογής των αντικειμένων (φορμών) και των διαδικασιών που χρησιμοποιούνται.

2 Θεωρητικό πλαίσιο

Κεφάλαιο 2: Θεωρητικό Πλαίσιο

2.1 Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ)

Τα πρώτα Συστήματα Διαχείρισης Βάσεων Δεδομένων (Database Management Systems – DBMS) γεννήθηκαν στην δεκαετία του 1960, όταν άρχισε να δημιουργείται η ανάγκη για την ανεξάρτητη ενημέρωση των αρχείων που χρησιμοποιούσαν οι χρήστες των υπολογιστών μέσω των εφαρμογών λογισμικού εκείνης της εποχής, από ένα ξεχωριστό σύστημα προγραμμάτων. Ειδικότερα, έπρεπε να βρεθεί μια λύση έτσι ώστε τα δεδομένα των εφαρμογών να αποθηκεύονται κάπου ανεξάρτητα από τις εφαρμογές και να υπάρχουν δικλίδες ασφαλείας στην προσπέλαση και τον χειρισμό αυτών.



Εικόνα 2: Τα δομικά στοιχεία ενός ΣΔΒΔ.

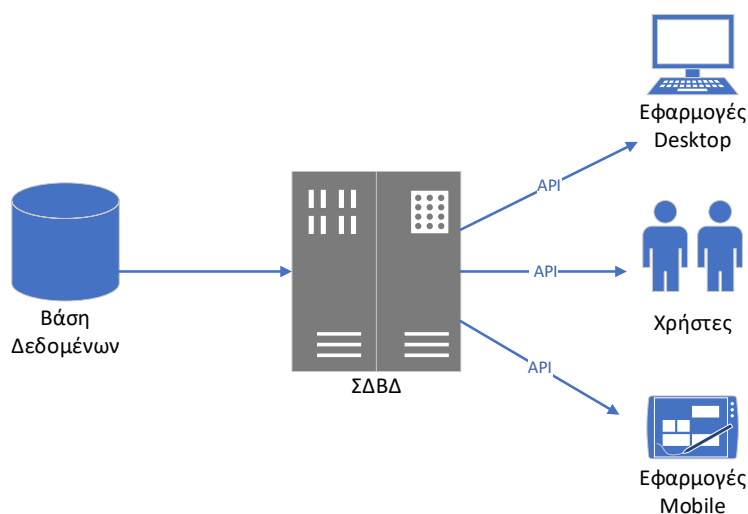
Γενικά, ένα ΣΔΒΔ αποτελείται από πέντε βασικούς πυλώνες, τα **δεδομένα** (data), το **υλικό** (hardware), το **λογισμικό** (software), τις **διαδικασίες** (procedures) και τους **χρήστες** (users), οι οποίοι πρέπει να παρουσιάζουν μια συμμετρία. Συγκεκριμένα:

- α. Οι «χρήστες» αλληλοεπιδρούν με το «υλικό» (υπολογιστές) για την χρήση του ΣΔΒΔ.
- β. Το «λογισμικό» και οι «διαδικασίες» είναι στην ουσία οι εντολές προς το ΣΔΒΔ. Ειδικά, το «λογισμικό» (εφαρμογές) είναι οι εντολές που δίνονται προς το «υλικό», ενώ οι «διαδικασίες» είναι οι εντολές που παράγονται από τους χρήστες μέσω των δεδομένων.
- γ. Τα δεδομένα είναι ο κορυφαίος πυλώνας καθώς αποτελούν την σύνδεση μεταξύ των χρηστών (δημιουργία, αποθήκευση) και των ηλεκτρονικών υπολογιστών (ανάλυση, επεξεργασία, αποτελέσματα).

Επομένως, ένα ΣΔΒΔ είναι στην πραγματικότητα ένα ενιαίο σύνολο εφαρμογών και διαδικασιών που ο κύριος σκοπός του είναι η δημιουργία, η επεξεργασία, η συντήρηση και η ασφάλεια μιας βάσης δεδομένων. Ταυτόχρονα, παρέχει πληροφορίες στους χρήστες μέσω κατάλληλου λογισμικού χωρίς απαραίτητα να «φανερώνεται» στο χρήστη ο χώρος και ο τρόπος αποθήκευσης των δεδομένων.

2.1.1 Οι βασικές λειτουργίες ενός ΣΔΒΔ

Ένα ΣΔΒΔ μπορεί είτε να είναι εγκατεστημένο σε ένα υπολογιστή και να χρησιμοποιείται από έναν μόνο χρήστη (Single User System) είτε να βρίσκεται εγκατεστημένο σε μια ομάδα υπολογιστών, που επικοινωνούν μέσω ενός δικτύου (τοπικού ή απομακρυσμένου δικτύου) και να χρησιμοποιείται από πολλούς χρήστες (Multi-User System). Κατά κάποιον τρόπο, ένα ΣΔΒΔ λειτουργεί ως ένας συνδεδεμένος κρίκος μεταξύ της βάσης δεδομένων, όπου βρίσκονται αποθηκευμένα τα δεδομένα, και του τελικού χρήστη μέσω του λογισμικού που διαχειρίζεται και μεταφέρει τις πληροφορίες. Τις περισσότερες φορές η βάση δεδομένων βρίσκεται αποθηκευμένη σε μια ενιαία θέση, δηλαδή σε έναν κεντρικό υπολογιστή (server), στον οποίο έχουν όλοι οι χρήστες του συστήματος.



Εικόνα 3: Διαγραμματική αναπαράσταση του ρόλου ενός ΣΔΒΔ.

Συνοπτικά, ο λειτουργικός ρόλος ενός ΣΔΒΔ αφορά:

- α. Την σωστή οργάνωση της βάσης δεδομένων στο μέσο που βρίσκεται αποθηκευμένη, όπως ένας σκληρός δίσκος (HDD/SSD) ή ένα νέφος (Cloud-based servers).
- β. Τους μηχανισμούς με τους οποίους παράγονται και διαχειρίζονται τα δεδομένα στη βάση δεδομένων.

- γ. Την σωστή και ασφαλή αναπαράσταση των δεδομένων στη μορφή που ζητούνται από τις εφαρμογές (desktop, mobile).

Σήμερα, ο όρος «ΣΔΒΔ» έχει αντικατασταθεί από τον όρο «**ΣΣΔΒΔ**», δηλαδή Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (Relational Database Management Systems), καθώς ενσωματώνουν το **Σχεσιακό μοντέλο** (Relational Model) και το **Μοντέλο Οντοτήτων-Συσχετίσεων** (Entity Relationship Model) στον σχεδιασμό της βάσης δεδομένων και στην γρήγορη αποθήκευση και ανάκτηση των δεδομένων, ενώ την τελευταία δεκαετία έχει επικρατήσει η έννοια «**Πληροφοριακά Συστήματα**». Τα Πληροφοριακά Συστήματα αποτελούν το αναπόσπαστο κομμάτι στις επιχειρήσεις και τους οργανισμούς, καθώς έχουν την δυνατότητα να διατυπώνουν ερωτήματα, να αντιλαμβάνονται τις πιθανές τους απαντήσεις και να λαμβάνει τις κατάλληλες σωστές αποφάσεις σχετικά με τα προβλήματα που καλείται να επεξεργαστεί.

2.1.2 Η αρχιτεκτονική ενός ΣΔΒΔ

Όπως αναλύσαμε στην προηγούμενη παράγραφο, ένα ΣΔΒΔ διαχειρίζεται (προσθήκη, ενημέρωση, διαγραφή) τα δεδομένα μιας βάσης, μέσω των ερωτημάτων που θέτουν οι χρήστες της. Τα ερωτήματα αυτά διεκπεραιώνονται από την εφαρμογή λογισμικού μέσω μιας διεπαφής χρήστη εξασφαλίζοντας έτσι την ανεξαρτησία των δεδομένων από τις εφαρμογές.

Με λίγα λόγια, οποιαδήποτε αλλαγή υπάρξει στον τομέα της βάσης που αφορά την οργάνωση των δεδομένων, δεν σημαίνει ότι πρέπει να υπάρξουν και αλλαγές στην εφαρμογή. Βεβαίως ισχύει και το αντίστροφο, δηλαδή οποιοσδήποτε αλλαγές συμβούν στην εφαρμογή δεν επηρεάζουν τα αρχεία της βάσης δεδομένων.

Έτσι με την παραπάνω προσέγγιση σύμφωνα με το πρότυπα ANSI/SPARC, προέκυψε ο αρχιτεκτονικός σχεδιασμός ενός ΣΔΒΔ των τριών επιπέδων:

1. Το **Εσωτερικό ή Φυσικό επίπεδο (Internal / Physical Level)**, που αφορά με τον τρόπο αποθήκευσης τις βάσης δεδομένων στις συσκευές αποθήκευσης (δίσκοι HDD/SDD).
2. Το **Εξωτερικό επίπεδο (External Level)**, που αφορά την κατανομή των χρηστών που χρησιμοποιούν τα δεδομένα, δηλαδή είτε πρόκειται για απλούς χρήστες είτε για προγραμματιστές είτε για διαχειριστές της βάσης δεδομένων.

3. Το **Εννοιολογικό επίπεδο (Conceptual Level)**, που αφορά την σύνδεση των δύο προηγούμενων επιπέδων μεταξύ τους και καλύπτει τη λογική σχεδίαση της βάσης δεδομένων.

Έτσι με τον παραπάνω διαχωρισμό, προκύπτουν και οι αντίστοιχες **ομάδες χρηστών** ενός ΣΔΒΔ. Αυτές συνοψίζονται ως εξής:

- A. **Χρήστες Προσκήνιου (Front-end Users)**, που διαχειρίζονται ένα ΣΔΒΔ στο εξωτερικό επίπεδο και είναι:

1. **Διαχειριστές (Administrators)**, οι οποίοι διαχειρίζονται το λογισμικό εφαρμογής του ΣΔΒΔ και του απαραίτητου hardware, παρέχουν τις απαραίτητες εξουσιοδοτήσεις στους υπόλοιπους χρήστες και είναι υπεύθυνοι για την συνολική απόδοση και ασφάλεια όλου του συστήματος.
2. **Προγραμματιστές εφαρμογών (Software Developers)** οι οποίοι έρχονται σε επαφή με τους χρήστες της εφαρμογής, καταγράφουν τις απαιτήσεις που προκύπτουν και σχεδιάζουν διεπαφές (όψεις) για διαφορετικές ομάδες χρηστών. Ορισμένες φορές συμβάλλουν και στον ανασχεδιασμό της βάσης δεδομένων.
3. **Απλοί ή Τελικοί χρήστες (Simple/End Users)**, που είναι οι χρήστες της εφαρμογής που μπορεί να είναι περιστασιακοί, απλοί ή εξειδικευμένοι και ο κύριος ρόλος τους είναι η εισαγωγή, η ενημέρωση και η ανάκτηση δεδομένων από το ΣΔΒΔ.

- B. **Χρήστες Παρασκήνιου (Back-end Users)**, που διαχειρίζονται ένα ΣΔΒΔ στο εσωτερικό και εννοιολογικό επίπεδο και είναι:

1. **Διαχειριστές Βάσεων Δεδομένων (Database Administrators)**, οι οποίοι έχουν ως κύρια αρμοδιότητα τον τρόπο οργάνωσης των δεδομένων στην βάση, την σωστή λειτουργία του ΣΔΒΔ, τη διαχείριση όλων των χρηστών και την αποκατάσταση των δεδομένων σε περιπτώσεις απώλειας δεδομένων λόγω αστοχίας του υλικού του συστήματος, με την βοήθεια εξειδικευμένων εφαρμογών διαχείρισης βάσεων δεδομένων.
2. **Αναλυτές / Σχεδιαστές Βάσεων Δεδομένων (Database Analysts / Developers)**, οι οποίοι είναι και οι άμεσα υπεύθυνοι τον σωστό σχεδιασμό της βάσης δεδομένων. Αρχικώς προσδιορίζουν με προσοχή τις απαραίτητες οντότητες και τις συσχετίσεις που υπάρχουν μεταξύ τους, σύμφωνα με τον

πραγματικό κόσμο. Έπειτα, σχεδιάζουν, περιγράφουν και αναλύουν τα αντικείμενα του πραγματικού κόσμου με τα αντικείμενα της βάσης δεδομένων. Τέλος, διαθέτουν εξειδικευμένα εργαλεία λογισμικού με την βοήθεια των οποίων αυτοματοποιούν την διαδικασία περιγραφής των δεδομένων.

- 3. Χειριστές και προσωπικό συντήρησης (Database Support)**, που είναι προγραμματιστές εφαρμογών αλλά ο ρόλος τους περιορίζεται στην σωστή λειτουργία και την συντήρηση του υλικού και του λογισμικού των παραπάνω χρηστών.

Τέλος, αξίζει να αναφερθεί, πως ένα ΣΔΒΔ εξαρτάται άμεσα από την υλικο-τεχνολογική υποδομή που βρίσκεται εγκατεστημένο. Σε **επίπεδο λογισμικού**, αποτελείται από πολλά υποσυστήματα όπως ο διαχειριστής αρχείων (File Manager), ο Διαχειριστής Αποθήκης (Storage Manager), ο Διαχειριστής Συναλλαγών (Transaction Manager), ο Μεταγλωττιστής Ερωτημάτων Data (DML Query Compiler), το σύστημα διαχείρισης προσωρινής μνήμης (Buffer Manager) και το σύστημα διαχείριση Αποθήκευσης δεδομένων (Storage Manager). Τα υποσυστήματα αυτά συνδέονται και ανατροφοδοτούνται μεταξύ τους ώστε να διαχειρίζονται αποτελεσματικά και αποδοτικά την βάση δεδομένων και εξαρτώνται άμεσα από το λειτουργικό σύστημα (Windows, Unix, κτλ) στο οποίο βρίσκονται εγκατεστημένα.

Σε **επίπεδο υλικού**, ένα ΣΔΒΔ αποτελείται από ένα σύνολο ηλεκτρονικών υπολογιστών εξοπλισμένοι με τις κατάλληλες συσκευές όπως μονάδες δίσκων HDD/SDD, κανάλια εισόδου/εξόδου, δρομολογητές, συστήματα ψύξης και οποιαδήποτε συσκευή που διασυνδέει τους υπολογιστές και τον πραγματικό κόσμο.

2.1.3 Τα ΣΔΒΔ σήμερα

Υπάρχουν εκατοντάδες ΣΔΒΔ σε όλον τον κόσμο τα οποία αλλάζουν και τροποποιούνται καθημερινώς από τους κατασκευαστές προκειμένου να καλύψουν τις ανάγκες των επιχειρήσεων. Σύμφωνα με την δημοφιλή ιστοσελίδα «DB-Engines», τα πιο δέκα πιο δημοφιλή ΣΔΒΔ που χρησιμοποιούνται αυτή τη στιγμή (Απρίλιος 2022) στο κόσμο είναι τα παρακάτω:

Πίνακας 1: Τα δέκα πιο δημοφιλή ΣΔΒΔ στον κόσμο (Απρίλιος 2022). Πηγή: <https://db-engines.com/en/ranking>

Απρίλιος 2022	Μάρτιος 2022	Απρίλιος 2021	ΣΔΒΔ (Κατασκευαστής)
1	1	1	Oracle
2	2	2	MySQL
3	3	3	Microsoft SQL Server
4	4	4	PostgreSQL

5	5	5	MongoDB
6	6	↑ 7	Redis
7	↑ 8	↑ 8	Elasticsearch
8	↓ 7	↓ 6	IBM Db2
9	9	↑ 10	Microsoft Access
10	10	↓ 9	SQLite

Σύμφωνα με την εταιρία, τα κριτήρια της κατάταξης των παραπάνω ΣΔΒΔ είναι ο αριθμός των αναφορών του συστήματος στις μηχανές αναζήτησης (Google, Bing, κτλ.), το ενδιαφέρον (τάση) των χρηστών για το ΣΔΒΔ, η συχνότητα των συζητήσεων σχετικά με το σύστημα σε φόρουμ προγραμματιστών (Stack Overflow, DBA Stack Exchange), ο αριθμός θέσεων εργασίας που προσφέρονται για το εκάστοτε σύστημα, το πλήθος των προφίλ χρηστών του LinkedIn που έχουν επαγγελματική εμπειρία με ένα ΣΔΒΔ και οι αναφορές των ΣΔΒΔ στα κοινωνικά δίκτυα όπως το Twitter.

2.2 Η γλώσσα SQL



Η γλώσσα **SQL (Structured Query Language)** είναι η γλώσσα διαχείρισης των σχεσιακών βάσεων Δεδομένων και αναπτύχθηκε, αρχικώς από την εταιρεία IBM στις αρχές της δεκαετίας του 1970 με το όνομα «SEQUEL». Αποτελεί την κυριότερη αιτία που οι σχεσιακές βάσεις δεδομένων κυριάρχησαν εμπορικά σε όλο τον κόσμο καθώς είναι απλή και εύκολη στη χρήση της. Καθιερώθηκε ως πρότυπο ANSI το 1986 και το 1987 ως πρότυπο ISO. Εκ τότε, έχουν υπάρξει εννέα αναθεωρήσεις με την πιο πρόσφατη αυτή του 2019.

2.2.1 Κύρια Χαρακτηριστικά

Με την καθιέρωση της SQL ως πρότυπο ANSI/ISO, μπορεί να χρησιμοποιηθεί από όλους τους προγραμματιστές ή σχεδιαστές βάσεων δεδομένων σε οποιοδήποτε σύστημα την υποστηρίζει. Με αυτό τον τρόπο εξασφαλίζεται η **φορητότητα** μιας βάσης δεδομένων από ένα σύστημα σε ένα άλλο, για παράδειγμα από Oracle σε MySQL ή και αντίστροφα.

Επίσης, ένας προγραμματιστής εφαρμογών μπορεί να χρησιμοποιεί την γλώσσα SQL για την δημιουργία ερωτημάτων (queries) σε διαφορετικές βάσεις δεδομένων που υποστηρίζουν

την γλώσσα SQL, για να αντλεί τα δεδομένα των εφαρμογών του είτε πρόκειται για SQL Server, είτε Oracle, είτε Microsoft Access.

Η **δομή** της γλώσσας SQL βασίστηκε στη **Σχισιακή Άλγεβρα**, καθώς οι πράξεις της παίζουν σημαντικό ρόλο στην αποσαφήνιση των διάφορων ερωτημάτων προς το ΣΔΒΔ και στην επεξεργασία και την βελτιστοποίησή τους. Βέβαια, ενώ στην Σχισιακή Άλγεβρα πρέπει να οριστεί επ' ακριβώς η σειρά με την οποία θα εκτελεστούν οι πράξεις μιας σύνθετης έκφρασης, στην SQL αναλαμβάνει το ΣΔΒΔ να εκτελέσει τις πράξεις/ερωτήματα υψηλού επιπέδου με τον πιο αποδοτικό τρόπο.

Ως προς το **περιεχόμενο** των **εντολών** της η γλώσσα SQL χωρίζεται σε τέσσερις βασικές κατηγορίες οι οποίες είναι:

- A.** Η **γλώσσα ορισμού Δεδομένων (Data Definition Language – DDL)**, η οποία περιέχει εντολές **CREATE**, **ALTER** και **DROP** που χρησιμοποιούνται για τη δημιουργία και την επεξεργασία της μορφής των πινάκων (σχήμα δεδομένων).
- B.** Η **γλώσσα χειρισμού δεδομένων (Data Manipulation Language – DML)**, η οποία περιέχει τις εντολές **INSERT**, **UPDATE** και **DELETE** που χρησιμοποιούνται για την εισαγωγή, ενημέρωση και διαγραφή των δεδομένων που είναι αποθηκευμένα σε πίνακες (χειρισμός δεδομένων).
- Γ.** Η **γλώσσα ερωτημάτων δεδομένων (Data Query Language – DQL)**, η οποία περιέχει την εντολή **SELECT** για την ανάκτηση των δεδομένων που είναι αποθηκευμένα σε πίνακες της βάσης δεδομένων.
- Δ.** Η **γλώσσα ελέγχου δεδομένων (Data Control Language – DCL)**, η οποία περιέχει τις εντολές **GRANT**, **DENY**, **REVOKE**, **COMMIT** και **ROLLBACK** που καθορίζουν τον έλεγχο πρόσβασης του κάθε χρήστη στα δεδομένα που είναι αποθηκευμένα στους πίνακες της βάσης και σε λειτουργίες της βάσης δεδομένων.

Οι **τύποι δεδομένων** που αποθηκεύονται στους πίνακες καθορίζονται από την ίδια την SQL. Υποστηρίζει διάφορους τύπους δεδομένων όπως αριθμητικούς, αλφαριθμητικούς, ημερομηνίας, ώρας, κλπ οι οποίοι επιλέγονται από το σχεδιαστή της βάσης ανάλογα με τις ανάγκες που θέλει να καλύψει. Βέβαια σε κάθε ΣΔΒΔ οι τύποι δεδομένων μπορεί να διαφέρουν ονομαστικά μεταξύ τους, για παράδειγμα ένα αλφαριθμητικό ορίζεται στην MySQL ως «*char*» ενώ στον SQL Server ως «*nvarchar*».

Τέλος αξίζει να σημειωθεί, πως όλες οι εντολές και οι τύποι δεδομένων της γλώσσας SQL υπάρχουν ενσωματωμένοι σε αρκετές γλώσσες προγραμματισμού, όπως η Java, η C/C++, η C#, η Delphi, η Visual Basic .NET και η Python.

2.2.2 Επεκτάσεις

Ενώ η γλώσσα SQL έχει σχεδιαστεί για την αποθήκευση και την αναζήτηση δεδομένων σε μια βάση δεδομένων, θα ήταν εξαιρετικά δύσκολο για τους προγραμματιστές και τους σχεδιαστές να αντλούν και να διαχειρίζονται τα δεδομένα αυτά, εάν δεν υπήρχαν οι επεκτάσεις της γλώσσας SQL.

Οι επεκτάσεις της SQL ουσιαστικά είναι η προσθήκη εντολών του διαδικαστικού προγραμματισμού στο σύνολο εντολών της, προσφέροντας ευκολία στην σύνταξη σύνθετων προγραμμάτων σε ένα ΣΔΒΔ.

Οι εντολές IF, ELSE, WHILE, FOR, BEGIN, END σε μια ομάδα εντολών και σε συνδυασμό τοπικών αναγνωριστικών (μεταβλητές και σταθερές) μπορούν να χρησιμοποιηθούν σε αποθηκευμένες διαδικασίες (procedures), σε συναρτήσεις χρηστών (functions) και σε σκανδάλες (triggers) που ενεργοποιούνται σε οποιαδήποτε αλλαγή γίνει στους πίνακες της βάσης. Οι επεκτάσεις της γλώσσας SQL είναι κοινές για όλα τα ΣΔΒΔ και διαφέρουν ελάχιστα στη δομή τους και κάποιες από τις δημοφιλείς επεκτάσεις περιέχονται στον παρακάτω πίνακα.

Πίνακας 2: Δημοφιλείς επεκτάσεις της γλώσσας SQL. Πηγή: https://en.wikipedia.org/wiki/SQL#Procedural_extensions

ΣΔΒΔ	Επέκταση
ANSI/ISO Standard	SQL/PSM
IBM Db2	SQL PL
Interbase / Firebird	PSQL
MariaDB	SQL/PSM, PL/SQL
Microsoft Access	Jet SQL
Microsoft / Sybase	T-SQL
MySQL	SQL/PSM
Oracle	PL/SQL
PostgreSQL	PL/pgSQL

2.2.3 Η επέκταση Jet SQL της Microsoft

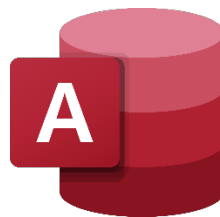
Η επέκταση Jet SQL (Joint Engine Technology) της γλώσσας SQL δημιουργήθηκε από τη Microsoft το 1992 και αποτελεί τον κύριο μηχανισμό που είναι υπεύθυνος για την σωστή

λειτουργία της MS Access. Όπως και όλες οι επεκτάσεις της SQL, η Jet SQL χρησιμοποιείται μόνο για την ανάκτηση πληροφοριών από μια βάση δεδομένων που έχει υλοποιηθεί με την MS Access. Αυτό βέβαια είναι και ο περιορισμός που έχει σαν επέκταση γιατί δεν είναι σχεδιασμένη για να διαχειρίζεται με ασφάλεια μια βάση δεδομένων.

Ωστόσο διαθέτει έναν μηχανισμό που δίνει στον προγραμματιστή την ευκολία για να διαχειρίζεται τα δεδομένα μιας βάσης, το Αντικείμενο Πρόσβασης Δεδομένων (Data Access Object – DAO). Το **DAO** περιέχει ενσωματωμένες βιβλιοθήκες που έχουν σχεδιαστεί για τη δημιουργία, την τροποποίηση και την διαχείριση της ασφάλειας μιας βάσης δεδομένων.

Η Jet SQL είναι ενσωματωμένη σε όλες τις εκδόσεις της Microsoft Access, σε συνδυασμό με την Visual Basic, μέχρις σήμερα με το όνομα «**Access Database Engine**». Ωστόσο προκειμένου να καλυφθούν οι ανάγκες για μεγαλύτερες βάσεις δεδομένων, η επέκταση Jet μπορεί να αναβαθμιστεί εύκολα στην επέκταση της T-SQL μέρος του MS SQL Server.

2.3 Η Microsoft Access



Το λογισμικό «Microsoft Access» ή «MS Access» κυκλοφόρησε από την Microsoft για πρώτη φορά στην αγορά στις 13 Νοεμβρίου 1992 ως ξεχωριστή εφαρμογή και από τον Αύγουστο του 1995 κυκλοφορεί, μέχρι και στις μέρες μας (πρόσφατη έκδοση 2019), και ως μέρος της ομάδας εφαρμογών Microsoft Office. Η εφαρμογή που θα υλοποιήσουμε βασίζεται στην έκδοση 2019.

2.3.1 Λειτουργίες

Η MS Access είναι ένα **πλήρες προγραμματιστικό περιβάλλον** (IDE) καθώς συνδυάζει τον μηχανισμό της βάσης δεδομένων ACDE με ένα γραφικό περιβάλλον χρήστη (GUI) με τα κατάλληλα εργαλεία ανάπτυξης λογισμικού και εφαρμογών. Επιπλέον, προσφέρει στον χρήστη τη δυνατότητα να εισάγει ή να συνδέεται απευθείας με δεδομένα που βρίσκονται αποθηκευμένα σε άλλες εφαρμογές (MS Excel, κτλ.) και βάσεις δεδομένων (SQL Server, MySQL, κτλ.).

Λόγω του **απλού** και **φιλικού** προς τον χρήστη περιβάλλον, χρησιμοποιείται ευρέως από τους προγραμματιστές εφαρμογών και τους διαχειριστές βάσεων δεδομένων για την ανάπτυξη απλών και σύνθετων εφαρμογών λογισμικού. Έτσι ένας προγραμματιστής μπορεί μέσα από το GUI της Access να δημιουργήσει πίνακες, ερωτήματα και όψεις, να ορίσει τις σχέσεις μεταξύ των πινάκων, να σχεδιάσει φόρμες και εκθέσεις και να προγραμματίσει λειτουργικές μονάδες και μακροεντολές.

Όπως σε όλες τις εφαρμογές της ομάδας του Microsoft Office, η MS Access έχει ενσωματωμένη την γλώσσα προγραμματισμού **VBA (Visual Basic for Applications)** που επιτρέπει στους προγραμματιστές να διαχειρίζονται και να χρησιμοποιούν μια ποικιλία από οπτικά αντικείμενα (κουμπιά, λίστες, κτλ.) σε φόρμες και εκθέσεις με την χρήση κώδικα VBA.

Διαθέτει **2 είδη αρχείων** (από την έκδοση **2007** και έπειτα) τα οποία είναι:

- α. *.accdb** (Access Database) που είναι η προεπιλεγμένη μορφή αρχείου της MS Access και δίνει την δυνατότητα στον χρήστη να κάνει αλλαγές και τροποποιήσεις στους πίνακες, στις φόρμες, να ενημερώσει τον κώδικα VBA, κτλ. και γενικότερα είναι το αρχείο της βάσης δεδομένων.

Για τις παλαιότερες εκδόσεις Access (1995, 2000, 2003) το αντίστοιχο αρχείο είναι το ***.mdb**, το οποίο υποστηρίζεται πλήρως από τις νέες εκδόσεις.

- β. *.accde** (Access Database Executable Only) που είναι το **εκτελέσιμο** αρχείο (μεταγλωττισμένο) της εφαρμογής και δεν δίνεται η δυνατότητα στον τελικό χρήστη να επέμβει στον κώδικα VBA της εφαρμογής και στους πίνακες της βάσης δεδομένων.

Για τις παλαιότερες εκδόσεις Access (1995, 2000, 2003) το αντίστοιχο αρχείο είναι το ***.mde**, το οποίο μπορεί να μετατραπεί εύκολα στη νέα έκδοση.

Συνοπτικά ένα αρχείο της MS Access μπορεί και περιέχει τα εξής αντικείμενα:

- α.** Πίνακες (Tables)
- β.** Ερωτήματα (Queries)
- γ.** Σχέσεις (Relationships)
- δ.** Φόρμες (Forms)
- ε.** Αναφορές (Reports)
- στ.** Μακροεντολές (Macros)
- ζ.** Λειτουργικές μονάδες (Modules)

Η MS Access μπορεί επίσης να χρησιμοποιηθεί και ως το **προσκήνιο** (front-end) μιας εφαρμογής, ενώ το **παρασκήνιο** (back-end), δηλαδή η βάση (πίνακες, όψεις, κτλ), να βρίσκεται αποθηκευμένο σε άλλο λογισμικό όπως ο MS SQL Server και σε λογισμικά που δεν ανήκουν στην Microsoft όπως η Oracle και η Sybase. Ακόμα, ένα αρχείο Access μπορεί να χρησιμοποιηθεί και να διαχειριστεί εύκολα σε ένα σύγχρονο προγραμματιστικό περιβάλλον όπως το MS Visual Studio (VB.NET, C#, ASP), το Embarcadero RAD Builder (C/C++, Delphi) και το Apache NetBeans (Java).

2.3.2 Περιβάλλον χρήσης

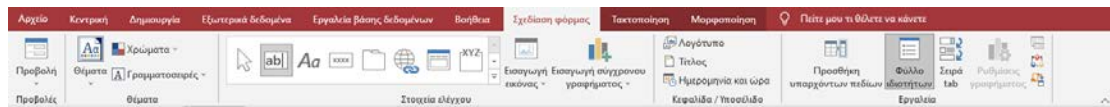
Το περιβάλλον χρήσης της MS Access προσφέρει στον προγραμματιστή ένα πλήρες προγραμματιστικό περιβάλλον και ακολουθεί τον σχεδιασμό όλων των προγραμμάτων της οικογένειας Office, από την έκδοση 2007 μέχρι και σήμερα.

Αριθμός	Όνομα	Παράρτημα	Ημερομηνία	Πόλη	Κόστος	Φόρος					
30	Αννα Μαρκοπούλου	Εταιρεία ΚΖ	15/1/2006	Μεσοφωρές Β	Καλώς ήρθατε Αρκαδίου 17	Αθήνα	Αττική	9999	Ελλάδα	200,00 €	0,00 €
31	Ιων Κωστός	Εταιρεία Δ	20/2/2006	Μεσοφωρές Α	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	5,00 €	0,00 €	
32	Μαρία Σαφάκια	Εταιρεία Β	22/2/2006	Μεσοφωρές Β	Μαρία Έργου Δολιχάνη 2	Αθήνα	Αττική	9999	Ελλάδα	5,00 €	0,00 €
33	Μυέλλη Παναδοί	Εταιρεία Η	30/2/2006	Μεσοφωρές Γ	Ελένη Ηλιακού Ηθάρης 8	Κόρινθος	Κορινθία	9999	Ελλάδα	50,00 €	0,00 €
34	Αννα Μαρκοπούλου	Εταιρεία Δ	6/2/2006	Μεσοφωρές Γ	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	4,00 €	0,00 €	
35	Ιων Κωστός	Εταιρεία ΚΘ	10/2/2006	Μεσοφωρές Β	Αλέξανδρος Α' Ηρώων 4	Καρόλινα	Καρόλινα	9999	Ελλάδα	7,00 €	0,00 €
36	Μαρία Σαφάκια	Εταιρεία Γ	21/2/2006	Μεσοφωρές Β	Γεώργιος Σαφέλης Αγ. Γεωργίου 1	Αρκαδία	9999	Ελλάδα	2,00 €	0,00 €	
37	Ζωή Νικολάου	Εταιρεία ΣΤ	6/2/2006	Μεσοφωρές Β	Βασίλειος Σαπής Κηφισίας 3848	Αργολίδα	9999	Ελλάδα	12,00 €	0,00 €	
38	Αννα Μαρκοπούλου	Εταιρεία ΚΗ	12/3/2006	Μεσοφωρές Γ	Δημήτριος Κατ. Αγίου Κωνσταντ. Ρέθυμνο	Ρέθυμνο	9999	Ελλάδα	10,00 €	0,00 €	
39	Ιων Κωστός	Εταιρεία Η	22/3/2006	Μεσοφωρές Β	Ελένη Ηλιακού Ηθάρης 8	Κόρινθος	Κορινθία	9999	Ελλάδα	3,00 €	0,00 €
40	Μαρία Σαφάκια	Εταιρεία Ι	24/3/2006	Μεσοφωρές Β	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	5,00 €	0,00 €	
41	Μαρία Νικολάου	Εταιρεία Ζ	24/3/2006	Μεσοφωρές Α	Δημήτριος Κιούρ Πυλίων 4	Αττική	9999	Ελλάδα	0,00 €	0,00 €	
42	Μαρία Νικολάου	Εταιρεία Α	24/3/2006	Μεσοφωρές Α	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	0,00 €	0,00 €	
43	Μαρία Νικολάου	Εταιρεία ΙΑ	24/3/2006	Μεσοφωρές Γ	Λεωνίδας Κατ. Αργολίδων 62	Αργολίδα	9999	Ελλάδα	0,00 €	0,00 €	
44	Μαρία Νικολάου	Εταιρεία Α	24/3/2006	Μεσοφωρές Β	Αννα Σαπής Αρκαδίας 11	Αρκαδία	9999	Ελλάδα	0,00 €	0,00 €	
45	Μαρία Νικολάου	Εταιρεία ΚΗ	7/4/2006	Μεσοφωρές Γ	Δημήτριος Κατ. Αγίου Κωνσταντ. Ρέθυμνο	Ρέθυμνο	9999	Ελλάδα	40,00 €	0,00 €	
46	Γιάννης Μαρκοπούλου	Εταιρεία Θ	5/4/2006	Μεσοφωρές Α	Σταύρος Τζεβελέκης Αρκαδίας 17	Αρκαδία	9999	Ελλάδα	300,00 €	0,00 €	
47	Μυέλλη Παναδοί	Εταιρεία ΣΤ	3/4/2006	Μεσοφωρές Β	Βασίλειος Σαπής Κηφισίας 3848	Αργολίδα	9999	Ελλάδα	300,00 €	0,00 €	
48	Μαρία Σαφάκια	Εταιρεία Η	5/4/2006	Μεσοφωρές Β	Ελένη Ηλιακού Ηθάρης 8	Κόρινθος	Κορινθία	9999	Ελλάδα	50,00 €	0,00 €
50	Αννα Μαρκοπούλου	Εταιρεία ΚΤ	5/4/2006	Μεσοφωρές Α	Ανδρέας Κατ. Σ. Πύργου 1	Μεσσηνία	9999	Ελλάδα	3,00 €	0,00 €	
51	Αννα Μαρκοπούλου	Εταιρεία ΚΤ	5/4/2006	Μεσοφωρές Γ	Σπύρος Σαπής Αρκαδίας 17	Αρκαδία	9999	Ελλάδα	60,00 €	0,00 €	
55	Μαρία Νικολάου	Εταιρεία ΚΘ	5/4/2006	Μεσοφωρές Β	Αλέξανδρος Α' Ηρώων 4	Καρόλινα	Καρόλινα	9999	Ελλάδα	200,00 €	0,00 €
56	Χρυσή Αμπελιά	Εταιρεία ΣΤ	5/4/2006	Μεσοφωρές Γ	Βασίλειος Σαπής Κηφισίας 3848	Αργολίδα	9999	Ελλάδα	0,00 €	0,00 €	
57	Αννα Μαρκοπούλου	Εταιρεία ΚΖ	22/2/2006	Μεσοφωρές Β	Καλώς ήρθατε Αρκαδίου 17	Αθήνα	Αττική	9999	Ελλάδα	200,00 €	0,00 €
58	Ιων Κωστός	Εταιρεία Δ	22/4/2006	Μεσοφωρές Α	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	5,00 €	0,00 €	
59	Μαρία Σαφάκια	Εταιρεία Β	22/4/2006	Μεσοφωρές Β	Μαρία Σαπής Αρκαδίας 17	Αθήνα	Αττική	9999	Ελλάδα	5,00 €	0,00 €
60	Μυέλλη Παναδοί	Εταιρεία Η	30/4/2006	Μεσοφωρές Γ	Ελένη Ηλιακού Ηθάρης 8	Κόρινθος	Κορινθία	9999	Ελλάδα	50,00 €	0,00 €
61	Αννα Μαρκοπούλου	Εταιρεία Δ	7/4/2006	Μεσοφωρές Γ	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	4,00 €	0,00 €	
62	Μαρία Σαφάκια	Εταιρεία ΚΘ	25/4/2006	Μεσοφωρές Β	Αλέξανδρος Α' Ηρώων 4	Καρόλινα	Καρόλινα	9999	Ελλάδα	7,00 €	0,00 €
63	Μαρία Σαφάκια	Εταιρεία Γ	25/4/2006	Μεσοφωρές Β	Γεώργιος Σαφέλης Αγ. Γεωργίου 1	Αρκαδία	9999	Ελλάδα	2,00 €	0,00 €	
64	Ζωή Νικολάου	Εταιρεία ΣΤ	5/5/2006	Μεσοφωρές Β	Βασίλειος Σαπής Κηφισίας 3848	Αργολίδα	9999	Ελλάδα	12,00 €	0,00 €	
65	Αννα Μαρκοπούλου	Εταιρεία ΚΗ	11/5/2006	Μεσοφωρές Γ	Δημήτριος Κατ. Αγίου Κωνσταντ. Ρέθυμνο	Ρέθυμνο	9999	Ελλάδα	10,00 €	0,00 €	
66	Ιων Κωστός	Εταιρεία Η	24/5/2006	Μεσοφωρές Β	Ελένη Ηλιακού Ηθάρης 8	Κόρινθος	Κορινθία	9999	Ελλάδα	3,00 €	0,00 €
67	Μαρία Σαφάκια	Εταιρεία Ι	24/5/2006	Μεσοφωρές Β	Χρυσή Αμπελιά Μάρκου Μπότσαρη	Αρκαδία	9999	Ελλάδα	5,00 €	0,00 €	
68	Μαρία Νικολάου	Εταιρεία Ζ	24/5/2006	Μεσοφωρές Α	Δημήτριος Κιούρ Πυλίων 4	Αττική	9999	Ελλάδα	0,00 €	0,00 €	

Εικόνα 4: Δείγμα του περιβάλλοντος χρήσης της MS Access.

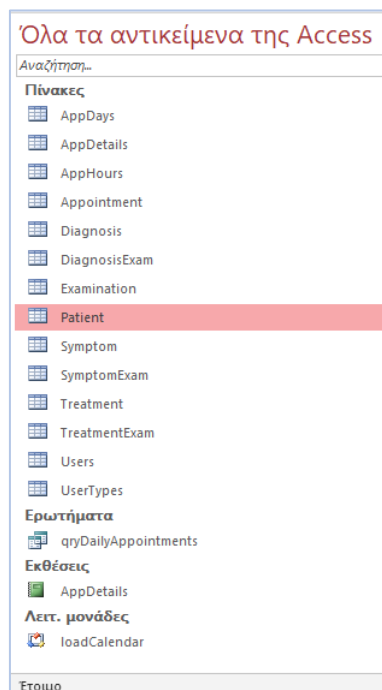
Το περιβάλλον της Access χωρίζεται σε τέσσερα βασικά μέρη:

1. Την **Κορδέλα των καρτελών** (Ribbon tabs), όπου περιέχονται ομαδοποιημένα τα κουμπιά των βασικών λειτουργιών της Access σε καρτέλες (tab) οι οποίες είναι η «Κεντρική», «Σχεδίαση», «Εξωτερικά Δεδομένα», «Εργαλεία Βάσης δεδομένων» και «Βοήθεια». Συμπληρωματικά, κάθε λειτουργία σχεδίασης εμφανίζει την δική της καρτέλα όταν αυτή ενεργοποιείται, για παράδειγμα «Σχεδίαση Ερωτήματος», «Μορφοποίηση», «Εργαλεία Σχεδίασης Φόρμας», κτλ.



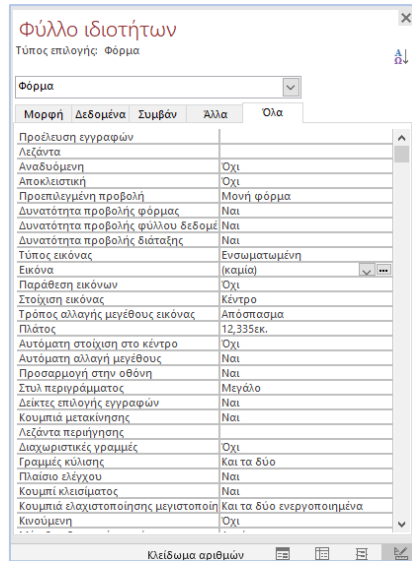
Εικόνα 5: Η κορδέλα των καρτέλων. Διακρίνονται οι κύριες καρτέλες και η πρόσθετες καρτέλες που «Σχεδίαση φόρμας», «Τακτοποίηση» και «Μορφοποίηση».

2. Τον **Πίνακα Αντικειμένων** (Navigation Panel), που βρίσκεται κατά προεπιλογή στα αριστερά του παραθύρου και περιέχει ομαδοποιημένα όλα τα αντικείμενα που περιέχονται στο αρχείο της Access, όπως πίνακες, ερωτήματα, φόρμες, εκθέσεις, μακροεντολές και λειτουργικές μονάδες, τα οποία μπορεί ο χρήστης να διαχειριστεί (μετονομασία, αντιγραφή, διαγραφή, κτλ).



Εικόνα 6: Ο Πίνακας Αντικειμένων της Access.

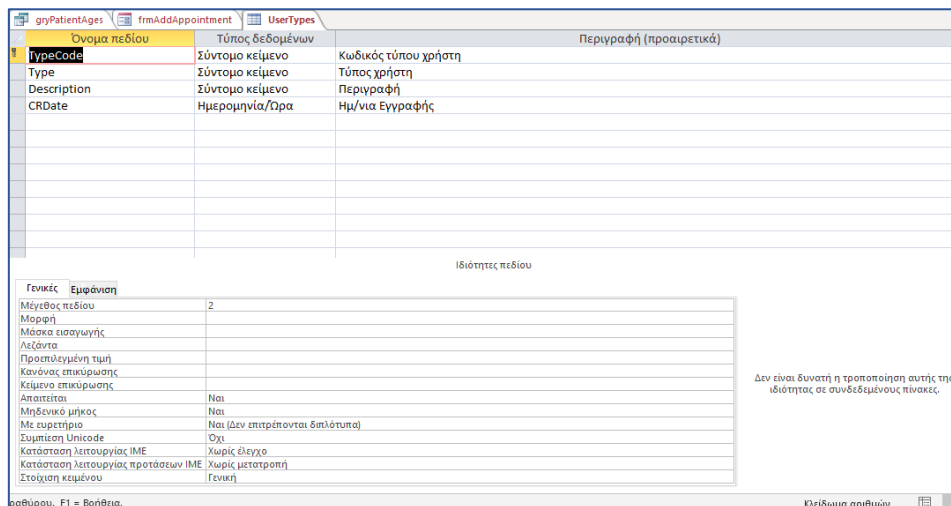
3. Το **Φύλλο Ιδιοτήτων (Properties Tab)** το οποίο περιέχει ταξινομημένα τις ιδιότητες και τα συμβάντα (events) ενός αντικειμένου.



Εικόνα 7: Το Φύλλο Ιδιοτήτων ενός αντικείμενου της MS Access.

Οι ιδιότητες αφορούν τις τιμές των χαρακτηριστικών που έχει ένα αντικείμενο (π.χ. γραμματοσειρά ενός textbox, χρώμα μιας φόρμας, κτλ.) ενώ τα συμβάντα αφορούν τις ενέργειες (κώδικας VBA ή μακροεντολές) που γίνονται όταν ενεργοποιηθούν τα στοιχεία ελέγχου μιας φόρμας (π.χ. συμβάν όταν πατιέται ένα κουμπί στη φόρμα, συμβάν όταν κλείνει μια φόρμα, κτλ.). Ενεργοποιείται κάθε φορά που ανοίγει προς επεξεργασία ένα αντικείμενο.

4. Η περιοχή των αντικειμένων (**Object area**) όπου τα αντικείμενα της Access (πίνακες, φόρμες, ερωτήματα, κτλ.) οργανώνονται σε καρτέλες και είναι διαθέσιμα στον χρήστη για επεξεργασία.



Εικόνα 8: Η περιοχή των αντικειμένων επιτρέπει στο χρήστη να ανοίγει περισσότερα από ένα αντικείμενα της Access, είτε για προβολή είτε για επεξεργασία. Οι λειτουργικές μονάδες ανοίγουν μόνο στον Επεξεργαστή VBA.

Το περιβάλλον της Access προσφέρει τη δυνατότητα στο χρήστη να επεξεργάζεται και να εκτελεί/ελέγχει ταυτόχρονα τα αντικείμενα που σχεδιάζει μέσω των διάφορων προβολών που διαθέτει η Access. Ανάλογα με το αντικείμενο οι προβολές αυτές διαφέρουν. Τις προβολές αυτές θα τις δούμε αναλυτικά στα επόμενα κεφάλαια 3 και 4 όπου περιγράφεται η δημιουργία των πινάκων της βάσης και οι φόρμες της εφαρμογής.

2.3.2 Πλεονεκτήματα – Μειονεκτήματα

Την εποχή που κυκλοφόρησε η MS Access κυκλοφόρησε στην αγορά θεωρούνταν ως καινοτόμος εφαρμογή, καθώς με το χαμηλό κόστος της και τις λειτουργίες που παρείχε, κάλυπτε ολοένα και περισσότερο τις ανάγκες όλων των επιχειρήσεων και των οργανισμών. Σήμερα, όμως τα πράγματα είναι διαφορετικά.

Μεγάλα ΣΔΒΔ υπάρχουν διαθέσιμα δωρεάν στο Διαδίκτυο και μπορούν να ληφθούν από τις επιχειρήσεις, όπως ο MS SQL Server Express Edition, όταν πρόκειται για μικρές εφαρμογές, ενώ εάν χρειάζονται να καλυφθούν μεγαλύτερες ανάγκες, οι εταιρίες διαθέτουν τα απαραίτητα χρηματικά ποσά για την αγορά των κατάλληλων αδειών λογισμικού.

Ωστόσο η MS Access χρησιμοποιείται ακόμα από πολλές επιχειρήσεις, όπως αναφέραμε και στην παράγραφο 2.1.3 λόγω ότι ικανοποιεί γρήγορα τις επιχειρηματικές ανάγκες για πολλούς τύπους λύσεων βάσεων δεδομένων σε μικρή κλίμακα. Τα **πλεονεκτήματα** που παρέχει σε έναν χρήστη ή προγραμματιστή μπορούμε να τα συνοψίσουμε ως εξής:

- α. Ευκολία στην εγκατάσταση και την χρήση.** Η εγκατάσταση της MS Access σε έναν υπολογιστή είναι εύκολη υπόθεση καθώς εγκαθίσταται μαζί με τις υπόλοιπες εφαρμογές του MS Office μέσα σε λίγα λεπτά. Διαθέτει έτοιμους οδηγούς χρήσης που βοηθούν τον προγραμματιστή να σχεδιάσει εύκολα και γρήγορα τα αντικείμενα που θέλει και δεν είναι απαραίτητο να είναι κάποιος γνώστης μιας γλώσσας προγραμματισμού για να την χρησιμοποιήσει. Και αυτός είναι ο λόγος που η MS Access θεωρείται το ευκολότερο ΣΔΒΔ για εκμάθηση μιας βάσης δεδομένων.
- β. Ευκολία στην ενσωμάτωση άλλων εφαρμογών.** Η MS Access μπορεί να αλληλοεπιδρά πλήρως με προγράμματα και εφαρμογές λογισμικού των Windows και βεβαίως με τις υπόλοιπες εφαρμογές της οικογένειας του MS Office (Word, Excel, Visio, κτλ.).
- γ. Φορητότητα και ευελιξία.** Ένα αρχείο της MS Access μπορεί να περιέχει την βάση δεδομένων, τις φόρμες και τις λειτουργικές μονάδες μιας εφαρμογής. Το γεγονός

αυτό επιτρέπει στους προγραμματιστές να αντιγράφουν ή να μετακινούν το αρχείο της Access σε διαφορετικές τοποθεσίες ανάλογα με την χρήση (το οποίο δεν είναι και απόλυτα ασφαλές όπως θα αναλύσουμε παρακάτω) προσφέροντας έτσι την ευελιξία στο χρήστη για να διαχειρίζεται εύκολα τα αρχεία της βάσης.

- δ. Ανάλυση Βάσεων Δεδομένων.** Το περιβάλλον της MS Access προσφέρει μεγάλη ευκολία στην ανάλυση μιας βάσης δεδομένων σε σχέση με άλλα ΣΔΒΔ, καθώς διαθέτει ενσωματωμένους οδηγούς ανάλυσης που μπορούν να καθοδηγούν τον προγραμματιστή σε κάθε βήμα στον σχεδιασμό της βάσης, των φορμών και των αναφορών.
- ε. Εύκολο και ευέλικτο Περιβάλλον χρήστη.** Διαθέτει ένα απλό, προηγμένο και ελκυστικό γραφικό περιβάλλον χρήστη, καθιστώντας το ιδανικό για τη δημιουργία, τη διαχείριση, την ανάλυση των βάσεων δεδομένων και την παραγωγή φορμών και αναφορών.
- στ. Πλήρης συμβατότητα με τη γλώσσα SQL.** Η MS Access παρέχει πλήρη υποστήριξη στη γλώσσα SQL. Ο προγραμματιστής μπορεί να δημιουργήσει εύκολα ερωτήματα στη βάση είτε με απλό τρόπο γράφοντας εντολές SQL είτε με γραφικό τρόπο. Παράλληλα με την ενσωματωμένη γλώσσα προγραμματισμού VBA, ο χρήστης μπορεί να χρησιμοποιεί τις εντολές SQL απευθείας στις λειτουργικές μονάδες ή τις μακροεντολές, πλησιάζοντας έτσι κατά κάποιον τρόπο την αντικειμενοστραφή σχεδίαση μιας εφαρμογής.
- ζ. Κόστος και αποτελεσματικότητα.** Το χαμηλό κόστος εφαρμογής και συντήρησης μιας βάσης δεδομένων της MS Access είναι πολύ πιο φθηνό σε σχέση με τα μεγαλύτερα ΣΔΒΔ όπως η Oracle και ο MS SQL Server, γεγονός που προτιμάται από προγραμματιστές και αναλυτές όταν πρόκειται για μικρές επιχειρήσεις και οργανισμούς ή και τμήματα αυτών.
- η. Εύκολη κοινή χρήση των δεδομένων.** Η MS Access επιτρέπει να γίνεται διαμοιρασμός της βάσης δεδομένων έως και 40 χρήστες ταυτόχρονα οι οποίοι μπορούν να επεξεργάζονται, να κάνουν αλλαγές και να αντλούν τα δεδομένα σε οποιοδήποτε χρόνο. Μάλιστα, οι νεότερες εκδόσεις της MS Access επιτρέπουν στους προγραμματιστές να υλοποιούν διαδικτυακές εφαρμογές που εκτελούνται από κάποιο πρόγραμμα περιήγησης του Διαδικτύου (Chrome, Edge, Firefox, κτλ.) όπου μπορούν να έχουν πρόσβαση οι χρήστες.

- θ. Αυτοματοποίηση συχνών ενεργειών.** Με την βοήθεια των μακροεντολών που παρέχει η MS Access, οι χρήστες μπορούν να αυτοματοποιούν ενέργειες που εκτελούνται συχνά, είτε αφορούν την βάση δεδομένων (ενημέρωση πινάκων, προσθήκη εγγραφών σε πίνακα, κτλ.) είτε λειτουργίες σε μια φόρμα ή αναφορά. Οι μακροεντολές που υλοποιούνται στην MS Access μπορούν να χρησιμοποιηθούν εύκολα και σε άλλες εφαρμογές του Office.
- ι. Οδηγοί χρήσης και πρότυπα εγκατάστασης.** Η MS Access διαθέτει έτοιμα πρότυπα προς χρήση βάσεων δεδομένων εντός του προγράμματος αλλά και για λήψη από τον ιστότοπο της Microsoft. Επιπλέον είναι διαθέσιμες κατά την εκκίνηση του προγράμματος και επιτρέπουν στους χρήστες να βελτιώσουν μια βάση δεδομένων με προκαθορισμένους πίνακες, ερωτήματα, φόρμες, αναφορές και μακροεντολές.
- ια. Εισαγωγή και εξαγωγή δεδομένων.** Μέσω της MS Access, ο χρήστης μπορεί να εισάγει δεδομένα στους πίνακες της βάσης από εξωτερικές πηγές, όπως αρχεία Excel, Word ή ακόμα και αρχεία HTML και μέσω των οδηγών ODBC μπορεί να εισαχθούν δεδομένα και από μια βάση δεδομένων του SQL Server. Επίσης, υποστηρίζει και την εξαγωγή δεδομένων σε διάφορες μορφές όπως αρχεία Excel, Word, HTML, XML, κτλ.
- ιβ. Δυνατότητες αναβάθμισης.** Εάν μια επιχείρηση ή ένας οργανισμός χρησιμοποιεί αρχεία MS Access για την μηχανογράφηση της, μπορεί εύκολα να την αναβαθμίσει σε πιο μεγάλες βάσεις όπως σε SQL Server σε συνδυασμό με ένα MS Access Data Project.

Αν και η MS Access είναι ένα χρήσιμο ΣΔΒΔ, ωστόσο, υπάρχουν ορισμένα **μειονεκτήματα** που αποτρέπουν κάποιον αναλυτή ή προγραμματιστή να χρησιμοποιήσει την Access για μεγάλες εφαρμογές βάσεων δεδομένων. Αυτά μπορούν να συνοψιστούν ως εξής:

- α. Περιορισμός στο μέγεθος του αρχείου.** Το μέγιστο όριο ενός αρχείου MS Access είναι 2 GB και ισχύει για όλες τις εκδόσεις της μετά το 2007, συμπεριλαμβάνοντας τις φόρμες, τα ερωτήματα, τις αναφορές και τις λειτουργικές μονάδες. Το όριο των 2GB είναι απαγορευτικό όταν η βάση χρησιμοποιείται από εφαρμογές πελάτη-εξυπηρετητή (client-server applications) την οποία χρησιμοποιούν πολλοί χρήστες.
- β. Περιορισμός σε συναρτήσεις και σκανδάλες.** Ως ΣΔΒΔ η MS Access δεν περιέχει σκανδάλες (triggers) και συναρτήσεις (functions) για προκαθορισμένες ενέργειες όπως τα αντίγραφα ασφαλείας. Ο προγραμματιστής υποχρεώνεται να γράφει ξε-

χωριστά πολύπλοκες συναρτήσεις που λειτουργούν σαν σκανδάλες ή να ενσωματώνει εξωτερικές βιβλιοθήκες που επιτελούν αυτούς τους σκοπούς.

- γ. Ταχύτητα πρόσβασης.** Τεχνικώς, η MS Access έχει όριο πρόσβασης 255 χρηστών σε ένα αρχείο βάσης δεδομένων. Σε περίπτωση όμως, που έχουν πρόσβαση στο αρχείο 10 χρήστες ταυτόχρονα, μειώνει δραστικά την ταχύτητα στην ανάκτηση των δεδομένων, ακόμα και όταν η βάση δεδομένων βρίσκεται σε περισσότερα από ένα αρχείο (split database).
- δ. Αποθήκευση δεδομένων σε ένα μόνο αρχείο.** Όλα τα δεδομένα της βάσης αποθηκεύονται σε μόνο αρχείο, το οποίο αρχείο μπορεί να «μετακινηθεί» ή να διαγραφεί χωρίς κάποιον ιδιαίτερο έλεγχο, προκαλώντας έτσι ένα μεγάλο κενό στην ασφάλεια των δεδομένων της εφαρμογής. Γι' αυτό το λόγο το αρχείο της βάσης δεδομένων θα πρέπει να βρίσκεται σε προστατευμένες τοποθεσίες του υπολογιστή ή του δικτύου.
- ε. Περιορισμοί στις αναβαθμίσεις και στη διαθεσιμότητα.** Η μετάπτωση μιας βάσης MS Access παλαιότερης έκδοσης σε νέα συστήματα προκαλεί προβλήματα στις αντιστοιχίσεις των δεδομένων στη νέα βάση, με αποτέλεσμα οι χρήστες να αναβαθμίζουν σε νεότερες εκδόσεις MS Access. Επιπλέον, η απόκτηση μιας έκδοσης MS Access δεν είναι δωρεάν αλλά διατίθεται στις εκδόσεις Professional, Business και Business Premium του MS Office και αφορά μόνο χρήστες Windows.

Τυπικά ένα ΣΔΒΔ MS Access **δεν προτιμάται** από τους προγραμματιστές και τους αναλυτές βάσεων δεδομένων όταν πρόκειται για:

- Βάσεις Δεδομένων που περιέχουν ευαίσθητα και ρυθμιζόμενα δεδομένα.
- Βάσεις δεδομένων που απαιτούν δικαιώματα αι ρόλους χρηστών.
- Βάσεις δεδομένων με μεγάλο αριθμό δεδομένων, ιδιαίτερα πολλούς σχεσιακούς πίνακες.
- Βάσεις δεδομένων που χρησιμοποιούνται από διαδικτυακές εφαρμογές (web-based database applications).
- Εφαρμογές ΒΔ με πολλούς ταυτόχρονους χρήστες.
- Αποθήκευση δεδομένων υψηλής απόδοσης ή μεγάλης κλίμακας.

Ωστόσο, η εφαρμογή και η βάση δεδομένων που θα υλοποιήσουμε στην MS Access με την παρούσα εργασία δεν περιέχεται στους παραπάνω περιορισμούς. Όπως αναφέρθηκε και σε

προηγούμενη παράγραφο πρόκειται για μια απλή και εύχρηστη εφαρμογή που καλύπτει ορισμένες ανάγκες μιας επιχείρησης ή γραφείου.

2.3.3 Το πακέτο εκτέλεσης Microsoft Access Runtime

Οι εφαρμογές που υλοποιούνται με την MS Access τρέχουν μόνο σε περίπτωση που η MS Access έχει εγκατασταθεί σε έναν Η/Υ με την προϋπόθεση ότι ο Η/Υ λειτουργεί με κάποια έκδοση Windows. Προκειμένου όμως οι χρήστες να εκτελούν μια εφαρμογή MS Access σε υπολογιστή που δεν έχει εγκαταστημένη την Access, η Microsoft διαθέτει στους προγραμματιστές το πακέτο εκτέλεσης Microsoft Access Runtime.

Το πακέτο αυτό εγκαθίσταται στον υπολογιστή που δεν έχει εγκατεστημένη μια έκδοση Access και η εφαρμογή εκτελείται, βάσει των προδιαγραφών του ΛΣ. Με αυτόν τον τρόπο επιτρέπεται στους προγραμματιστές να δημιουργούν βάσεις δεδομένων και να τις διανέμουν ελεύθερα σε απεριόριστο αριθμό τελικών χρηστών και είναι διαθέσιμο δωρεάν από την Microsoft για τις νεότερες εκδόσεις (από 2007 και έπειτα).

Τέλος επιτρέπει στους χρήστες να προβάλλουν, να επεξεργάζονται και να διαγράφουν δεδομένα, δηλαδή να «τρέχουν» την εφαρμογή, αλλά δεν επιτρέπει την αλλαγή στη σχεδίαση των αντικειμένων της MS Access (πίνακες, ερωτήματα, σχέσεις, φόρμες, εκθέσεις, μακροεντολές και λειτουργικές μονάδες).

2.4 Η γλώσσα προγραμματισμού Visual Basic for Applications (VBA)



Η γλώσσα προγραμματισμού «Visual Basic for Applications» (VBA) είναι μια έκδοση της γλώσσας προγραμματισμού Visual Basic που διατίθεται αποκλειστικά από την Microsoft, στις εφαρμογές του MS Office. Αν και η έκδοση Visual Basic 6.0 έπαψε να υπάρχει σαν γλώσσα, αφού ενσωματώθηκε πλήρως στη γλώσσα προγραμματισμού Visual Basic .NET (VB.NET), η VBA αναβαθμίστηκε το 2010 προσφέροντας ένα ολοκληρωμένο IDE στον προγραμματιστή.

Ουσιαστικά πρόκειται για μια «ελαφριά» έκδοση της VB έκδοση όσον αφορά τα χαρακτηριστικά λειτουργίας της και τον τρόπο διαχείρισης των εντολών και των συναρτήσεων. Η VBA δεν δημιουργήθηκε για να αντικαταστήσει την VB, αλλά για να προσφέρει τις δυνατότητες της VB στις εφαρμογές γραφείου του MS Office, συμπεριλαμβανομένου του χειρισμού λειτουργιών του περιβάλλοντος χρήστη, όπως τα μενού, οι γραμμές εργαλείων, οι φόρμες και τα παράθυρα διαλόγου. Με αυτόν τον τρόπο, για να εκτελεστεί μια εφαρμογή γραμμένη σε VBA βασική προϋπόθεση είναι η ύπαρξη μιας εφαρμογής Office, όπως η Access και το Excel.

Γενικώς, η VBA βοηθάει τον προγραμματιστή να προσαρμόζει τον υπάρχον λογισμικό στις εκάστοτε ανάγκες του. Εκτός από το MS Office και τις υπόλοιπες εφαρμογές της Microsoft, η VBA είναι ενσωματωμένη και σε άλλες εφαρμογές γραφείου και σχεδιαστικά προγράμματα, όπως το StarOffice, το AutoCad, το CorelDraw, το WordPerfect, το LibreOffice ενώ υποστηρίζεται πλήρως και στις εκδόσεις του MS Office για Mac OS.

2.4.1 Χαρακτηριστικά

Κύριο χαρακτηριστικό της VBA είναι ο προγραμματισμός αντικειμένων για κάθε εφαρμογή του Office βάσει των αρχών της VB. Τα αντικείμενα αυτά διαφέρουν σε κάθε εφαρμογή. Για παράδειγμα, στο Excel ως αντικείμενο ορίζεται ένα φύλλο εργασίας (Worksheet) ή ένα κελί (Cell) ενώ στην Access αντικείμενο είναι μια φόρμα ή ένα στοιχείο ελέγχου μιας φόρμας (κουμπί, λίστα, κτλ.). Συνοψίζοντας, τα κοινά γνωρίσματα της VBA για όλες τις εφαρμογές είναι τα εξής:

- α. Ο κώδικας (Code):** Η σύνταξη των κατάλληλων εντολών που αποθηκεύονται είτε σε λειτουργικές μονάδες (modules) είτε στα συμβάντα χρήσης (event) του κάθε αντικειμένου.
- β. Οι λειτουργικές μονάδες (Modules):** Πρόκειται για ομαδοποιημένα πακέτα εντολών, τα οποία περιέχουν διαδικασίες ή συναρτήσεις και αποθηκεύονται ξεχωριστά ως ένα αντικείμενο στο αρχείο Access.
- γ. Οι διαδικασίες (Procedures):** Πρόκειται ομάδα εντολών κώδικα, η οποία εκτελεί μία ή περισσότερες ενέργειες. Αναλόγως με την χρήση τους οι διαδικασίες χωρίζονται σε διαδικασίες τύπου «Sub» όταν πρόκειται για εκτέλεση λειτουργιών και σε συναρτήσεις (functions) όταν πρόκειται για τον υπολογισμό μιας συγκεκριμένης τιμής.

δ. Τα αντικείμενα (Objects): Σε ένα αρχείο Access τα κύρια αντικείμενα είναι:

- Πίνακες (tables)
- Ερωτήματα (Queries)
- Φόρμες (Forms)
- Αναφορές (Reports)
- Λειτουργικές Μονάδες (Modules)
- Μακροεντολές (Macros)

Κάθε ένα από τα παραπάνω, περιέχει επιπλέον αντικείμενα με την σειρά τους όπως ομάδες εγγραφών (Recordsets), Λίστες τιμών (Listboxes), Ετικέτες (Labels), Γραφήματα (Charts) κτλ. Σε όλα αυτά τα παραπάνω σημαντικότερο ρόλο στην εκτέλεση του προγράμματος διαδραματίζουν και οι μέθοδοι και οι ιδιότητες αυτών των αντικειμένων.

ε. Οι μεταβλητές και οι σταθερές (Variables / Constants): Τα αναγνωριστικά του κώδικα που δεσμεύουν μέρος της κύριας μνήμης για τις ανάγκες του προγράμματος.

Στη VBA, τα αντικείμενα είναι χρήσιμα για έναν προγραμματιστή καθώς μπορεί να αναφέρεται στις μεθόδους και τις ιδιότητες του είτε ξεχωριστά είτε ως μια ενιαία κλάση, δίνοντας του έτσι την ευελιξία να χειρίζεται τα αντικείμενα σύμφωνα με τις ανάγκες του.

Γενικά τα αντικείμενα σε ένα αρχείο Access έχουν όλα μοναδικές και ιδιότητες. Υπάρχουν όμως και ορισμένες ιδιότητες (όπως Name, Caption, κτλ.) και μέθοδοι (π.χ AddItem, Delete, Refresh, κτλ) που είναι κοινές σε ορισμένα αντικείμενα. Τέλος, η VBA προσφέρει στον προγραμματιστή την δυνατότητα να αναφέρεται σε ένα αντικείμενο με διαφορετικούς τρόπους και να χειρίζεται ένα αντικείμενο χωρίς να έχει επιλεγθεί.

2.4.2 Τύποι δεδομένων

Όπως σε όλες τις γλώσσες προγραμματισμού, έτσι και στην VBA, όταν θέλουμε να εκχωρήσουμε τιμές σε ένα αναγνωριστικό (μεταβλητή ή σταθερά), δηλώνουμε πάντα και τον αντίστοιχο τύπο της, δηλαδή τον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα στην μνήμη του υπολογιστή. Στον παρακάτω πίνακα αναφέρονται οι γνωστοί τύποι δεδομένων που χρησιμοποιούνται στη γλώσσα VBA:

Πίνακας 3: Οι υποστηριζόμενοι τύποι δεδομένων της VBA.

Τύπος Δεδομένων	Περιγραφή	Μέγεθος
Boolean	Λογική τιμή	2 bytes
Integer	Ακέραιος	2 bytes
Long	Μεγάλος Ακέραιος	4 bytes
Single	Πραγματικός απλής ακρίβειας	4 bytes
Double	Πραγματικός διπλής ακρίβειας	8 bytes
Currency	Νομισματική τιμή	8 bytes
String	Αλφαριθμητικό	1 byte ανά χαρακτήρα
Date	Ημερομηνία	8 bytes
Variant	Παραλλαγή	16 bytes για αριθμούς. 23 bytes για χαρακτήρες.

Όλες οι μεταβλητές δηλώνονται στην αρχή του προγράμματος ενώ σε περίπτωση που δε δηλωθεί τύπος μεταβλητής, τότε ορίζεται αυτομάτως ο τύπος «Variant» από τη VBA. Παράλληλα η εμβέλεια τους καθορίζεται, όπως σε όλα τις προγραμματιστικές τεχνικές, ως εξής:

- α.** Μεταβλητές **τοπικής εμβέλειας** που δηλώνονται με χρήση της εντολής **Dim** ή **Static** μέσα σε μία διαδικασία και έχουν εμβέλεια εντός αυτής.
- β.** Μεταβλητές **περιορισμένης εμβέλειας** που δηλώνονται με χρήση της εντολής **Dim** ή **Private** πριν την πρώτη διαδικασία μιας λειτουργικής μονάδας και έχουν εμβέλεια στα όρια της λειτουργικής μονάδας. Οποιοσδήποτε διαδικασίες εντός οποιασδήποτε άλλης λειτουργικής μονάδας δεν μπορούν να τις καλέσουν.
- γ.** Μεταβλητές **απεριόριστης εμβέλειας** που δηλώνονται με χρήση της **Public** και έχουν εμβέλεια σε όλες τις λειτουργικές μονάδες.

2.4.3 Διαδικασίες (Procedures)

Μια διαδικασία (procedure) είναι στην ουσία μια ομάδα εντολών κώδικα οι οποίες εκτελούν μια συγκεκριμένη λειτουργία όπως την ενημέρωση ενός πίνακα, προσθήκη εγγραφών, εξαγωγή αποτελεσμάτων, κτλ, η οποία χρησιμοποιείται συχνά από τον χρήστη. Στην VBA μια διαδικασία αναφέρεται και ως υπο-ρουτίνα (sub-routine) και βρίσκεται στο τμήμα σύνταξης μιας φόρμας ή αναφοράς ή να αποτελεί μέρος μιας λειτουργικής μονάδας. Η δήλωση μιας διαδικασίας γίνεται ως εξής:

(Public ή Static ή Private) Sub όνομα_διαδικασίας (λίστα παραμέτρων)

Σώμα Εντολών

End Sub

Οι χαρακτηρισμοί «Private», «Public» και «Static» είναι προαιρετικοί και υποδηλώνουν την εμβέλεια χρήσης της διαδικασίας:

- **Private (Ιδιωτική):** δηλώνει πως η διαδικασία είναι προσπελάσιμη από όλες διαδικασίες που ανήκουν στην ίδια λειτουργική μονάδα.
- **Public (Δημόσια):** δηλώνει ότι η διαδικασία είναι προσπελάσιμη από όλες τις διαδικασίες που ανήκουν στο ίδιο αντικείμενο (φόρμα/αναφορά).
- **Static (Στατική):** δηλώνει ότι οι τιμές των μεταβλητών της διαδικασίας διατηρούνται όταν ολοκληρωθεί η διαδικασία.

Η χρήση παραμέτρων σε μια διαδικασία είναι προαιρετική. Ωστόσο, εάν χρησιμοποιηθούν παράμετροι σε μια διαδικασία τότε αυτές μπορεί να είναι μεταβλητές, σταθερές, πίνακες ή αντικείμενα. Η κλήση μιας διαδικασίας γίνεται μέσω της εντολής **Call**.

Γενικά, η γλώσσα VBA δεν μπορεί να θεωρηθεί απολύτως αντικειμενοστραφής γλώσσα προγραμματισμού καθώς δεν υποστηρίζει την αρχή της κληρονομικότητας στις κλάσεις της. Ωστόσο, ακολουθεί τους κανόνες της αφαιρετικότητας, της ενθυλάκωσης και του πολυμορφισμού στη διαχείριση των αντικείμενων και ασφαλώς την παραδοσιακές τεχνικές του τμηματικού προγραμματισμού.

Ειδικότερα, όταν σχεδιάζουμε μια εφαρμογή που διαχειρίζεται δεδομένα είναι προτιμότερο να υποδιαιρούμε την εφαρμογή (το κύριο πρόγραμμα) σε μικρότερα υποπρογράμματα για μεγαλύτερη διευκόλυνση στην ανάπτυξη του κώδικα του προγράμματος και στη γρήγορη και ευκολότερη διόρθωση του κώδικα όταν αυτό κρίνεται απαραίτητο. Γι' αυτό ακριβώς το λόγο, όταν πρόκειται να κάνουμε οποιαδήποτε αλλαγή μέσα στο αρχείο της βάσης, το σωστότερο είναι να γίνεται με την χρήση των διαδικασιών.

2.4.4 Συναρτήσεις (Functions)

Μια συνάρτηση (function) της VBA αποτελεί και αυτή μια διαδικασία, αλλά επιστρέφει μία μόνο τιμή και λειτουργεί όπως και οι ενσωματωμένες συναρτήσεις της VBA (πχ. Sum, Avg, Nz, κτλ.). Ακόμα, χρησιμοποιούνται ως μέρος μίας έκφρασης σε μία διαδικασία VBA επεκτείνοντας έτσι τις λειτουργίες της και βοηθούν τον προγραμματιστή να συντάξει τις δικές του συναρτήσεις που ταιριάζουν στο πρόγραμμα του. Μια συνάρτηση μπορεί να είναι μέρος μιας λειτουργικής μονάδας ή ως ξεχωριστό τμήμα εντολών και συντάσσεται ως εξής:

(Private ή Public ή Static) Function όνομα_συνάρτησης (λίστα παραμέτρων) **AS** τύπος_συνάρτησης

Εντολές

όνομα_συνάρτησης = έκφραση ή μεταβλητή ή τιμή

End Function

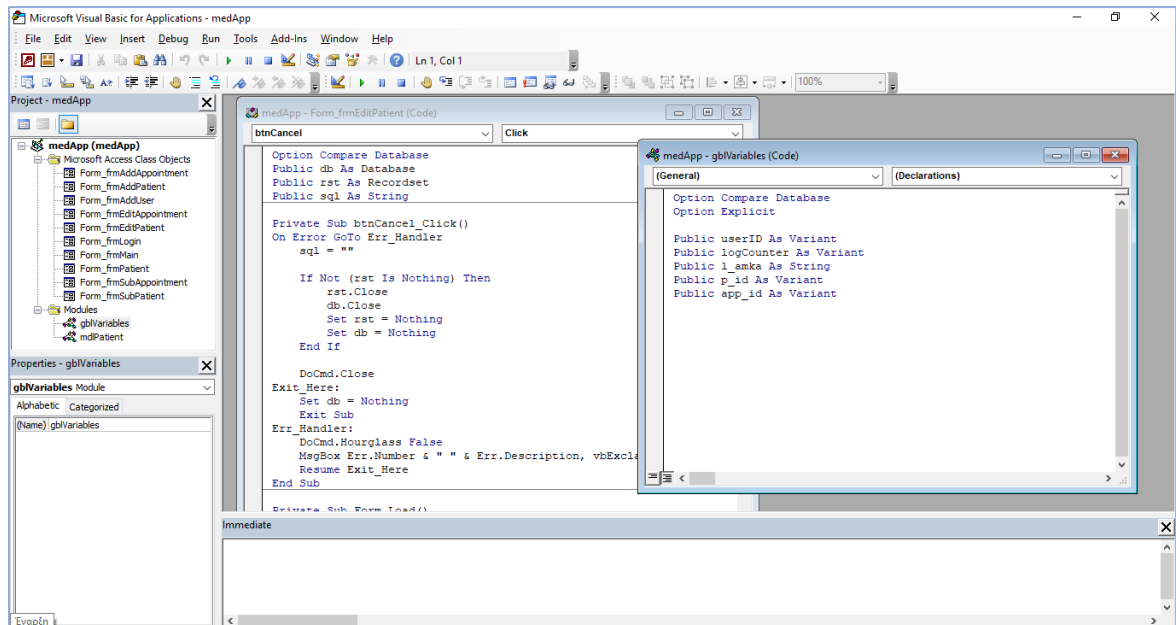
Όπως και στις διαδικασίες οι χαρακτηρισμοί «Private», «Public» και «Static» είναι προαιρετικοί και υποδηλώνουν την εμβέλεια χρήσης της συνάρτησης. Σε περίπτωση που δεν δοθεί χαρακτηρισμός εμβέλειας τότε η συνάρτηση θεωρείται Public.

Στο σώμα εντολών πρέπει να υπάρχει μια εντολή η οποία εκχωρεί μια τιμή στο όνομα της συνάρτησης τουλάχιστον μια φορά. Η τιμή αυτή μπορεί να προκύψει είτε από μια έκφραση είτε από μια άλλη μεταβλητή. Ανάλογα με την τιμή που επιστρέφει η συνάρτηση, δηλώνεται και ο τύπος της στην επικεφαλίδα της μετά την εντολή **AS** (Integer, Double, Boolean, κτλ.). Τέλος, μια συνάρτηση θα πρέπει να έχει τουλάχιστον μία παράμετρο (μεταβλητή, σταθερά, πίνακα, αντικείμενο).

2.4.5 Το περιβάλλον ανάπτυξης VBA

Το περιβάλλον ανάπτυξης (IDE) της VBA είναι ενσωματωμένο στην Access και ενεργοποιείται είτε μέσω των συμβάντων των αντικειμένων είτε μέσω της καρτέλας «Δημιουργία» → «Μακροεντολές & Κώδικας» → Visual Basic (Alt+F11). Αποτελείται από τα εξής στοιχεία:

- α. **Γραμμές μενού και εργαλείων**, όπου υπάρχουν τα βασικά κουμπιά ενεργειών επεξεργασίας εντολών, αποσφαλμάτωσης και μεταγλώττισης κώδικα, κτλ.
- β. **Η λίστα αντικειμένων (Project Explorer)**, η οποία εμφανίζει ιεραρχικά τη λίστα των αντικειμένων που περιέχονται και αναφέρονται στο πρόγραμμα.
- γ. **Το παράθυρο Ιδιοτήτων (Properties Window)**, που παραθέτει τις ιδιότητες σχεδίασης των επιλεγμένων αντικειμένων και τις τρέχουσες ρυθμίσεις.
- δ. **Το Παράθυρο κώδικα (Code Window)**, όπου ο προγραμματιστής γράφει, εμφανίζει και επεξεργάζεται τον κώδικα της VBA, από πολλές λειτουργικές μονάδες.
- ε. **Το Παράθυρο Άμεσων Μηνυμάτων (Immediate Window)**, το οποίο εμφανίζει μηνύματα που προκύπτουν από τον εντοπισμό σφαλμάτων στον κώδικά. Η εμφάνιση αυτών των μηνυμάτων γίνεται με την προσθήκη της εντολής **DEBUG PRINT** όνομα_μεταβλητής.



Εικόνα 9: Το IDE της VBA όπως είναι ενσωματωμένο στην MS Access από την έκδοση 2010 και μετά. Διακρίνεται το παράθυρο του κώδικα, το παράθυρο άμεσων μηνυμάτων και το παράθυρο με τη λίστα των αντικειμένων.

Το IDE της VBA μπορεί να είναι ανοικτό κατά την διάρκεια σχεδίασης μιας φόρμας ή αναφοράς ή όταν σχεδιάζεται ένας πίνακας στην Access, προσφέροντας έτσι ευελιξία στον προγραμματιστή όταν σχεδιάζει μια φόρμα και ελέγχει για τυχόν σφάλματα στο πρόγραμμα.

Τέλος, όπως αναφέραμε και πιο πάνω, το IDE της VBA δεν πωλείται ως ξεχωριστή εφαρμογή ανάπτυξης λογισμικού αλλά έρχεται ενσωματωμένο με την έκδοση της Access και την εκάστοτε έκδοση του Office. Οι εφαρμογές που υλοποιούνται μέσω της VBA έρχονται να καλύψουν επιπλέον ανάγκες στις εφαρμογές που προσφέρουν οι εκδόσεις του MS Office και δεν αποτελούν υψηλού επιπέδου εφαρμογές. Για την ανάπτυξη υψηλού επιπέδου εφαρμογών, η Microsoft διαθέτει το Visual Studio, το οποίο υποστηρίζει όλες τις γνωστές γλώσσες προγραμματισμού (C/C++, VB.Net, C#) και προσφέρει ένα πλήρες και αυτόνομο προγραμματιστικό περιβάλλον στον χρήστη.

3

Σχεδιασμός της Βάσης Δεδομένων

Κεφάλαιο 3: Σχεδιασμός της Βάσης Δεδομένων

3.1 Βήματα υλοποίησης της ΒΔ

Μια βάση δεδομένων έχει τις ακόλουθες υπονοούμενες ιδιότητες:

- α. Αναπαριστά μια άποψη του πραγματικού κόσμου που τις περισσότερες φορές αναφέρεται ως μικρόκοσμος (mini-world).
- β. Αποτελεί μια λογικά συνεκτική συλλογή δεδομένων που έχει κάποια εγγενή σημασία. οποιαδήποτε τυχαία διευθέτηση δεδομένων δεν μπορεί να θεωρηθεί ως βάση δεδομένων.
- γ. Σχεδιάζεται, χτίζεται και γεμίζει με δεδομένα για κάποιο συγκεκριμένο σκοπό. Προορίζεται για μια συγκεκριμένη ομάδα χρηστών και για κάποιες προκαθορισμένες εφαρμογές.

Βέβαια, ένα πρόβλημα που παρουσιάζεται συνήθως κατά την διάρκεια του σχεδιασμού μιας βάσης δεδομένων, αφορά τον σχεδιασμό της λογικής και φυσικής δομής της, για την κάλυψη των πληροφοριακών αναγκών του/των χρήστη/χρηστών της εφαρμογής. Γι' αυτό όταν σχεδιάζουμε μια βάση δεδομένων πρέπει απαραίτητα:

- α. Να ικανοποιηθούν οι απαιτήσεις των καθορισμένων χρηστών ως προς το περιεχόμενο των πληροφοριών.
- β. Να παρέχεται μια φυσική και κατανοητή δόμηση των πληροφοριών για τους χρήστες.
- γ. Να υποστηρίζονται οι απαιτήσεις σε επεξεργασία δεδομένων και όλοι οι στόχοι απόδοσης μιας ΒΔ όπως ο χρόνος απόκρισης, ο χρόνος επεξεργασίας και ο αποθηκευτικός χώρος της.

Ως ένα πολύπλοκο σύστημα μπορεί να αναπαρασταθεί με διαφορετικούς συνδυασμούς, συνεπώς και η ανάπτυξη μιας εφαρμογής βάσης δεδομένων απαιτεί έναν σωστό σχεδιασμό, ο οποίος προϋποθέτει τα παρακάτω βήματα:

1. **Καταγραφή και ανάλυση απαιτήσεων**, όπου προσδιορίζουμε το αρχικό πρόβλημα που θα επιλυθεί μέσω της εφαρμογής. Οι απαιτήσεις αυτές πρέπει να είναι τεκμηριωμένες με τέτοιο τρόπο ώστε να είναι αντιληπτές από αυτούς που θα εμπλακούν στην ανάπτυξη της. Με την ολοκλήρωση αυτής της διαδικασίας ξεκινάει ο σχεδιασμός της βάσης δεδομένων.

2. **Εννοιολογικός Σχεδιασμός**, όπου περιλαμβάνει την περιγραφή των οντοτήτων της βάσης δεδομένων και τις συσχετίσεις μεταξύ τους. Στο στάδιο αυτό περιλαμβάνεται η διαγραμματική αναπαράσταση των συσχετίσεων μεταξύ των οντοτήτων που είναι το Διάγραμμα Οντοτήτων Συσχετίσεων, ο προσδιορισμός των οντοτήτων και των συσχετίσεων παράλληλα με τις ιδιότητες και τους τύπους τους και ο καθορισμός των πρωτευόντων κλειδιών τους.
3. **Λογικός Σχεδιασμός**, όπου γίνεται η καταγραφή του Σχεσιακού Μοντέλου που προκύπτει από την μεταφορά του διαγράμματος ΟΣ. Στο στάδιο αυτό γίνεται η εφαρμογή των μεθόδων της κανονικοποίησης και ο έλεγχος της ακεραιότητας και του πλεονασμού των δεδομένων. Τέλος σχεδιάζεται το Σχεσιακό Μοντέλο Οντοτήτων.
4. **Φυσικός Σχεδιασμός**, που αφορά την φυσική οργάνωση της βάσης, δηλαδή τους πίνακες, τα ερωτήματα, ο τρόπος προσπέλασης των δεδομένων από τους χρήστες και ο τρόπος αποθήκευσης στον ηλεκτρονικό υπολογιστή.

3.2 Καταγραφή Απαιτήσεων

Η βάση δεδομένων που θα σχεδιάσουμε στην MS Access θα είναι απλή και πρακτική στη διαχείριση της. Σύμφωνα με την μελέτη περίπτωσης που περιγράψαμε στο προηγούμενο κεφάλαιο (παράγραφος 1.3), η εφαρμογή της βάσης δεδομένων που θα υλοποιήσουμε πρέπει να καλύπτει τις παρακάτω προδιαγραφές:

1. Για έναν ασθενή, ο ιατρός (χρήστης) αποθηκεύει τον ΑΜΚΑ, το ονοματεπώνυμο του, την ημερομηνία γέννησης του, το φύλο του, την διεύθυνση κατοικίας του, τα τηλέφωνα επικοινωνίας του και έναν χαρακτηριστικό κωδικό αριθμό εγγραφής.
2. Για κάθε ραντεβού, ο ιατρός αποθηκεύει την ημέρα, την ώρα, την αιτιολογία επίσκεψης και σημειώσεις σχετικά με τον λόγο επισκέψεως του ασθενούς ενώ παράλληλα αποθηκεύεται και ένας ξεχωριστός αριθμός που χαρακτηρίζει το ραντεβού.
3. Ένας ασθενής μπορεί να κλείνει πολλά ραντεβού στο ιατρείο. Τα ραντεβού αυτά μπορεί να είναι:
 - Ένα ή περισσότερα σε μια μέρα αλλά σε διαφορετικές ώρες.
 - Ένα ή περισσότερα σε διαφορετικές ημέρες και ώρες.

4. Ο ιατρός μπορεί και αποθηκεύει στην βάση δεδομένων το ωράριο λειτουργίας, δηλαδή τα χρονικά διαστήματα (ώρες ραντεβού) οι οποίες είναι διαθέσιμες για όλες τις ημέρες.

Προκειμένου να καλύψουμε και την ορθότητα χρήσης της εφαρμογής από την γραμματεία του ιατρού, η βάση δεδομένων θα αποθηκεύει δεδομένα για τους χρήστες της εφαρμογής. Ενώ η Access δεν υποστηρίζει διαχείριση χρηστών και ρόλους χρηστών, θα σχεδιαστεί κατάλληλος πίνακας που θα περιέχει τα ονόματα και τους τύπους των χρηστών της εφαρμογής και κατάλληλη διεπαφή χρήστη (φόρμα σύνδεσης) όπως θα παρουσιαστεί στο επόμενο κεφάλαιο. Οι χρήστες ανήκουν σε δυο ομάδες, τον **διαχειριστή-γιατρό (administrator)** και τον **χρήστη-γραμματεία (user)**.

Σύμφωνα με τα παραπάνω, οι ρόλοι που θα αποδώσουμε στην κάθε ομάδα χρηστών είναι οι εξής:

5. Ο διαχειριστής-γιατρός έχει πλήρη πρόσβαση στην εφαρμογή και μπορεί:
 - Να αποθηκεύει και να επεξεργάζεται τα στοιχεία των ασθενών και τα στοιχεία των ραντεβού του.
 - Να επεξεργάζεται το ωράριο εργασίας (χρονικά διαστήματα) που είναι διαθέσιμο για το ιατρείο.
 - Να έχει πρόσβαση σε αναφορές (εκθέσεις) σχετικά με τα ημερήσια και μηνιαία ραντεβού, με τις ηλικίες των επισκέψεων και τα ποσοστά επισκέψεων των ασθενών ανά φύλο.
 - Να δημιουργεί χρήστες που θα έχουν πρόσβαση στην εφαρμογή, καταχωρώντας το όνομα χρήστη, τον κωδικό πρόσβασης του και την κατηγορία του.
6. Η γραμματεία-χρήστης έχει περιορισμένη πρόσβαση στην εφαρμογή και μπορεί:
 - Να αποθηκεύει και να επεξεργάζεται τα στοιχεία των ασθενών και τα στοιχεία των ραντεβού του.
 - Να έχει πρόσβαση μόνο στις αναφορές (εκθέσεις) σχετικά με τα ημερήσια και μηνιαία ραντεβού.

Σύμφωνα με τα παραπάνω, θα προχωρήσουμε στο επόμενο βήμα που αφορά τον εννοιολογικό σχεδιασμό, το κύριο μέρος του σχεδιασμού της βάσης δεδομένων.

3.3 Εννοιολογικός Σχεδιασμός

Όπως αναφέραμε προηγουμένως, ο εννοιολογικός σχεδιασμός περιλαμβάνει την περιγραφή των βασικών οντοτήτων της βάσης και τις συσχετίσεις μεταξύ τους. Αυτό περιγράφεται μέσω του εννοιολογικού μοντέλου μέσω του διαγράμματος Οντοτήτων-Συσχετίσεων.

3.3.1 Εννοιολογικό Μοντέλο

Πρώτο βήμα στον σχεδιασμό μιας Βάσης Δεδομένων είναι η απεικόνιση των δεδομένων που περιλαμβάνονται στο σύστημα, σε οντότητες και σε σχέσεις (συσχετίσεις) που υπάρχουν μεταξύ των δεδομένων και είναι ανεξάρτητη από το λογισμικό διαχείρισης που θα χρησιμοποιηθεί. Οι οντότητες της βάσης μας αναφέρονται στον παρακάτω πίνακα:

Πίνακας 4: Οι οντότητες με τους αντίστοιχους πίνακες της ΒΔ.

Οντότητα	Πίνακας Βάσης	Περιγραφή
Ασθενής	Patient	Καταχωρούνται τα στοιχεία των ασθενών.
Ραντεβού	Appointment	Καταχωρούνται τα στοιχεία του ραντεβού
Ημέρες Εργασίας	AppDays	Καταχωρούνται οι ημερολογιακές ημέρες λειτουργίας του ιατρού.
Ωράριο	AppHours	Καταχωρούνται τα χρονικά διαστήματα (ώρες) των ραντεβού.
Χρήστες	Users	Καταχωρούνται τα στοιχεία των χρηστών.
Τύποι Χρηστών	UserTypes	Καταχωρούνται οι τύποι των χρηστών.

Στη συνέχεια αναφέρουμε τα χαρακτηριστικά της κάθε ιδιότητας και τις σχέσεις που υπάρχουν μεταξύ τους.

Οντότητα «Ασθενής»:

Πίνακας 5: Τα χαρακτηριστικά της οντότητας «Ασθενής».

Χαρακτηριστικό	Περιγραφή
A/A Ασθενή	Μοναδικός αριθμός, αναγνωριστικό ασθενή στη βάση
ΑΜΚΑ	Αριθμός Μητρώου Κοινωνικής Ασφάλισης του ασθενή
Επώνυμο	Το επώνυμο του ασθενή.
Όνομα	Το όνομα του ασθενή.
Ημερομηνία Γέννησης	Η ημερομηνία γέννησης του ασθενή.
Φύλο	Το φύλο του ασθενή.
Διεύθυνση	Η διεύθυνση κατοικίας του ασθενή.

Πόλη	Η πόλη κατοικίας του ασθενή.
Τηλέφωνο	Το τηλέφωνο επικοινωνίας
Σημειώσεις	Σημειώσεις σχετικά με τον ασθενή.
Ημ/νία Εγγραφής	Ημερομηνία εγγραφής του ασθενή στη ΒΔ.
Ημ/νία Επεξεργασίας	Ημερομηνία τελευταίας ενημέρωσης στοιχείων.

Οντότητα «Ραντεβού»:**Πίνακας 6:** Τα χαρακτηριστικά της οντότητας «Ραντεβού».

Χαρακτηριστικό	Περιγραφή
A/A Ραντεβού	Μοναδικός αριθμός ραντεβού στη βάση.
Ημερομηνία	Ημέρα κράτησης ραντεβού.
Αιτιολογία	Σκοπός επισκέψεως του ασθενή.
Παρατηρήσεις	Σημειώσεις σχετικά με την επίσκεψη του ασθενή.
Ημ/νία Εγγραφής	Ημερομηνία εγγραφής του ραντεβού στη ΒΔ.
Ημ/νία Επεξεργασίας	Ημερομηνία τελευταίας ενημέρωσης ραντεβού.

Συσχέτιση «Ασθενής» - «Ραντεβού»:

«Κλείνει»: μέσω αυτής της συσχέτισης δηλώνεται ότι ένας ασθενής κλείνει ένα (1) ή περισσότερα ραντεβού (N) στο ιατρείο, οποιαδήποτε μέρα και ώρα. Ωστόσο ένα ραντεβού μιας συγκεκριμένης ημέρας και ώρας αντιστοιχίζεται σε έναν μόνο ασθενή. Επομένως ο λόγος πληθικότητας είναι **1-προς-πολλά (1:N)**, με την οντότητα «Ασθενής» να συμμετέχει πλήρως στη συσχέτιση με το «Ραντεβού».

Οντότητα «Ημέρες Εργασίας»**Πίνακας 7:** Το χαρακτηριστικό της οντότητας «Ημέρες Εργασίας».

Χαρακτηριστικό	Περιγραφή
Ημερομηνία	Μοναδική ημέρα εργασίας/λειτουργίας του ιατρείου.

Συσχέτιση «Ημέρα Εργασίας» - «Ραντεβού»:

«Περιέχει»: μέσω αυτής της συσχέτισης δηλώνεται μια εργασιακή ημέρα μπορεί και περιέχει από ένα (1) και περισσότερα ραντεβού (N), αλλά ένα ραντεβού (1) κλείνεται αποκλειστικά σε μια και μόνο μέρα (1). Επομένως ο λόγος πληθικότητας μεταξύ των δύο οντοτήτων είναι **1-προς-πολλά (1:N)**.

Οντότητα «Ωράριο»**Πίνακας 8:** Το χαρακτηριστικό της οντότητας «Ωρες Λειτουργίας».

Χαρακτηριστικό	Περιγραφή
Ωρες Λειτουργίας	Μοναδικές ώρες των ραντεβού του ιατρού.

Συσχέτιση «Ωρες Λειτουργίας» - «Ραντεβού»:

«Ανήκει»: μέσω αυτής της συσχέτισης δηλώνεται πως σε μια ημέρα ένας ασθενής μπορεί να κλείσει μία ή περισσότερες ώρες για ραντεβού (M), ενώ μια ώρα ραντεβού μπορεί να ανήκει σε μία ή περισσότερες ημέρες ραντεβού (N) ενός ασθενή. Επομένως ο λόγος πληθικότητας σε αυτή τη συσχέτιση είναι **πολλά-προς-πολλά (M:N)**.

Σύμφωνα με την παραπάνω συσχέτιση προκύπτει η ασθενής οντότητα «**ΩραΡαντεβού**» η δεν διαθέτει δικά της αναγνωριστικά αλλά όπως θα μελετήσουμε σε επόμενη παράγραφο είναι μια σχεσιακή οντότητα, καθώς περιέχει τον μοναδικό αριθμό ενός ραντεβού και την ώρα λειτουργίας του ιατρού.

Οντότητα «Χρήστης»:**Πίνακας 9:** Τα χαρακτηριστικά της οντότητας «Χρήστης».

Χαρακτηριστικό	Περιγραφή
A/A Χρήστη	Μοναδικός αριθμός χρήστη.
Όνομα Χρήστη	Το username του χρήστη.
Κωδικός Πρόσβασης	Το password του χρήστη.
Τύπος Χρήστη	Ο τύπος του χρήστη.
Ημ/νία Δημιουργίας	Ημερομηνία εγγραφής του χρήστη στη ΒΔ.
Ημ/νία Επεξεργασίας	Ημερομηνία τελευταίας ενημέρωσης του χρήστη.

Οντότητα «Τύπος Χρήστη»:**Πίνακας 10:** Τα χαρακτηριστικά της οντότητας «Τύπος Χρήστη».

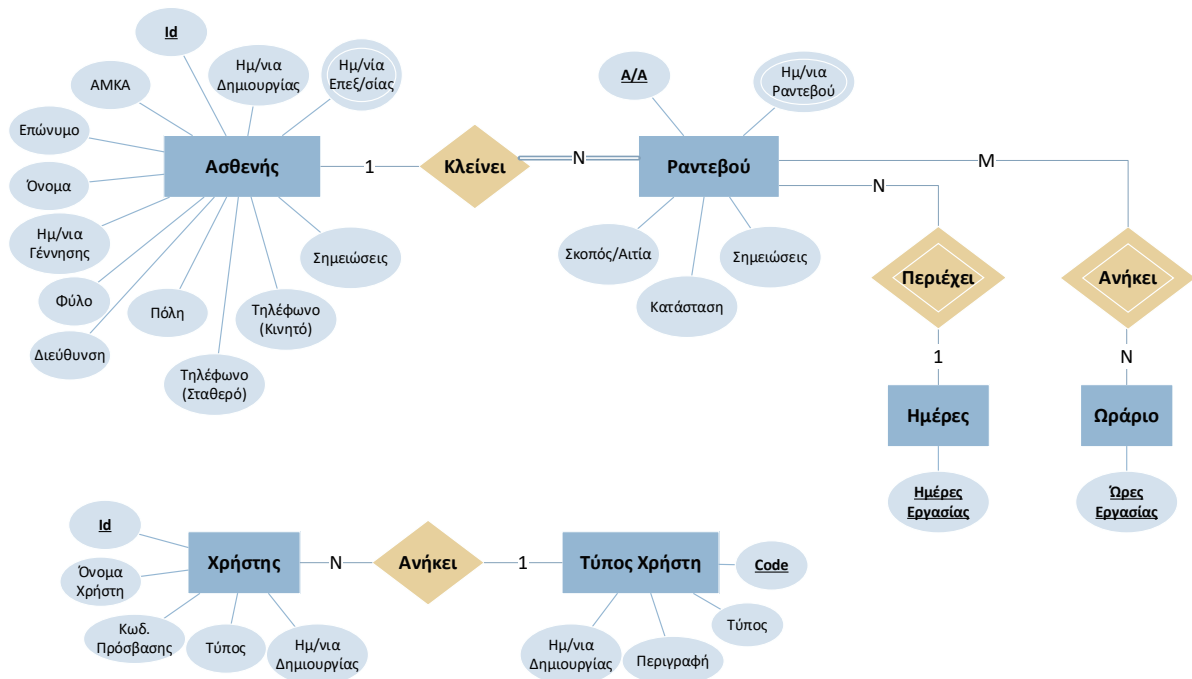
Χαρακτηριστικό	Περιγραφή
Κωδικός Τύπου Χρήστη	Μοναδικός κωδικός τύπου χρήστη.
Τίτλος	Τίτλος τύπου χρήστη.
Περιγραφή	Σύντομη περιγραφή του τύπου χρήστη.
Ημ/νία Δημιουργίας	Ημερομηνία εγγραφής στη ΒΔ.

Συσχέτιση «Τύπος Χρήστη» - «Χρήστης»:

«Ανήκει»: μέσω αυτής της συσχέτισης δηλώνεται ένας τύπος χρήστη (1) μπορεί και ανήκει σε πολλούς χρήστες (N), αλλά ένας χρήστης διαθέτει μόνο έναν τύπο χρήστη. Επομένως ο λόγος πληθικότητας σε αυτή τη συσχέτιση είναι **ένα-προς-πολλά (1:N)**.

3.3.2 Διάγραμμα Οντοτήτων – Συσχετίσεων

Έχοντας λοιπόν αναλύσει τις οντότητες της βάσης δεδομένων που δημιουργήσαμε, σχεδιάσαμε το Διάγραμμα Οντοτήτων-Συσχετίσεων, όπου φαίνονται οι συσχετίσεις μεταξύ των οντοτήτων καθώς και οι λόγοι πληθικότητας αυτών.



Εικόνα 10: Το διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣ) της βάσης δεδομένων.

Με την ολοκλήρωση του εννοιολογικού σχεδιασμού, περνάμε στην επόμενη φάση που αφορά τον λογικό σχεδιασμό της βάσης μας με τους κατάλληλους πίνακες και τα γνωρίσματα τους.

3.4 Λογικός Σχεδιασμός

Ο λογικός σχεδιασμός της βάσης δεδομένων περιλαμβάνει το σχεσιακό μοντέλο, δηλαδή την αντιστοίχιση των οντοτήτων του εννοιολογικού μοντέλου με τους κατάλληλους σχεσιακούς πίνακες της βάσης.

3.4.1 Το Σχεσιακό Μοντέλο

Γενικά το σχεσιακό μοντέλο αποτελείται από πίνακες που ο κάθε ένας έχει ένα μοναδικό όνομα στη ΒΔ, αποτελούμενος από ένα σύνολο γραμμών και στηλών. Κάθε γραμμή του πίνακα αποτελεί μια εγγραφή δεδομένων, ενώ κάθε στήλη προσδιορίζει τις ιδιότητες αυτής της εγγραφής. Επίσης σε έναν πίνακα, η κάθε εγγραφή είναι ενός δεδομένου τύπου και ο τύπος της εγγραφής αυτής ορίζει έναν σταθερό αριθμό πεδίων ή ιδιοτήτων. Επίσης, σε έναν πίνακα εκτός από τις εγγραφές που σχετίζονται άμεσα με τις ιδιότητες των οντοτήτων, υπάρχουν και εγγραφές που περιγράφουν τις συσχετίσεις μεταξύ των οντοτήτων. Σύμφωνα λοιπόν με αυτές τις παραδοχές, για την κάθε οντότητα παραθέτουμε παρακάτω τα ονόματα των πινάκων τα οποία προσδιορίζουν τις αντίστοιχες έννοιες με τα πεδία του.

Πίνακας 11: Ο πίνακας-οντότητα «Patient»

Πίνακας «Patient» (Οντότητα: «Ασθενής»)	
Πεδίο	Περιγραφή
Id	Πρωτεύον κλειδί του πίνακα και δηλώνει το αναγνωριστικό του κάθε ασθενή.
AMKA	Πεδίο που αποθηκεύεται ο ΑΜΚΑ του ασθενή. Δεν επιτρέπονται οι κενές εγγραφές και οι διπλότυπες τιμές.
LName	Πεδίο που αποθηκεύεται το επώνυμο του ασθενή.
FName	Πεδίο που αποθηκεύεται το επώνυμο του ασθενή.
DOB	Πεδίο που αποθηκεύεται η ημερομηνία γέννησης του ασθενή.
Gender	Πεδίο που αποθηκεύεται το φύλο του ασθενή.
Address	Πεδίο που αποθηκεύεται η διεύθυνση κατοικίας του ασθενή.
City	Πεδίο που αποθηκεύεται η πόλη κατοικίας του ασθενή.
PhoneA	Πεδίο που αποθηκεύεται το σταθερό τηλέφωνο επικοινωνίας.
PhoneB	Πεδίο που αποθηκεύεται το κινητό τηλέφωνο επικοινωνίας.
Notes	Πεδίο που αποθηκεύονται σημειώσεις σχετικά με τον ασθενή.
CreationDate	Πεδίο που αποθηκεύει την ημερομηνία εγγραφής του ασθενή στη ΒΔ (timestamp).
UpdDate	Πεδίο που αποθηκεύει την ημερομηνία της τελευταίας ενημέρωσης

	της εγγραφής του ασθενή στη ΒΔ (timestamp).
--	---

Πίνακας 12: Ο πίνακας-οντότητα «Appointment».

Πίνακας «Appointment» (Οντότητα: «Ραντεβού»)	
Πεδίο	Περιγραφή
Id	Πρωτεύον κλειδί του πίνακα και δηλώνει το αναγνωριστικό της εγγραφής του ραντεβού στον πίνακα.
AppDate	Πεδίο που αποθηκεύεται η ημερομηνία που κλείνει ο ασθενής το ραντεβού. Επιτρέπονται διπλότυπες εγγραφές σε επίπεδο ημέρας.
Subject	Πεδίο που αποθηκεύεται η αιτιολογία της επισκέψεως του ασθενή.
Notes	Πεδίο που αποθηκεύονται σημειώσεις σχετικά με το ραντεβού.
CrDate	Πεδίο που αποθηκεύει την ημερομηνία εγγραφής του ασθενή στη ΒΔ (timestamp).
Pid	Ξένο κλειδί του πίνακα που υποδηλώνει τη σχέση-σύνδεση με τον πίνακα «Patient» περιέχοντας την αντίστοιχη τιμή του πεδίου «Id» (A/A ασθενή) του πίνακα «Patient».

Πίνακας 13: Ο πίνακας-οντότητα «AppDays».

Πίνακας «AppDays» (Οντότητα: «Ημέρες Εργασίας»)	
Πεδίο	Περιγραφή
AppDate	Πρωτεύον κλειδί και μοναδικό πεδίο του πίνακα. Δηλώνει τις ημέρες εργασίας του ιατρού και περιέχει τις ημέρες ενός έτους (1/1 – 31/12)

Πίνακας 14: Ο πίνακας-οντότητα «AppHours».

Πίνακας «AppHours» (Οντότητα: «Ωράριο»)	
Πεδίο	Περιγραφή
AppHour	Πρωτεύον κλειδί και μοναδικό πεδίο του πίνακα. Δηλώνει τις ώρες (χρονικά διαστήματα) που μπορεί να κλείσει ένας ασθενής ραντεβού στο ιατρείο.

Πίνακας 15: Ο πίνακας-οντότητα «AppHours» που αποτελεί σχεσιακή οντότητα

Σχεσιακός Πίνακας «AppDetails»	
Πεδίο	Περιγραφή
AppId	Ξένο κλειδί του πίνακα που περιέχει τον μοναδικό αριθμό (Id) του ραντεβού και υποδηλώνει τη σχέση του πίνακα «Appointment».
AppHour	Ξένο κλειδί του πίνακα που περιέχει την μοναδική ώρα που αντιστοιχεί στο ραντεβού που έκλεισε ο ασθενής και υποδηλώνει τη σχέση με τον πίνακα «AppHours».

Πίνακας 16: Ο πίνακας-οντότητα «Users».

Πίνακας «Users»	
Πεδίο	Περιγραφή
Id	Πρωτεύον κλειδί του πίνακα και δηλώνει το αναγνωριστικό της εγγραφής του χρήστη στον πίνακα.
Username	Πεδίο που περιέχει το όνομα του χρήστη.
Password	Πεδίο που περιέχει τον κωδικό πρόσβασης του χρήστη.
UserType	Ξένο κλειδί του πίνακα που περιέχει το αναγνωριστικό που υποδηλώνει τον τύπο του χρήστη και συσχετίζεται με τον πίνακα «UserTypes».
CreationDate	Πεδίο που περιέχει την ημερομηνία δημιουργίας στη ΒΔ.

Πίνακας 17: Ο πίνακας-οντότητα «UserTypes».

Πίνακας «UserTypes»	
Πεδίο	Περιγραφή
TypeCode	Πρωτεύον κλειδί του πίνακα και δηλώνει το αναγνωριστικό της εγγραφής του τύπου χρήστη στον πίνακα.
Type	Πεδίο που περιέχει τον τύπο του χρήστη.
Description	Πεδίο που περιέχει μια σύντομη περιγραφή του τύπου χρήστη.
CreationDate	Πεδίο που περιέχει την ημερομηνία δημιουργίας στη ΒΔ.

Παρατήρηση:

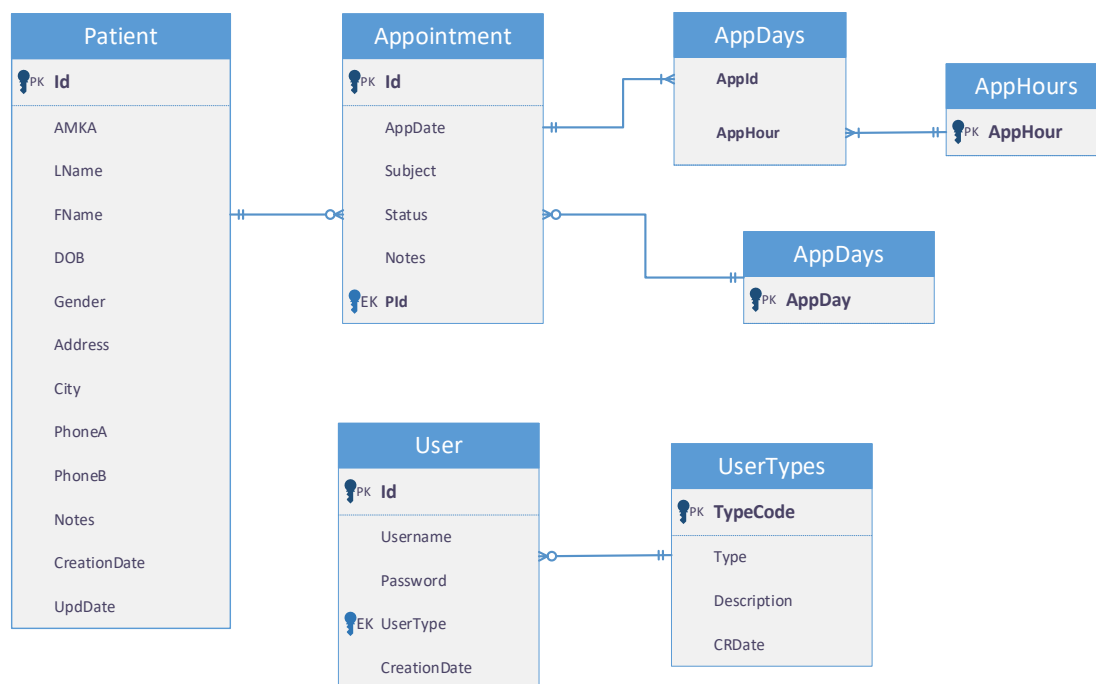
Σε όλους τους πίνακες, εκτός του σχεσιακού «AppDetails», έχουν προστεθεί πεδία ημερομηνιών (timestamp) όπου καταχωρούνται η ημερομηνία δημιουργίας και τελευταίας ενημέρωσης της εγγραφής του πίνακα. Αυτός είναι ένας πολύ καλός τρόπος να ελέγχουμε πότε δημιουργήθηκαν ή ενημερώθηκαν οι εγγραφές του πίνακα, βοηθώντας έτσι στην καλύτερη διαχείριση και τήρηση της ακεραιότητας των δεδομένων της ΒΔ.

3.4.2 Διάγραμμα Σχεσιακού Μοντέλου

Σε πολλά ΣΔΒΔ η σχεδίαση του διαγράμματος σχεσιακού μοντέλου γίνεται αυτόματα. Βέβαια αυτό τις περισσότερες φορές δεν είναι απόλυτα σωστό, γι' αυτό η σχεδίαση γίνεται πάντα από τον προγραμματιστή.

Αυτό βέβαια προϋποθέτει την κατασκευή των πινάκων, δηλαδή ο σχεδιασμός της ΒΔ να βρίσκεται στο «Φυσικό» επίπεδο, με τις ιδιότητες και τα χαρακτηριστικά των πεδίων, χωρίς να έχουν εισαχθεί δεδομένα στους πίνακες.

Για την δημιουργία του διαγράμματος του σχεσιακού μοντέλου χρησιμοποιήσαμε το λογισμικό Microsoft Visio 2019, και προκύπτει ως εξής:



Εικόνα 11: Το διάγραμμα του Σχεσιακού Μοντέλου της ΒΔ, σχεδιασμένο στο λογισμικό Microsoft Visio Professional 2019.

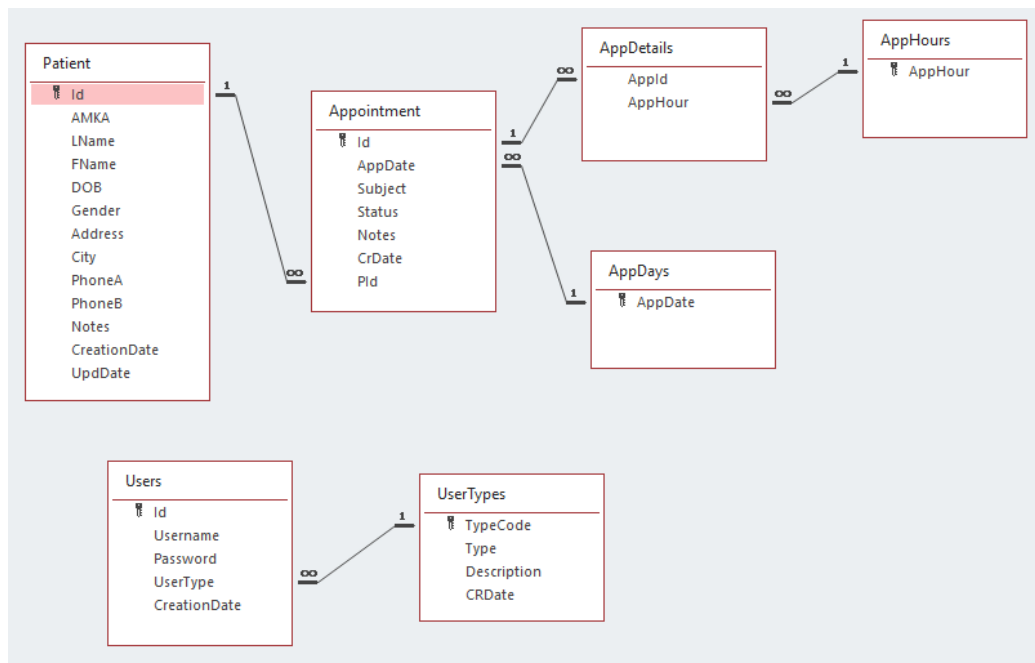
Ασφαλώς και η MS Access παρέχει την δυνατότητα σχεδίασης του διαγράμματος των σχέσεων των οντοτήτων. Αυτό γίνεται εύκολα από την καρτέλα «**Εργαλεία Βάσης Δεδομένων**» από το κουμπί «**Σχέσεις**».

Έπειτα επιλέγουμε, από την λίστα που μας εμφανίζεται, τους πίνακες που θέλουμε να συσχετίσουμε. Εφόσον τους επιλέξουμε, εμφανίζονται στο παράθυρο σχεδίασης με τα πεδία τους. Στη συνέχεια, για να σχεδιάσουμε την συσχέτιση μεταξύ δύο πινάκων, επιλέγουμε το πεδίο-πρωτεύον κλειδί του πίνακα και το αντιστοιχίζουμε στον πίνακα που το ίδιο πεδίο αποτελεί ξένο κλειδί και εμφανίζεται το παρακάτω «**Επεξεργασία Σχέσεων**»:

Εικόνα 12: Το παράθυρο διαλόγου «Επεξεργασία Σχέσεων».

Μέσω αυτού του παραθύρου ορίζουμε τον τύπο της σχέσης (ένα-προς-ένα, ένα-προς-πολλά, κτλ.) και τους σχετιζόμενους πίνακες με τα αντίστοιχα πεδία-κλειδιά. Πατώντας το κουμπί «**Δημιουργία**», δημιουργείται η συσχέτιση.

Εφαρμόζοντας την παραπάνω διαδικασία για όλους τους πίνακες προκύπτει το παρακάτω διάγραμμα.



Εικόνα 13: Το διάγραμμα του Σχεσιακού Μοντέλου όπως προκύπτει μέσα από το περιβάλλον της MS Access.

Στο παραπάνω σχήμα φαίνονται οι σχέσεις μεταξύ των πινάκων μέσω των γνωρισμάτων τους. Επίσης επισημαίνονται τα πεδία που είναι τα πρωτεύοντα κλειδιά με το σύμβολο του κλειδιού (κάθετα), ενώ το κλειδί-πεδίο αναφοράς έχει το χαρακτηριστικό σύμβολο του «∞» μπροστά από την γραμμή της συσχέτισης.

3.5 Φυσικός Σχεδιασμός

Όπως αναφέραμε και σε προηγούμενη παράγραφο, το στάδιο του φυσικού σχεδιασμού περιλαμβάνει την φυσική οργάνωση της βάσης στον ηλεκτρονικό υπολογιστή, δηλαδή τον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα στα αποθηκευτικά μέσα του συστήματος.

3.5.1 Το αρχείο της Βάσης Δεδομένων

Στην περίπτωση της εφαρμογής μας, θα υιοθετήσουμε την **μέθοδο διαχωρισμού** της ΒΔ σε δυο αρχεία MS Access: **α)** το αρχείο που θα περιέχει τους πίνακες και τα δεδομένα της εφαρμογής, δηλαδή το αρχείο «back-end» και **β)** το αρχείο «front-end» που περιέχει τα λειτουργικά στοιχεία της εφαρμογής, δηλαδή τις φόρμες, τις εκθέσεις, τα ερωτήματα και τις λειτουργικές μονάδες που χρησιμοποιούνται.

Η μέθοδος διαχωρισμού μιας βάσης δεδομένων βοηθάει στη βελτίωση της απόδοσής της και στη μείωση των πιθανοτήτων να καταστραφούν τα δεδομένα που βρίσκονται στο αρχείο της ΒΔ. Ειδικότερα ο διαχωρισμός αυτός προσφέρει:

- α. Βελτιωμένη αξιοπιστία**, καθώς μειώνεται ο κίνδυνος, λόγω μιας πιθανής αστοχίας του κώδικα να επηρεαστούν δραματικά τα δεδομένα της ΒΔ. Σε μια τέτοια περίπτωση που η MS Access καταρρεύσει, το πρόβλημα περιορίζεται μόνο στο λειτουργικό μέρος (front-end) της εφαρμογής ενώ η ΒΔ μπορεί να είναι διαθέσιμη.
- β. Επεκτασιμότητα εφαρμογής**, διότι διαχωρίζοντας μια βάση δεδομένων της MS Access, μπορούν να χρησιμοποιηθούν περισσότερα του ενός αρχεία «back-end» με διαφορετικά σύνολα πινάκων το καθένα και να είναι συνδεδεμένα στο ίδιο αρχείο «front-end». Έτσι μπορεί να ξεπεραστεί το όριο των 2GB που είναι διαθέσιμο στον χρήστη από την MS Access. Σε περίπτωση που ξεπεραστεί το όριο αυτό, μπορεί εύκολα να μετακινηθεί σε μια βάση SQL Server, ενώ το «front-end» να εξακολουθεί να βρίσκεται σε περιβάλλον Access.
- γ. Μέγιστη ασφάλεια των δεδομένων**, καθώς οι χρήστες δεν έχουν άμεση πρόσβαση στους πίνακες, υπάρχει μικρότερος κίνδυνος οι χρήστες να διαγράψουν κατά λάθος ολόκληρους πίνακες ή να υποκλέψουν δεδομένα. Η διεπαφή της εφαρμογής (front-end) είναι διαθέσιμη στους χρήστες μέσω του εκτελέσιμου αρχείου .accde και οι χρήστες δεν μπορούν να αλλάξουν κανένα από τα λειτουργικά αντικείμενα της εφαρμογής.

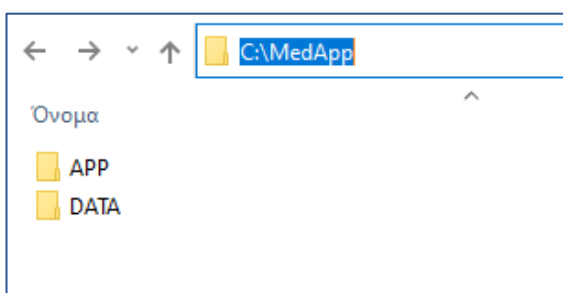
Το πρώτο βήμα για τον σχεδιασμό του φυσικού μοντέλου της βάσης θα είναι να φτιάξουμε την φυσική τοποθεσία της βάσης σύμφωνα με τα παρακάτω βήματα:

1. Στην τοποθεσία του τοπικού δίσκου **C:**, δημιουργούμε τον φάκελο «**MedApp**» και έπειτα μέσα σε αυτό το φάκελο, δημιουργούμε τους φακέλους «**DATA**» και «**APP**».

Σημείωση:

Επιλέγουμε την τοποθεσία C:\ για να είναι πιο εύκολη η πρόσβαση και η μετακίνηση των αρχείων σε περίπτωση αστοχίας ή μεταφοράς σε άλλον υπολογιστή.

2. Στον φάκελο «**DATA**» θα αποθηκευτεί το αρχείο της βάσης, ενώ στο φάκελο «**APP**» θα αποθηκευτεί το αρχείο της εφαρμογής (front-end).



Εικόνα 14: Ο φάκελος της εφαρμογής με τους καταλόγους DATA και APP που περιέχουν τα αρχεία της εφαρμογής.

3. Τρέχουμε από τον υπολογιστή μας την εφαρμογή MS Access 2019. Στη συνέχεια δημιουργούμε το αρχείο «**medAppDB.accdb**» και το αποθηκεύουμε στην τοποθεσία **C:\MedApp\DATA**.

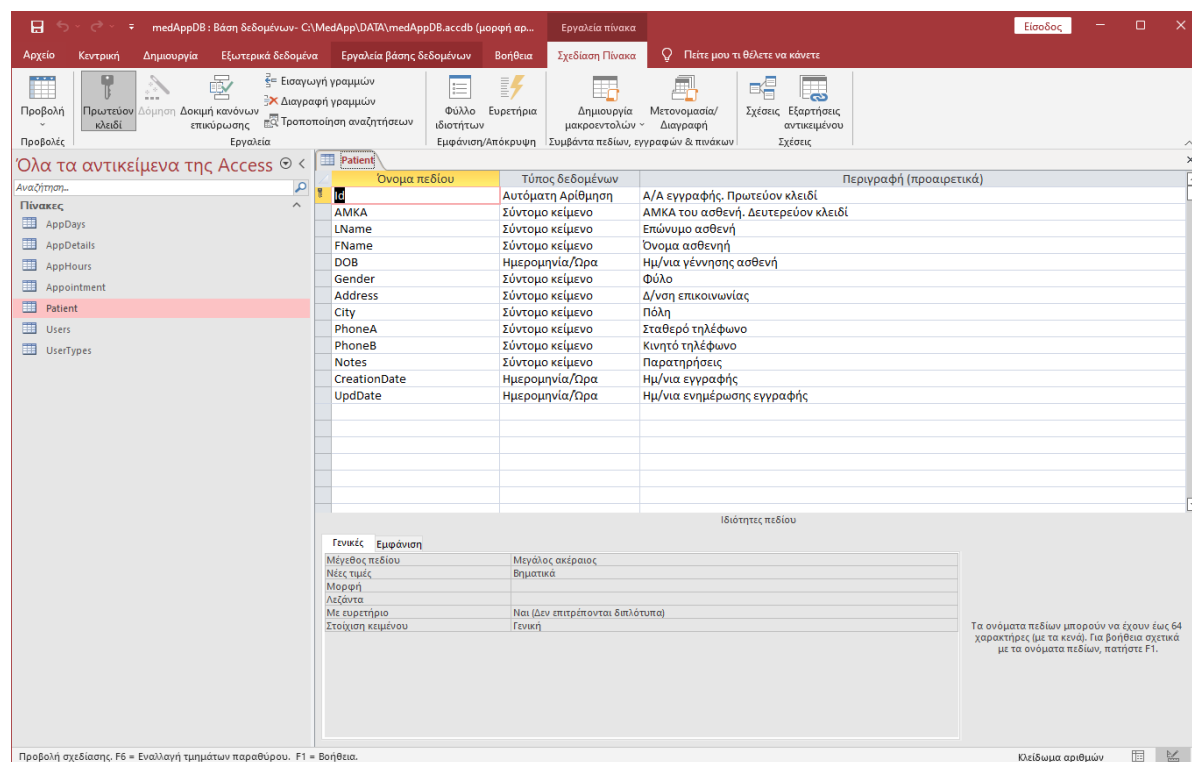


Εικόνα 15: Το εικονίδιο-αρχείο της εφαρμογής της ΒΔ.

Στη συνέχεια δημιουργούμε τους πίνακες της ΒΔ, βάσει του σχεσιακού μοντέλου που δημιουργήσαμε στην προηγούμενη παράγραφο ακολουθώντας τα εξής βήματα:

- Καρτέλα «**Δημιουργία**» → «**Σχεδίαση Πίνακα**».
- Δίνουμε ένα όνομα στον πίνακα και ορίζουμε τα **πεδία** και τις **ιδιότητες** τους.
- Μόλις ολοκληρώσουμε την δημιουργία των πεδίων αποθηκεύουμε τις αλλαγές και προχωράμε με τον επόμενο πίνακα.

3.5.2 Οι Πίνακες της ΒΔ



Εικόνα 16: Το αρχείο back-end που περιέχει τους πίνακες της ΒΔ.

Παρακάτω παρατίθενται αναλυτικά οι πίνακες που περιέχονται στο αρχείο της βάσης μας, με τους περιορισμούς και τους τύπους των πεδίων τους.

Πίνακας «Patient»

Πίνακας 18: Τα πεδία του πίνακα «Patient»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
Id	Αυτόματη αριθμηση	Μεγάλος Ακέραιος	Πρωτεύον κλειδί. Α/Α εγγραφής
AMKA	Κείμενο	11	ΑΜΚΑ. Δεν επιτρέπονται διπλότυπα.
LName	Κείμενο	100	Επώνυμο ασθενή.
FName	Κείμενο	100	Όνομα ασθενή.
DOB	Ημερομηνία/Ωρα	Σύντομη Ημερομηνία	Ημ/νια γέννησης ασθενή.
Gender	Κείμενο	1	Φύλο ασθενή («Α» ή «Γ»)
Address	Κείμενο	255	Δ/υση κατοικίας.
City	Κείμενο	100	Πόλη.
PhoneA	Κείμενο	10	Σταθερό τηλέφωνο.
PhoneB	Κείμενο	10	Κινητό τηλέφωνο.
Notes	Κείμενο	255	Παρατηρήσεις.

CreationDate	Ημερομηνία/Ωρα	Πλήρης Ημερομηνία	Ημ/νια εγγραφής.
UpdDate	Ημερομηνία/Ωρα	Πλήρης Ημερομηνία	Ημ/νία τελευταίας ενημέρωσης εγγραφής.

Πίνακας «Appointment»

Πίνακας 19: Τα πεδία του πίνακα «Appointment»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
Id	Αυτόματη αρίθμηση	Μεγάλος Ακέραιος	Πρωτεύον κλειδί. Α/Α επίσκεψης (ραντεβού)
AppDate	Ημερομηνία/Ωρα	Σύντομη ημερομηνία	Ημ/νία ραντεβού.
Subject	Κείμενο	100	Λόγος επίσκεψης.
Notes	Κείμενο	200	Παρατηρήσεις επίσκεψης.
CreationDate	Ημερομηνία/Ωρα	Πλήρης Ημερομηνία	Ημ/νια εγγραφής.
PId	Αριθμός	Μεγάλος Ακέραιος	Ξένο κλειδί. Α/Α ασθενή.

Πίνακας «AppDays»

Πίνακας 20: Τα πεδία του πίνακα «AppDays»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
AppDate	Ημερομηνία/Ωρα	Σύντομη ημερομηνία	Πρωτεύον κλειδί. Ημερολογιακή ημέρα εργασίας.

Πίνακας «AppHours»

Πίνακας 21: Τα πεδία του πίνακα «AppHours»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
AppHour	Ημερομηνία/Ωρα	Ώρα. Μορφή «ω:λλ» με κανόνα επικύρωσης των λεπτών για τιμές της μορφής ή 00:00 ή 30:00.	Πρωτεύον κλειδί. Ώρα επίσκεψης.

Πίνακας «AppDetails»

Πίνακας 22: Τα πεδία του πίνακα «AppDetails»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
AppId	Αριθμός	Μεγάλος Ακέραιος	Ξένο κλειδί. Α/Α επίσκεψης.
AppHour	Ημερομηνία/Ωρα	Ώρα της μορφής «ω:λλ:δδ»	Ξένο κλειδί. Ώρα επίσκεψης.

Πίνακας «Users»**Πίνακας 23:** Τα πεδία του πίνακα «Users»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
Id	Αυτόματη αριθμηση	Μεγάλος Ακέραιος	Πρωτεύον κλειδί. Α/Α εγγραφής
Username	Κείμενο	20	Username
Password	Κείμενο	8	Password. Περιέχει μάσκα εισαγωγής «****»
UserType	Κείμενο	2	Τύπος χρήστη.
CreationDate	Ημερομηνία/Ωρα	Πλήρης Ημερομηνία	Ημ/νία δημιουργίας.

Πίνακας «UserTypes»**Πίνακας 24:** Τα πεδία του πίνακα «UserTypes»

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
TypeCode	Κείμενο	2	Πρωτεύον κλειδί. Κωδικός τύπου χρήστη.
Type	Κείμενο	50	Τύπος χρήστη.
Description	Κείμενο	100	Περιγραφή.
CreationDate	Ημερομηνία/Ωρα	Γενική ημερομηνία	Ημ/νία εγγραφής

Παρατηρήσεις:

Όλα τα στοιχεία της ΒΔ που χρησιμοποιήσαμε, τα ονόματα, οι ΑΜΚΑ, οι διευθύνσεις, οι ημερομηνίες και γενικότερα τα στοιχεία των ασθενών αποτελούν **εικονικά** ονόματα και **δεν ανταποκρίνονται στην πραγματικότητα.**

Τα μοναδικά στοιχεία που ανταποκρίνονται στην πραγματικότητα είναι οι ημερολογιακοί πίνακες, δηλαδή οι τιμές των ημερών και των ωρών που λειτουργεί το ιατρείο.

4 Σχεδιασμός της Εφαρμογής

Κεφάλαιο 4: Σχεδιασμός της εφαρμογής

4.1 Βασικές αρχές σχεδίασης ενός γραφικού περιβάλλοντος.

Εφόσον σχεδιάσαμε την βάση δεδομένων με τους πίνακες και τα πεδία και ολοκληρώσαμε τις συσχετίσεις μεταξύ τους, επόμενο βήμα είναι η σχεδίαση του γραφικού περιβάλλοντος της εφαρμογής, δηλαδή του αρχείου «**front-end**» που περιέχει τις φόρμες και τις ειδικές λειτουργίες της εφαρμογής.

Γενικά οι φόρμες συνήθως έχουν τη μορφή οθονών εισόδου και εξόδου δεδομένων (παράθυρα διαλόγου ή διάλογοι) ενώ οι αναφορές έχουν τη μορφή οθονών εξόδου δεδομένων, οι οποίες συχνά προορίζονται αποκλειστικά για την εκτύπωση αυτών των δεδομένων. Γι' αυτό το λόγο, μια φόρμα ή μια έκθεση πρέπει να σχεδιάζεται βάσει των **απαιτήσεων** και των **δραστηριοτήτων** του χρήστη πάνω στην εφαρμογή.

Ακόμα, το γραφικό περιβάλλον μιας εφαρμογής πρέπει να είναι σχεδιασμένο έτσι ώστε να **διευκολύνει τον χρήστη** στην **εύρεση**, στην **προβολή** και στην **ανάκτηση** των διαφορετικών τμημάτων της εφαρμογής εύκολα και γρήγορα.

Αξίζει να σημειώσουμε πως ένα γραφικό περιβάλλον πρέπει να χαρακτηρίζεται από **συνέπεια** και **ευελιξία**, έτσι ώστε ο χρήστης να μπορεί να πλοηγείται ελεύθερα μεταξύ των πεδίων, τα δεδομένα να αποθηκεύονται όταν και εφόσον το ζητάει ο χρήστης και κάθε κουμπί ή εντολή να αντιστοιχίζεται αποκλειστικά και μόνο σε μια λειτουργία.

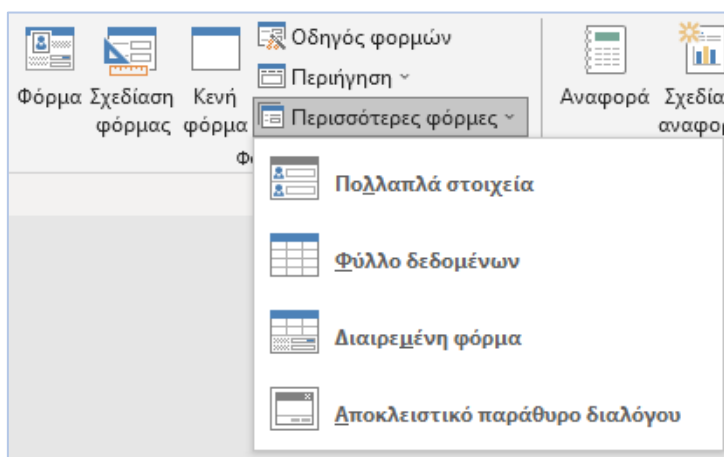
Τέλος, σε ένα γραφικό περιβάλλον, είναι πολύ σημαντική οι **πληροφορίες κατάστασης** της εφαρμογής που παρέχονται μέσω **μηνυμάτων προειδοποιήσεων** και **σφαλμάτων**. Με αυτό το τρόπο ο χρήστης διατηρείται ενημερωμένος σχετικά με τις αλλαγές που γίνονται στα δεδομένα της εφαρμογής μέσω των προειδοποιήσεων, οι οποίες πρέπει να είναι συγκεκριμένες, χωρίς ειδικές ορολογίες και να καθοδηγούν τον χρήστη σε μια συγκεκριμένη ενέργεια χωρίς επιπληκτικό τρόπο.

4.1.1 Σχεδίαση φορμών και αναφορών με την MS Access.

Η MS Access είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον, που δίνει τη δυνατότητα στο χρήστη να σχεδιάσει, παράλληλα με τους πίνακες της ΒΔ, τις φόρμες και τις αναφορές που προβάλλουν και διαχειρίζονται τα δεδομένα της ΒΔ. Οι φόρμες στην MS Access μπορούν να αναπαραχθούν είτε αυτόματα από τους πίνακες ή τα ερωτήματα της βάσης

(δεσμευμένη φόρμα) είτε βάσει τις ανάγκες του χρήστη. Οι πιο γνωστές κατηγορίες φορμών που μας παρέχει η MS Access είναι:

- α. Η **φόρμα λεπτομερειών**, που είναι μια κλασική κενή φόρμα της Access όπου εμφανίζει τα δεδομένα μιας εγγραφής. Όλες οι φόρμες της εφαρμογής είναι σχεδιαστικά φόρμες λεπτομερειών.
- β. Η **φόρμα πολλαπλών στοιχείων** ή **Συνεχής φόρμα**, που χρησιμοποιείται για την εμφάνιση περισσότερων εγγραφών κάθε φορά. Το σχέδιο της παραπέμπει σε έναν πίνακα ή φύλλο δεδομένων και βοηθάει εμφανίζει πληροφορίες από περισσότερες από μία εγγραφή κάθε φορά.
- γ. Η **διαιρεμένη φόρμα** μέσω της οποίας τα δεδομένα εμφανίζονται ταυτόχρονα και σε πρόβλη φόρμας και σε προβολή φύλλου δεδομένων. Είναι χρήσιμη όταν η εφαρμογή απαιτεί την προβολή πολλών πληροφοριών από πολλαπλούς πίνακες που συνδέονται μεταξύ τους και απαιτείται η υλοποίηση «master-detail».
- δ. Η **φόρμα περιήγησης** που περιέχει ένα στοιχείο ελέγχου περιήγησης και αφορά εφαρμογές που είναι δημοσιευμένες στο διαδίκτυο (Web-based data applications), χωρίς αυτό να σημαίνει ότι δεν μπορεί να χρησιμοποιηθεί και σαν περιβάλλον περιήγησης για μια επιτραπέζια εφαρμογή.



Εικόνα 17: Το μενού επιλογών της MS Access 2019 για τον σχεδιασμό φορμών. Διακρίνονται όλες οι διαθέσιμες επιλογές

Επίσης μια χρήσιμη λειτουργία για τον προγραμματιστή είναι οι **προβολές** (views) των φορμών. Μια φόρμα στην MS Access μπορεί να προβληθεί στον χρήστη με τρεις διαφορετικούς τρόπους:

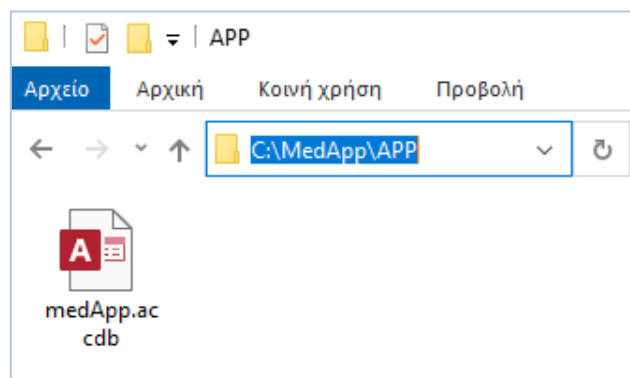
- α. Την **προβολή φόρμας**, η οποία είναι η προεπιλεγμένη προβολή και χρησιμοποιείται από τον χρήστη για την εισαγωγή και την προβολή των δεδομένων. Ουσιαστικά είναι το μεταγλωττισμένο» αποτέλεσμα της σχεδίασης της φόρμας.
- β. Την **προβολή σχεδίασης**, που είναι η βασική προβολή σχεδίασης μέσω της οποίας ο χρήστης σχεδιάζει και τροποποιεί μια φόρμα με τα εργαλεία ελέγχου (λίστες, ετικέτες, επιλογές, πλαίσια, κτλ.) που ο ίδιος επιθυμεί καθώς του προσφέρονται περισσότερες σχεδιαστικές δυνατότητες
- γ. Την **προβολή διάταξης**, στην οποία τα δεδομένα προβάλλονται ομαδοποιημένα σε γραμμές (σειρές) και στήλες, επιτρέποντας έτσι τον χρήστη να τις τροποποιήσει μαζί. Είναι κατάλληλη για φόρμες που θέλουμε να εμφανίσουμε ως πίνακα ή φύλλου δεδομένων.

Οι φόρμες στην MS Access είναι διαθέσιμες από τη καρτέλα «**Δημιουργία**», ενότητα «**Φόρμες**», και επιλέγουμε «**Σχεδίαση Φόρμας**». Επίσης μέσω των επιλογών αυτής της καρτέλας, μπορούμε να δημιουργήσουμε μια φόρμα με τη βοήθεια οδηγού (Οδηγός Φορμών) ή να φτιάξουμε άμεσα μια φόρμα τύπου πλαίσιο διαλόγου.

4.1.2 Δημιουργία του αρχείου front-end - Σύνδεση με ΒΔ

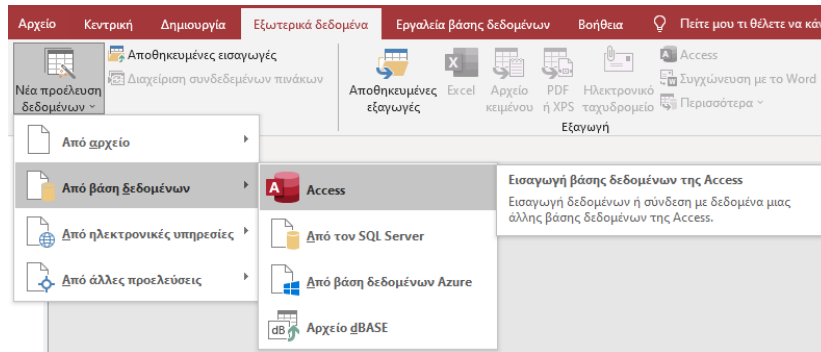
Πριν ξεκινήσουμε να σχεδιάζουμε τις φόρμες της εφαρμογής μας, πρέπει αφ' ενός να κατασκευάσουμε το αρχείο «front-end» της εφαρμογής αφ' ετέρου να συνδέσουμε σε αυτό τους πίνακες της βάσης μας, που φτιάξαμε στο προηγούμενο κεφάλαιο.

Στην τοποθεσία του καταλόγου της εφαρμογής «**C:\MedApp**», δημιουργούμε έναν νέο φάκελο με το όνομα «**APP**», (από τη λέξη *application*) και μέσα σε αυτό φτιάχνουμε ένα νέο αρχείο MS Access και το ονομάζουμε «**medApp.accdb**»



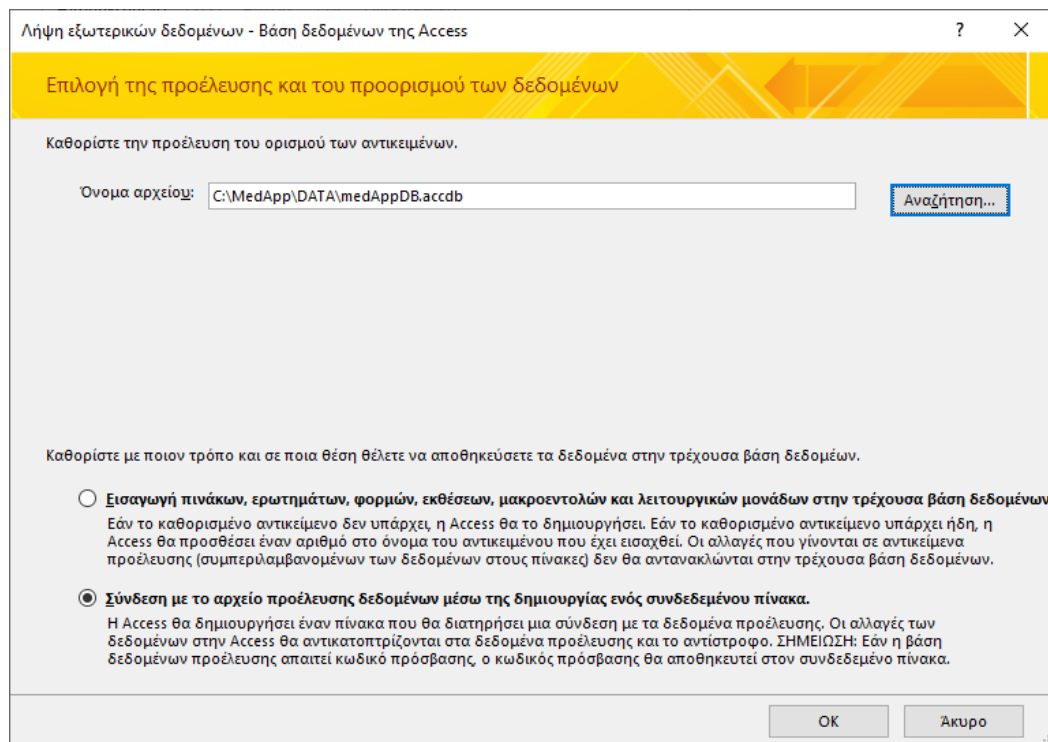
Εικόνα 18: Το αρχείο της εφαρμογής (front-end), medApp.accdb, στο φάκελο APP.

Ανοίγουμε το αρχείο που μόλις δημιουργήσαμε και οδηγούμαστε στην αρχική οθόνη της MS Access. Στο σημείο αυτό πρέπει να συνδέσουμε τους πίνακες που έχουμε φτιάξει στο αρχείο «medAppDB.accdb», στο προηγούμενο κεφάλαιο. Από την καρτέλα «Εξωτερικά Δεδομένα», επιλέγουμε «Νέα προέλευση δεδομένων» → «Από βάση δεδομένων» → «Access», όπως φαίνεται στην παρακάτω εικόνα:



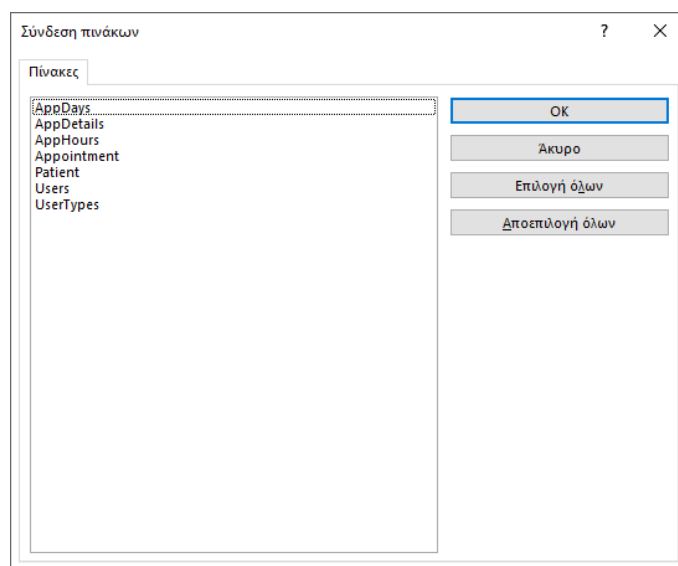
Εικόνα 19: Το μενού "Νέα Προέλευση δεδομένων" από όπου επιλέγουμε την Access ως πηγή των εξωτερικών δεδομένων.

Στη συνέχεια ανοίγει το παράθυρο διαλόγου «Λήψη Εξωτερικών Δεδομένων». Πατώντας το κουμπί «Αναζήτηση», μεταφερόμαστε στην «Εξερεύνηση αρχείων» και εντοπίζουμε το αρχείο «medAppDB.accdb» στο φάκελο «C:\MedApp\DATA». Επιλέγουμε το αρχείο και στη συνέχεια πατάμε «Άνοιγμα». Η διαδρομή του αρχείου της βάσης έχει συμπληρωθεί στο πλαίσιο «Όνομα αρχείου» όπως φαίνεται στην παρακάτω εικόνα.



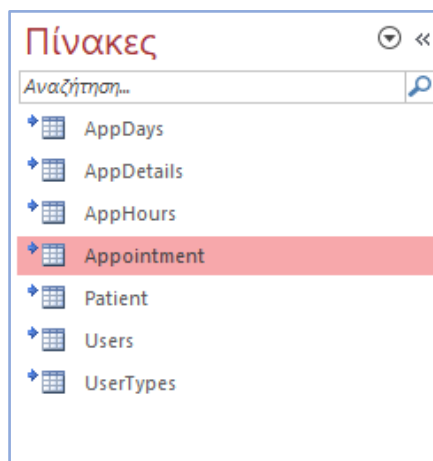
Εικόνα 20: Η εισαγωγή του αρχείου της βάσης δεδομένων.

Στη συνέχεια επιλέγουμε την επιλογή «**Σύνδεση με το αρχείο προέλευσης δεδομένων μέσω της δημιουργίας ενός συνδεδεμένου πίνακα**» και πατάμε **OK**. Ο οδηγός σύνδεσης ανοίγει το παρακάτω πλαίσιο διαλόγου, στο οποίο φαίνονται οι πίνακες της βάσης.



Εικόνα 21: Οι πίνακες της βάσης δεδομένων θα συνδεθούν μέσω του οδηγού στο νέο αρχείο.

Στη συνέχεια επιλέγουμε όλους τους πίνακες και πατάμε το κουμπί «**OK**». Οι πίνακες της βάσης έχουν συνδεθεί στο αρχείο της εφαρμογής (front-end) και από αυτό το σημείο μπορούμε να ξεκινήσουμε τον σχεδιασμό της εφαρμογής μας, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 22: Οι πίνακες του αρχείου της βάσης (back-end) έχουν συνδεθεί στο αρχείο εφαρμογής (front-end).

Σε περίπτωση που μετακινήσουμε το αρχείο της βάσης θα πρέπει να επαναλάβουμε την παραπάνω διαδικασία από την αρχή. Γι' αυτό και σε τέτοιες περιπτώσεις φροντίζουμε το αρχείο της βάσης να βρίσκεται σε κάποια προστατευμένη τοποθεσία όπου πρόσβαση έχουν μόνο εξουσιοδοτημένοι χρήστες.

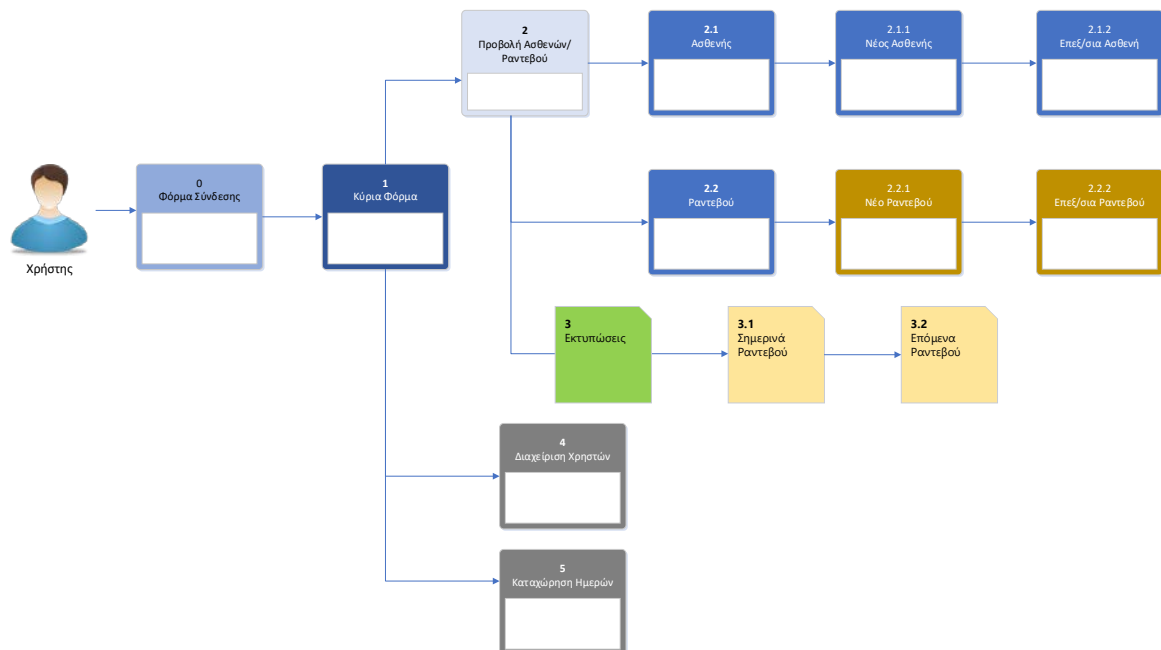
Ωστόσο, στην δική μας περίπτωση, το αρχείο της βάσης το τοποθετούμε στον ίδιο αρχικό κατάλογο (**MedApp**) σε διαφορετικό υποφάκελο (DATA), θεωρώντας την απλή περίπτωση ενός μόνου χρήστη (γιατρός) που έχει εγκατεστημένη την εφαρμογή στον προσωπικό του υπολογιστή.

Εφόσον οι πίνακες της βάσης δεδομένων έχουν συνδεθεί επιτυχώς, μπορούμε να περάσουμε στην φάση του σχεδιασμού των φορμών, των ερωτημάτων, των αναφορών και των λειτουργικών μονάδων που θα χρησιμοποιήσουμε στην εφαρμογή μας.

4.1.3 Διάγραμμα των διαλόγων/λειτουργιών της εφαρμογής

Γενικά στη σχεδίαση λογισμικού, είναι πολύ χρήσιμο η κατασκευή του διαγράμματος διαλόγων της εφαρμογής καθώς έτσι παριστάνουμε οπτικά την εφαρμογή (σύστημα) που θέλουμε να κατασκευάσουμε προσδιορίζοντας παράλληλα και την συνολική συμπεριφορά του συστήματος. Η τακτική αυτή ονομάζεται «**Ενοποιημένη Γλώσσα Σχεδιασμού**» **UML** (Unified Modelling Language) και είναι μια γραφική γλώσσα για την οπτική αναπαράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε οποιοδήποτε λογισμικό αντικειμενοστραφούς κυρίως σχεδίασης.

Ωστόσο το διάγραμμα αυτό μπορεί να σχεδιαστεί για όλου των τύπων της εφαρμογές όπως και στην εφαρμογή μας όπως φαίνεται παρακάτω.



Εικόνα 23: Το διάγραμμα διαλόγων της εφαρμογής. Περιέχει τις βασικές λειτουργίες της εφαρμογής που είναι διαθέσιμες στο χρήστη.

4.2 Τα Ερωτήματα

Τα ερωτήματα (queries) που θα χρησιμοποιήσουμε στην εφαρμογή είναι στατικά, δηλαδή έχουν δημιουργηθεί απευθείας από τους πίνακες και προβάλλουν συγκεκριμένα δεδομένα των πινάκων μέσα από τις κατάλληλες φόρμες που θα περιγράψουμε στη συνέχεια. Αν και τα ερωτήματα είναι προτιμότερο να τα δημιουργούμε στο αρχείο της βάσης (back-end), θα τα σχεδιάσουμε στο αρχείο της εφαρμογής για να βελτιώσουμε έτσι την ταχύτητα της ανάκτησης των δεδομένων και να έχουμε καλύτερο έλεγχο αυτών των ερωτημάτων.

Για να σχεδιάσουμε ένα ερώτημα, μπορούμε να το κάνουμε εύκολα από τον οδηγό των ερωτημάτων που προσφέρει η MS Access, από την καρτέλα «**Δημιουργία**» → «**Ερωτήματα**» → «**Σχεδίαση Ερωτήματος**». Ωστόσο θα προτιμήσουμε την σύνταξη τους με την βοήθεια του ενσωματωμένου επεξεργαστή εντολών **SQL (Προβολή SQL)** που διαθέτει η MS Access.

4.2.1 Το ερώτημα «qryTodayAppointments»

Χρησιμοποιούμε το ερώτημα «qryTodayAppointments» για να προβάλλουμε τα ραντεβού της τρέχουσας ημέρας. Προβάλλει την ημερομηνία, την ώρα, το ΑΜΚΑ και το ονοματεπώνυμο του ασθενή και χρησιμοποιείται ως δεδομένα προέλευσης για την φόρμα «sfmTodayAppts» και την έκθεση «rptTodayAppts».

Κώδικας SQL:

```
SELECT Appointment.AppDate,
       AppDetails.AppHour,
       Patient.AMKA,
       Patient.LName,
       Patient.FName,
       Appointment.Subject, [LName] & ' ' & [FName] AS PName
FROM Patient
      INNER JOIN (Appointment INNER JOIN
                  (AppHours INNER JOIN AppDetails ON AppHours.AppHour = AppDetails.AppHour)
                  ON Appointment.Id = AppDetails.AppId)
      ON Patient.Id = Appointment.PId
WHERE Appointment.AppDate = Date()
ORDER BY Appointment.AppDate, AppDetails.AppHour;
```

4.2.2 Το ερώτημα «qryNextDayAppointments»

Χρησιμοποιούμε το ερώτημα «qryNextDayAppointments» για να προβάλουμε όλα τα ραντεβού της επόμενης ημέρας. Ομοίως με το προηγούμενο προβάλλει την ημερομηνία, την ώρα, το ΑΜΚΑ και το ονοματεπώνυμο του ασθενή και χρησιμοποιείται ως δεδομένα προέλευσης για την φόρμα «sfmNextAppts» και την έκθεση «rptComingAppts».

Κώδικας SQL:

```
SELECT Appointment.AppDate,
       AppDetails.AppHour,
       Patient.AMKA,
       Patient.LName,
       Patient.FName,
       Appointment.Subject,
       [LName] & ' ' & [FName] AS PName
FROM Patient
     INNER JOIN Appointment INNER JOIN
         (AppHours INNER JOIN AppDetails
          ON AppHours.AppHour = AppDetails.AppHour)
     ON Appointment.Id = AppDetails.AppId
     ON Patient.Id = Appointment.PId
WHERE Appointment.AppDate = Date()+1
ORDER BY Appointment.AppDate, AppDetails.AppHour;
```

4.2.3 Το ερώτημα «qryAppointments»

Χρησιμοποιούμε το ερώτημα «qryAppointments» για να προβάλουμε τις λεπτομέρειες ενός ραντεβού που κλείνει κάποιος ασθενής. Προβάλλονται η ημερομηνία, η ώρα, το θέμα της επίσκεψης, τα σχόλια και ο κωδικός του ασθενή και χρησιμοποιείται ως δεδομένα προέλευσης για την φόρμα «frmSubAppointment».

Κώδικας SQL:

```
SELECT Appointment.Id,
       Appointment.PId,
       Appointment.AppDate,
       AppDetails.AppHour,
       Appointment.Subject,
       Appointment.Notes
FROM Appointment
     LEFT JOIN AppDetails ON Appointment.Id = AppDetails.AppId;
```

4.2.4 Το ερώτημα «qryPatientGender»

Χρησιμοποιούμε το ερώτημα «qryPatientGender» για να προβάλουμε τον αριθμό των ασθενών ανά φύλο. Θα το χρησιμοποιήσουμε για την εκτύπωση κατάλληλου γραφήματος όπου θα προβάλλονται το ποσοστό των ανδρών και των γυναικών που έχουν επισκεφτεί το ιατρείο.

Κώδικας SQL:

```
SELECT Patient.Gender,
        Count(Patient.Gender) AS ΠλήθοςΤουGender
FROM Patient
GROUP BY Patient.Gender
ORDER BY Patient.Gender, Count(Patient.Gender);
```

4.2.5 Το ερώτημα «qryAgeGroups»

Χρησιμοποιούμε το ερώτημα «qryAgeGroups» για να προβάλουμε τον αριθμό των ασθενών ανά ηλικιακή ομάδα και θα το χρησιμοποιήσουμε για την εκτύπωση κατάλληλου γραφήματος όπου θα προβάλλονται τα δεδομένα αυτά.

Κώδικας SQL:

```
SELECT AG.AgeGroups AS Ages,
        Count(AG.AgeGroups) AS NumofAgeGroups
FROM (SELECT Switch
      (
        [Age] Between 18 And 25, "18-25",
        [Age] Between 26 And 35, "26-35",
        [Age] Between 36 And 45, "36-45",
        [Age] Between 46 And 55, "46-55",
        [Age] Between 56 And 60, "56-60",
        [Age] > 60, "Άνω 60"
      ) AS AgeGroups
FROM
      (
        SELECT
          DateDiff("yyyy", [DOB], Date()) - IIf(Format(Date(), "mmd") <
          Format([DOB], "mmd"), 1, 0) AS Age
        FROM Patient
      )
      ) AS AG
GROUP BY AG.AgeGroups;
```

4.3 Οι Φόρμες

Πριν περάσουμε στην περιγραφή των φορμών, θα αναφέρουμε περιληπτικά τις δυνατότητες της εφαρμογής.

Όπως αναφέραμε και σε προηγούμενο κεφάλαιο, η εφαρμογή είναι κυρίως επιτραπέζια (desktop application), που αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί από έναν μόνο χρήστη-υπολογιστή. Βέβαια με την μέθοδο διαχωρισμού της εφαρμογής σε δεδομένα-λειτουργίες, μπορεί να λειτουργήσει και ως δικτυακή εφαρμογή, δηλαδή να χρησιμοποιηθεί από περισσότερους τους ενός χρήστες, αρκεί η τοποθεσία αποθήκευσης του αρχείου δεδομένων να είναι μια ασφαλής τοποθεσία δικτύου.

Λόγω ότι η MS Access δεν υποστηρίζει ρόλους και είδη χρηστών, προσαρμόσαμε την εφαρμογή έτσι ώστε να προσεγγίζει τους ρόλους των χρηστών και έτσι ο χρήστης ανάλογα τον τύπο του (administrator, superuser, κτλ.) θα μπορεί να χρησιμοποιεί τις κατάλληλες ενότητες. Στην περίπτωση μας, έχουμε δυο τύπους χρηστών, α) τον γιατρό-διαχειριστής και β) την γραμματεία-user.

Σύμφωνα με το διάγραμμα διαλόγων που σχεδιάσαμε σε προηγούμενη ενότητα (Εικόνα 23) οι λειτουργίες της εφαρμογής είναι:

- Σύνδεση Χρήστη.
- Προβολή ραντεβού τρέχουσας και επόμενης ημέρας.
- Προβολή ραντεβού ανά ασθενή.
- Προσθήκη νέου ασθενή.
- Τροποποίηση στοιχείων ασθενή.
- Προσθήκη νέου ραντεβού/επίσκεψης.
- Τροποποίηση ραντεβού ασθενή.
- Εκτύπωση ραντεβού τρέχουσας και επόμενης ημέρας.
- Προβολή στατιστικών στοιχείων.
- Προσθήκη και τροποποίηση χρηστών της εφαρμογής.
- Διαχείριση ημερών και ωρών λειτουργίας ιατρείου.

Από τις παραπάνω λειτουργίες, σχεδιάσαμε τις παρακάτω φόρμες:

- Σύνδεση Χρήστη

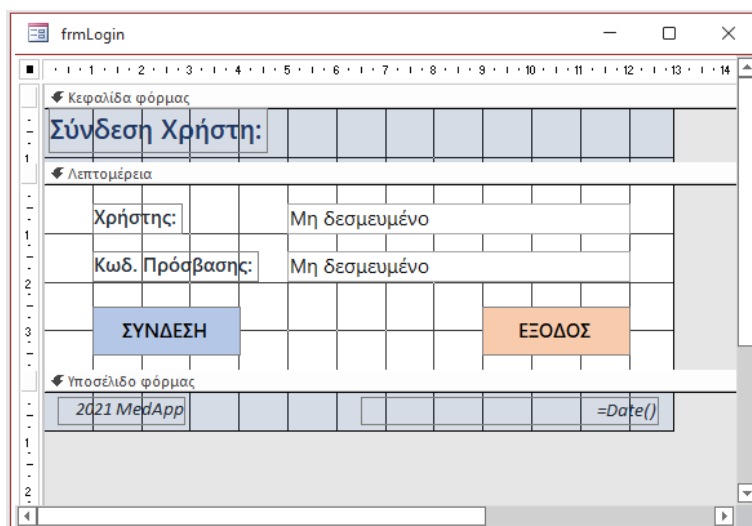
- Κεντρική Φόρμα (Αρχική)
- Καρτέλα Ραντεβού Ασθενή
- Νέος Ασθενής
- Επεξ/σια Στοιχείων Ασθενή
- Νέο ραντεβού
- Επεξεργασία Ραντεβού
- Προβολή Στοιχείων Ασθενή
- Προβολή Στοιχείων Ραντεβού
- Προβολή Σημερινών ραντεβού
- Προβολή Ραντεβού επόμενης ημέρας
- Διαχείριση Χρηστών
- Διαχείριση Ημερών και Ωρών εργασίας.

Παρατηρήσεις:

- Χρησιμοποιούμε την **ελληνική έκδοση της MS Access 2019**.
- Σε όλες τις φόρμες ακολουθήσαμε τους προτεινόμενους χρωματικούς συνδυασμούς του MS Office.
- Τα στοιχεία ελέγχου (κουμπιά, λίστες, κτλ) που χρησιμοποιούμε σε κάθε φόρμα αναφέρονται και με την **ελληνική** αλλά και με την **αγγλική** ορολογία (κατά προτίμηση).
- Τα εικονίδια που χρησιμοποιήθηκαν σε στοιχεία ελέγχου είναι ελεύθερης διανομής από τον ιστότοπο IconArchive.com
- Για την προβολή των φορμών του αρχείου MS Access, εφαρμόσαμε την προεπιλεγμένη ρύθμιση για «**Έγγραφα με καρτέλες**», ενώ για την παραγωγή των παρακάτω στιγμιότυπων των φορμών ορίσαμε την ρύθμιση ως «**Επικαλυπτόμενα παράθυρα**». Για την αλλαγή των ρυθμίσεων: «**Αρχείο**» → «**Επιλογές**» → «**Τρέχουσα βάση δεδομένων**» → «**Επιλογές εφαρμογής**» → «**Επιλογές Παραθύρου εγγράφου**».
- Ο κώδικας VBA που αναφέρεται στα συμβάντα (event) της κάθε φόρμας είναι διαθέσιμος στο κεφάλαιο «**ΠΑΡΑΡΤΗΜΑΤΑ**».

- Οποιοδήποτε μέρος κώδικα που εκτελεί μια λειτουργία που αφορά τα δεδομένα της ΒΔ, «κλείνεται» μέσα σε έλεγχο ασφαλείας. Ο έλεγχος ασφαλείας γίνεται μέσω του μπλοκ εντολών **On Error GoTo ... Exit Here ... Exit Sub** (Περισσότερα στο κεφάλαιο «ΠΑΡΑΡΤΗΜΑΤΑ»).

4.3.1 Η φόρμα «frmLogin» - Σύνδεση Χρήστη



Εικόνα 24: Η φόρμα «Σύνδεση Χρήστη» (frmLogin)

Η φόρμα «Σύνδεση Χρήστη» είναι η πρώτη φόρμα που εμφανίζεται κατά την λειτουργία της εφαρμογής. Για να ενεργοποιηθεί η εφαρμογή, ο χρήστης πληκτρολογεί το όνομα του στο πλαίσιο «Χρήστης».

Εάν το όνομα χρήστη δεν υπάρχει καταχωρημένο στη ΒΔ εμφανίζεται σχετικό μήνυμα που προτρέπει τον χρήστη να επικοινωνήσει με τον διαχειριστή του συστήματος. Σε διαφορετική περίπτωση, ο χρήστης πληκτρολογεί τον κωδικό πρόσβασης του στο πεδίο «Κωδ. Πρόσβασης» και πατάει το κουμπί «ΣΥΝΔΕΣΗ». Εάν ο κωδικός πρόσβασης είναι λάθος, ο χρήστης δοκιμάζει ξανά, διαφορετικά η σύνδεση είναι επιτυχής, η φόρμα κλείνει και ανοίγει η κεντρική φόρμα επιλογών. Με το κουμπί «ΕΞΟΔΟΣ» τερματίζεται η εφαρμογή. Για την υλοποίηση της φόρμας χρησιμοποιήσαμε τα παρακάτω στοιχεία ελέγχου:

Πίνακας 25: Τα στοιχεία ελέγχου της φόρμας «Σύνδεση Χρήστη»

Στοιχεία Ελέγχου (Components)	Όνομασία Στοιχείου	Χρήση
Κουμπί (Button)	btnLogin	ΣΥΝΔΕΣΗ
	btnExit	ΕΞΟΔΟΣ
Πλαίσιο Κειμένου (Textbox)	txtUsername	Όνομα Χρήστη
	txtPassword	Κωδικός Πρόσβασης
Ετικέτες (Label)	lblUser	Χρήστης:

	lblPwd	Κωδ. Πρόσβασης
--	--------	----------------

Για το κουμπί «ΕΙΣΟΔΟΣ» ο κώδικας VBA που συντάξαμε είναι ο παρακάτω:

```
Private Sub btnLogin_Click()
On Error GoTo Err_Handler
' Ελέγχουμε εάν έχει συμπληρωθεί το πεδίο Χρήστης (username)
If IsNull(Me.txtUsername) Or Me.txtUsername = "" Then
MsgBox "Συμπληρώστε το πεδίο ΧΡΗΣΤΗΣ!", vbOKOnly, "Στοιχεία
χρήστη - Χρήστης"
Me.txtUsername.SetFocus
Exit Sub
' Ελέγχουμε εάν έχει συμπληρωθεί το πεδίο Κωδ. Πρόσβασης (pass-
word)
ElseIf IsNull(Me.txtPassword) Or Me.txtPassword = "" Then
MsgBox "Συμπληρώστε το πεδίο ΚΩΔ. ΠΡΟΣΒΑΣΗΣ!", vbOKOnly,
"Στοιχεία χρήστη - Κωδ. Πρόσβασης"
Me.txtPassword.SetFocus
Exit Sub
Else
' Ελέγχουμε εάν ο Κωδ. Πρόσβασης είναι σωστός. Εάν είναι σω-
στός, ανοίγει η κύρια φόρμα
' και η φόρμα login κλείνει.
If DLookup("Password", "Users", "[Id] = " & userID) =
Me.txtPassword.Value Then
DoCmd.Close acForm, "frmLogin", acSaveNo
DoCmd.OpenForm "frmMain"
Else
MsgBox "Λάθος Κωδικός Πρόσβασης. Προσπαθείτε ξανά!", _
vbOKOnly + vbExclamation, "Σφάλμα Σύνδεσης Χρήστη"
Me.txtPassword.SetFocus
End If
End If
Exit_Here:
CurrentDb.Close
Exit Sub
Err_Handler:
DoCmd.Hourglass False
MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub
```

Για τον έλεγχο του ονόματος του χρήστη όταν δοθεί στο πλαίσιο ελέγχου «Χρήστης», ο κώ-
δικας που συντάξαμε είναι ο παρακάτω:

```
Private Sub txtUsername_AfterUpdate()
On Error GoTo Err_Handler
' Ελέγχουμε κατά την διάρκεια της πληκ/σης εάν το
' Username υπάρχει καταχωρημένο στη βάση
' μόλις αλλάξει πεδίο
Dim criteria As String
```

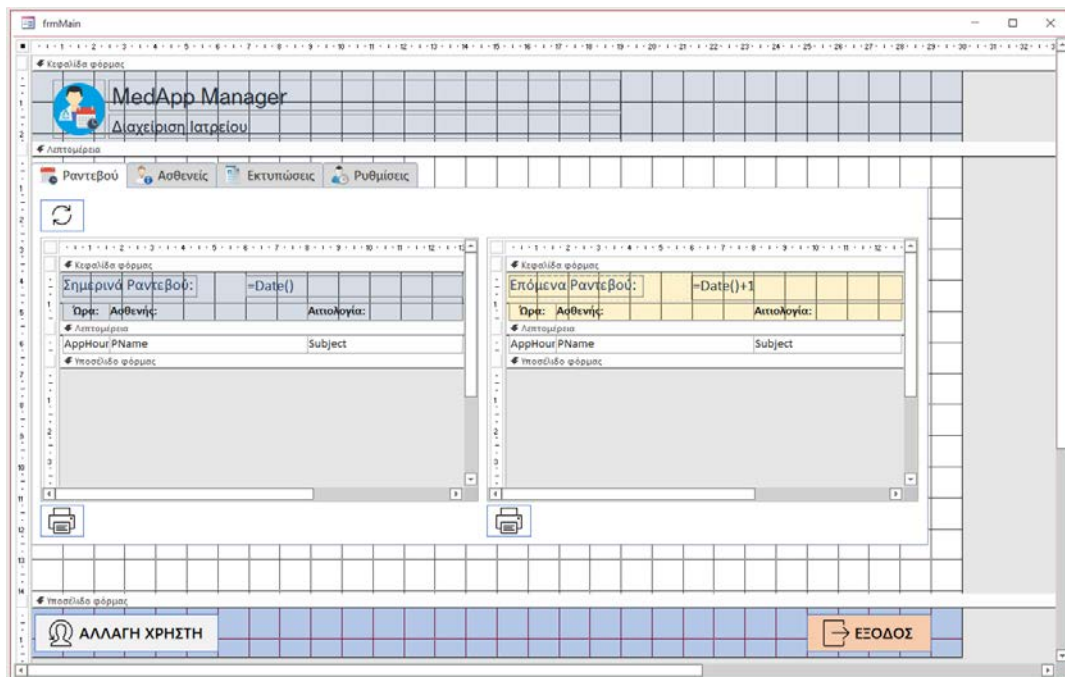


```

criteria = "[Username] = '" & Me.txtUsername.Value & "'"
' Σε περίπτωση που δεν βρεθεί ο χρήστης της εφαρμογής
' εμφανίζεται σχετικό μήνυμα λάθους, διαφορετικά
' εκχωρούμε στην public μεταβλητή userID την τιμή του πίνακα.
If DCount("Id", "Users", criteria) = 0 Then
    MsgBox "Ο χρήστης '" & Me.txtUsername.Value & "' δεν υπάρχει!"
    + " Επικοινωνήστε με τον διαχειριστή του συστήμα-
    τος!", vbExclamation + vbOKCancel, _
    "Έλεγχος Χρήστη"
    Me.txtUsername.SetFocus
    Exit Sub
Else
    userID = DLookup("Id", "Users", criteria)
    Me.txtPassword.SetFocus
End If
Exit_Here:
CurrentDb.Close
Exit Sub
Err_Handler:
DoCmd.Hourglass False
MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub

```

4.4.2 Η φόρμα «frmMain» - Κεντρική Φόρμα Επιλογών



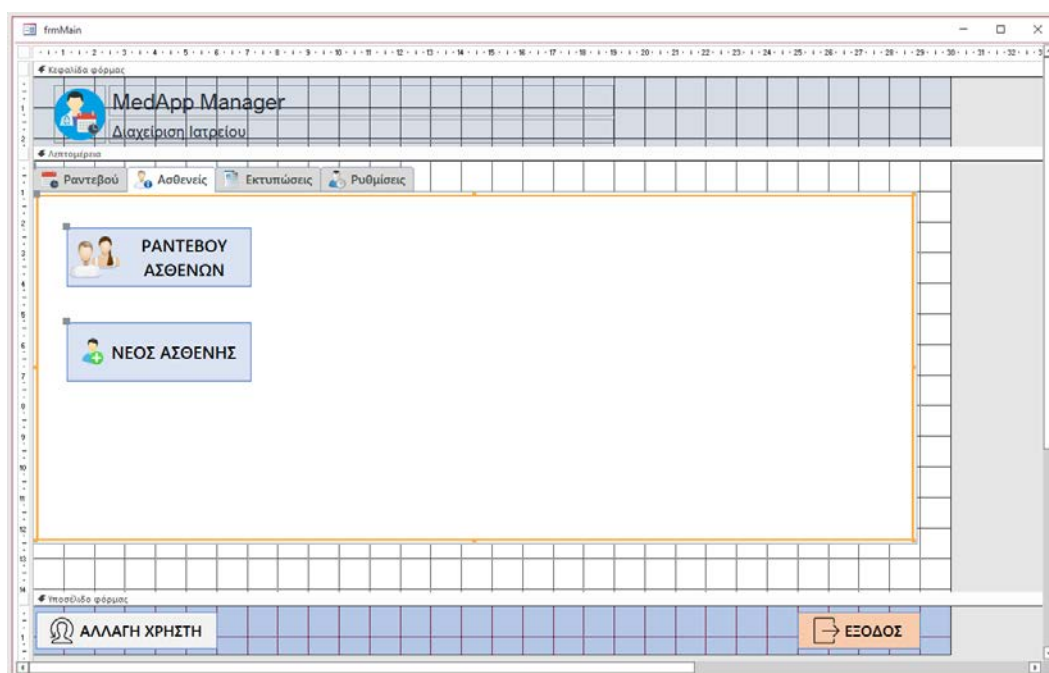
Εικόνα 25: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ραντεβού»

Η κεντρική φόρμα επιλογών (**Αρχική**) αποτελεί τον κορμό της εφαρμογής καθώς μέσω αυτής προσφέρεται η πρόσβαση σε όλες τις βασικές λειτουργίες της εφαρμογής, ανάλογα βέ-

βια με τον τύπο του χρήστη. Η υλοποίηση μιας κεντρικής φόρμας επιλογών είναι συνηθισμένη στο περιβάλλον της MS Access γιατί απουσιάζει η δυνατότητα υλοποίησης ξεχωριστών κλασικών μενού επιλογών-λειτουργιών. οι λειτουργίες της εφαρμογής έχουν οργανωθεί σε καρτέλες (tabs) για την καλύτερη διαχείριση από τον χρήστη. Οι καρτέλες αυτές είναι:

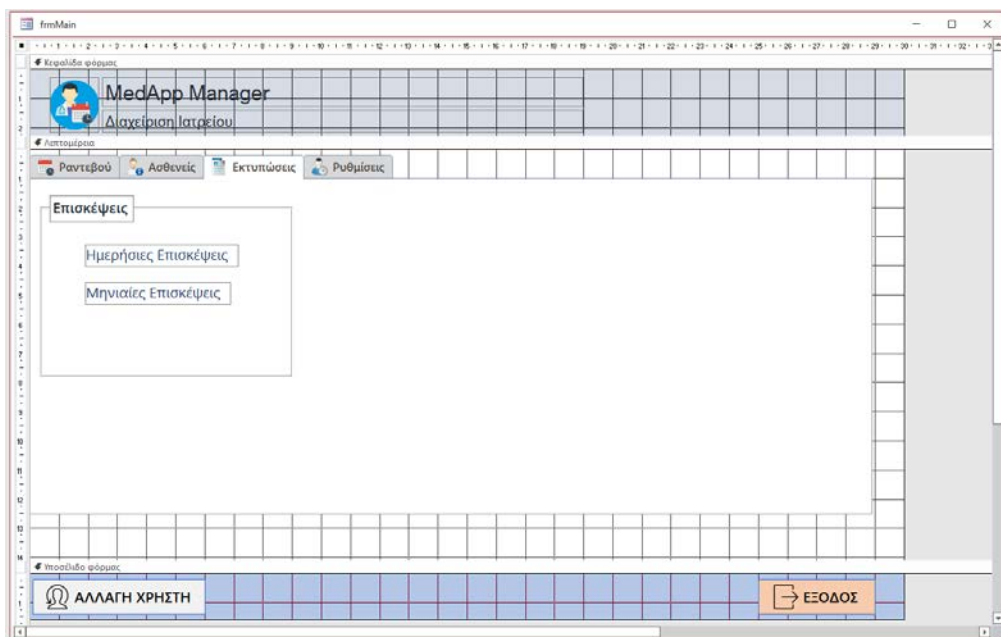
- Ραντεβού
- Ασθενείς
- Εκτυπώσεις
- Ρυθμίσεις

Η καρτέλα «**Ραντεβού**» περιέχει δυο υποφόρμες-πίνακες στους οποίους προβάλλονται τα ραντεβού της τρέχουσας και της επόμενης ημέρας ενώ μέσω κατάλληλων κουμπιών ο χρήστης μπορεί να εκτυπώσει τις σχετικές αναφορές που προκύπτουν από αυτές. Η καρτέλα «**Ασθενής**» περιέχει δυο κουμπιά (**Εικόνα 26**) μέσω των οποίων ο χρήστης μπορεί να ανοίξει την φόρμα «Ραντεβού Ασθενών» και την φόρμα «Νέος Ασθενής».



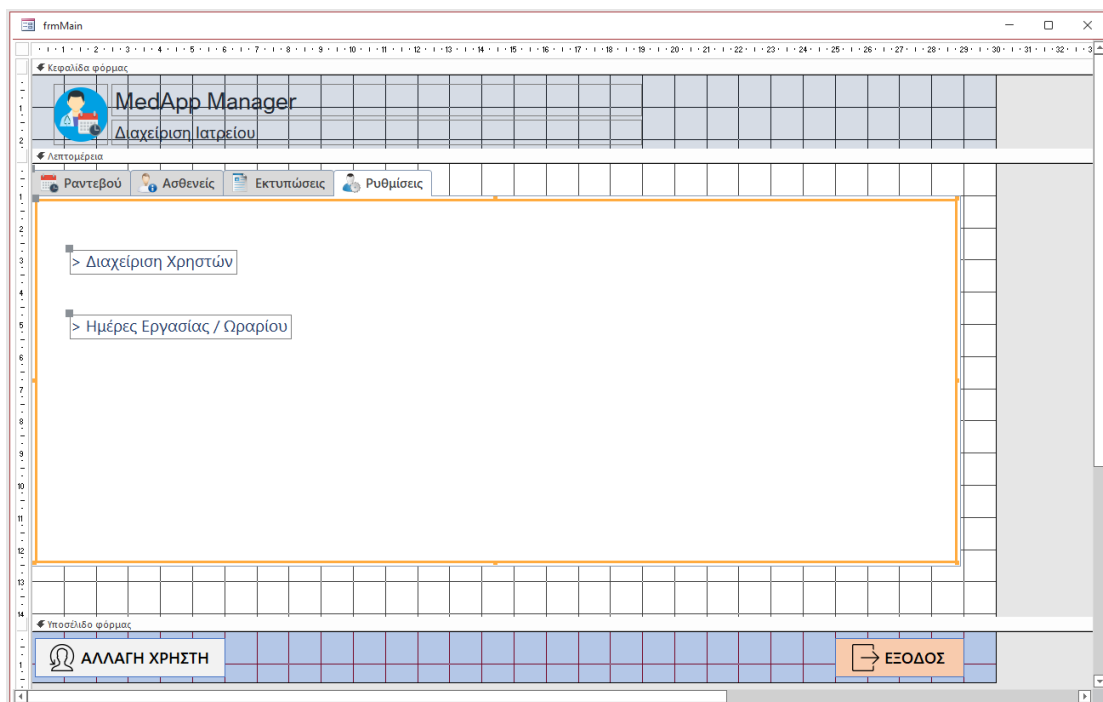
Εικόνα 26: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ασθενείς»

Η καρτέλα «**Εκτυπώσεις**» παρέχει δυο επιλογές (**Εικόνα 27**) για την εκτύπωση δυο βασικών εκτυπώσεων (αναφορών), των ημερήσιων επισκέψεων και των μηνιαίων επισκέψεων.



Εικόνα 27: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Εκτυπώσεις»

Η καρτέλα «**Ρυθμίσεις**» (Εικόνα 28) παρέχει στο χρήστη πρόσβαση σε δυο επιπλέον λειτουργίες της εφαρμογής, α) την διαχείριση των χρηστών της εφαρμογής (δημιουργία, επεξεργασία ,κτλ.) και την δημιουργία του ημερολογίου (ημέρες και ώρες εργασίας) του ιατρού. Και οι δυο αυτές λειτουργίες διαθέτουν τις δικές τους φόρμες λειτουργίας που θα περιγράψουμε σε επόμενες παραγράφους.



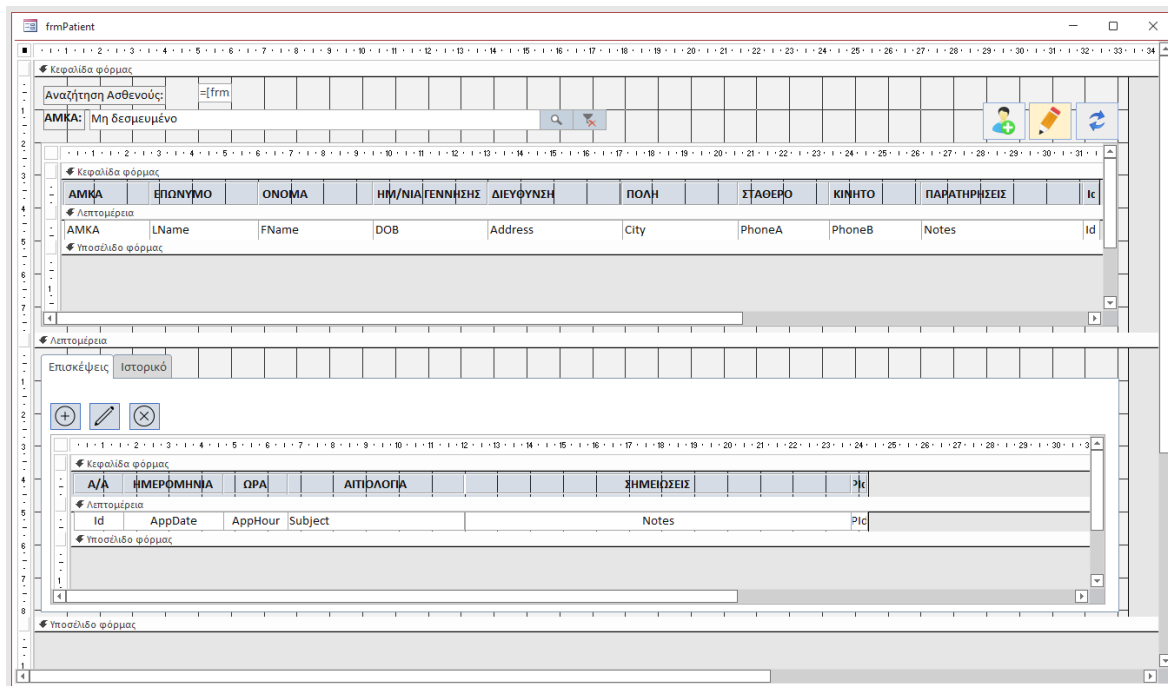
Εικόνα 28: Η Κύρια (Εισαγωγική) φόρμα της εφαρμογής (frmMain) με επιλεγμένη την καρτέλα «Ρυθμίσεις»

Τέλος η κεντρική φόρμα διαθέτει δυο κουμπιά, το κουμπί «ΑΛΛΑΓΗ ΧΡΗΣΤΗ» μέσω του οποίου ανοίγει η φόρμα «Σύνδεση Χρήστη» για να γίνει η αλλαγή πρόσβασης ενός χρήστη και το κουμπί «ΕΞΟΔΟΣ» με το οποίο τερματίζεται η εφαρμογή. Συνοπτικά τα στοιχεία ελέγχου της κεντρικής φόρμας είναι:

Πίνακας 26: Τα στοιχεία ελέγχου της φόρμας «Αρχική»

Στοιχεία Ελέγχου (Components)	Ονομασία Στοιχείου	Χρήση
Κουμπί (Button)	btnOpenLogin	ΑΛΛΑΓΗ ΧΡΗΣΤΗ
	btnTerminate	ΕΞΟΔΟΣ
	btnRefreshAll	Ανανέωση δεδομένων
	btnPrintToday	Άνοιγμα αναφοράς
	btnPrintNextday	Άνοιγμα αναφοράς
	btnOpenPatient	PANTEBOY ΑΣΘΕΝΩΝ
	btnOpenNewPatient	ΝΕΟΣ ΑΣΘΕΝΗΣ
Καρτέλα Ελέγχου (Tab Control)	tabMain	Καρτέλα Επιλογών
Ετικέτες (Label)	lblDailyArpts	Ημερήσιες επισκέψεις
	lblMonthArpts	Μηνιαίες επισκέψεις
	lblAddUsers	Διαχείριση Χρηστών
	lblAddCalendar	Ημέρες/Ωρες Εργασίας

4.4.3 Η φόρμα «frmPatientArpts» - Επισκέψεις Ασθενών



Εικόνα 29: Η φόρμα «Ασθενείς» (frmPatient). Διακρίνονται στο πάνω μέρος τα στοιχεία του ασθενή και στο κάτω μέρος τα ραντεβού του ασθενή.







Η φόρμα «**Ραντεβού Ασθενών**» είναι η βασική φόρμα της εφαρμογής. Μέσα από αυτή την φόρμα, ο χρήστης μπορεί να διαχειρίζεται τα στοιχεία των ασθενών και τις επισκέψεις τους στο ιατρείο. Ο σχεδιασμός της φόρμας βασίστηκε στο μοντέλο σχεδιασμού **master-detail** (κύρια-λεπτομέρεια).

Το μοντέλο master-detail περιγράφει την σχέση «**ένα-προς-πολλά**» μεταξύ δυο οντοτήτων, που σε αυτή την περίπτωση ο ασθενής είναι η κύρια σχέση (master) και τα ραντεβού είναι η λεπτομέρεια (detail). Για αυτό το λόγο σχεδιάστηκε μια φόρμα-πίνακας που προβάλλει τα στοιχεία των ασθενών και μια φόρμα-πίνακας που προβάλλει τα ραντεβού του κάθε ασθενή. Οι δυο αυτές φόρμες συνδέονται μεταξύ τους μέσω του μοναδικού αριθμού του ασθενή (Patient.Id <-> Appointment.PId).

Όταν ανοίγει η φόρμα «**Ραντεβού ασθενών**» οι φόρμες προβάλλουν όλα τα δεδομένα ασθενών και ραντεβού. Για να δημιουργήσει ο χρήστης ένα νέο ραντεβού, είτε επιλέγει τον ασθενή από την λίστα είτε τον αναζητά με το **ΑΜΚΑ** του μέσω του πεδίου «**Αναζήτηση**». Εάν ο ασθενής βρεθεί, τα στοιχεία προβάλλονται στους πίνακες-φόρμες. Διαφορετικά, ο χρήστης ενημερώνεται ότι δεν υπάρχει καταχωρημένος ασθενής στην βάση και προχωράει μέσω του κατάλληλου κουμπιού στην καταχώρηση του νέου ασθενή.

Τα κουμπιά που αφορούν την διαχείριση των στοιχείων του ασθενή και των ραντεβού φαίνονται στον παρακάτω πίνακα:

Πίνακας 27: Τα κουμπιά των βασικών λειτουργιών επί των ασθενών και των ραντεβού

	Προσθήκη Νέου Ασθενή
	Επεξεργασία Ασθενή
	Ανανέωση Δεδομένων
	Νέο ραντεβού
	Επεξεργασία Ραντεβού
	Ακύρωση Επίσκεψης

Τα στοιχεία ελέγχου που χρησιμοποιούνται στην φόρμα δίνονται στον επόμενο πίνακα:

Πίνακας 28: Τα στοιχεία ελέγχου της φόρμας «Ραντεβού Ασθενή»

Στοιχεία Ελέγχου (Components)	Ονομασία Στοιχείου	Χρήση
Κουμπί (Button)	btnOpenAddPatient	Άνοιγμα «Νέος Ασθενής»
	btnOpenEditPatient	Άνοιγμα «Επεξ/σια Ασθενή»
	btnRefresh	Ανανέωση φορμών
	btnAddApp	Άνοιγμα «Νέο Ραντεβού»
	btnEditApp	Άνοιγμα «Επεξ/σια Ραντεβού»
	btnCancelApp	Ακύρωση επίσκεψης
	btnSearch	Αναζήτηση Ασθενή
	btnShowAll	Εμφάνιση όλων
Δευτερεύουσα φόρμα (Sub Form)	frmsubPatient	Φόρμα-πίνακας Ασθενών
	frmsubAppointment	Φόρμα-πίνακας Επισκέψεων ασθενή
Πλαίσιο Κειμένου (Textbox)	txtSearchAMKA	Αναζήτηση AMKA
	txtp_id	Πεδίο προσωρ. αποθήκευσης κωδικού ασθενή
Ετικέτες (Label)	lblPatients	Ασθενείς
	lblAppts	Επισκέψεις

Η αναζήτηση ενός ασθενή με βάση το **AMKA** του γίνεται, πληκτρολογώντας το AMKA στο πεδίο της αναζήτησης, είτε πατώντας ENTER είτε το εικονίδιο της αναζήτησης. Και οι δυο λειτουργίες καλούν την ίδια ενέργεια. Ωστόσο επειδή είναι πιο βολική για τον χρήστη η πληκτρολόγηση του AMKA και έπειτα η αναζήτηση του, στο συμβάν «Μετά από ενημέρωση» (**AfterUpdate** event) του πεδίου **txtSearchAMKA** γράψαμε τον παρακάτω κώδικα:

```
' Αναζήτηση Ασθενή με βάση το AMKA του
Private Sub txtSearchAMKA_AfterUpdate()
On Error GoTo Err_Handler
    Dim criteria As String
    ' Εάν δεν δοθούν τιμές στο πεδίο αναζήτησης, εμφανίζονται
    ' όλες οι εγγραφές του πίνακα Patient
    If (Trim(Me.txtSearchAMKA.Value & "") = vbNullString) Then
        Forms!frmPatient!frmsubPatient.Form.RecordSource = "SELECT *
FROM Patient ORDER BY Id"
    Else
        ' Διαφορετικά, εμφανίζεται ο ασθενής βάσει το AMKA του.
        criteria = " LIKE '*' & Me.txtSearchAMKA.Value & '*'"
        Forms!frmPatient!frmsubPatient.Form.RecordSource = "SELECT *
FROM Patient WHERE AMKA " & criteria
    End If
    ' Σε περίπτωση που δεν βρεθεί ασθενής, εμφανίζεται κατάλληλο μή-
    νυμα
```

```

    If Forms!frmPatient!frmsubPatient.Form.RecordsetClone.RecordCount
= 0 Then
        Cancel = True
        RESPONSE = MsgBox("Δεν βρέθηκαν εγγραφές με αυτό το ΑΜΚΑ.",
vbCritical + vbOKOnly, "Εύρεση ΑΜΚΑ")
        Forms!frmPatient!frmsubPatient.Form.RecordSource = "SELECT *
FROM Patient ORDER BY Id"
        Me.txtSearchAMKA.SetFocus
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

Τον ίδιο κώδικα τον περνάμε και στο κουμπί της Αναζήτησης. Εύκολα γράφουμε μόνο την κλήση της ρουτίνας txtSearchAMKA_AfterUpdate στο κατάλληλο συμβάν (Με το κλικ) του κουμπιού «Αναζήτηση»:

```

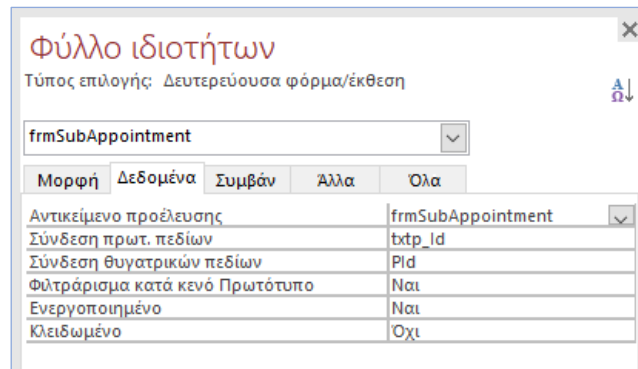
Private Sub btnSearch_Click()
    txtSearchAMKA_AfterUpdate
End Sub

```

Αξίζει να σημειωθεί πως για να εμφανίζονται εναλλακτικά οι επισκέψεις του κάθε ασθενή, όταν ο χρήστης επιλέγει διαφορετικούς ασθενείς από την υπο-φορμα των ασθενών, θα πρέπει να συνδεθούν οι δυο φόρμες-πίνακες, ώστε να καλύπτουν την λειτουργία master-detail και να οριστεί ένα «κρυφό» textbox (txtp_id) το οποίο θα κρατάει κάθε φορά το Id του ασθενή. Έτσι εξασφαλίζουμε ότι κάθε φορά που ο χρήστης θα επιλέγει έναν ασθενή από την φόρμα-πίνακα ασθενών, θα εμφανίζονται οι αντίστοιχες επισκέψεις του στην φόρμα-πίνακα των ραντεβού.

Για να καλύψουμε όλα τα παραπάνω κάναμε τις εξής ενέργειες:

- Επιλέγουμε το στοιχείο ελέγχου «frmSubAppointment».
- Από την καρτέλα «Σχεδίαση Φόρμας», επιλέγουμε το «Φύλλο Ιδιοτήτων». Στο δεξί μέρος της οθόνης μας εμφανίζονται οι ιδιότητες του αντικειμένου οργανωμένες κατάλληλα σε καρτέλες.
- Από την καρτέλα «Δεδομένα» ορίζουμε κατάλληλα τις τιμές των παρακάτω ιδιοτήτων ως εξής:
 - Σύνδεση πρωτ. πεδίων: **txtp_id**
 - Σύνδεση θυγατρικών πεδίων: **PId**



Πίνακας 29: Οι απαραίτητες τιμές των ιδιοτήτων της θυγατρικής φόρμας frmSubAppointment

Έπειτα επιλέγουμε το αντικείμενο **txtp_id** και στην τιμή της ιδιότητας «**Προέλευση δεδομένων**» γράφουμε την εντολή:

```
=[frmsubPatient].[Form]![Id]
```

Για να εξασφαλίσουμε ότι κάθε φορά η τιμή του Id του ραντεβού θα αλλάζει στην θυγατρική φόρμα (φόρμα-πίνακας) «**Επισκέψεις**» ανάλογα με το Id ασθενή που επιλέχθηκε στην κύρια (Master) φόρμα, στο συμβάν «**Με το τρέχον**» (Current event) της κύριας και της θυγατρικής φόρμας προσθέτουμε τις παρακάτω εντολές:

```
' Κύρια (Master) φόρμα Ασθενή
Private Sub Form_Current()
    p_id = Null
    p_id = Me.Id.Value
End Sub

' Θυγατρική (Detail) Επισκέψεις
Private Sub Form_Current()
    app_id = Null
    app_id = Me.Id.Value
End Sub
```

Για να εξασφαλίζουμε κάθε φορά ότι ο χρήστης πρέπει να επιλέγει έναν ασθενή από την κύρια φόρμα για να αλλάξει τις ιδιότητες ενός ραντεβού, όταν θα πατάει το κουμπί «**Επεξ/σία Επίσκεψης**» θα εμφανίζεται κατάλληλο μήνυμα. Αυτό το ελέγχουμε με τον παρακάτω κώδικα στο συμβάν «Με το κλικ» του κουμπιού «**Επεξ/σία Επίσκεψης**»:

```
Private Sub btnEditApp_Click()
On Error GoTo Err_Handler
    Dim criteria As String
    p_id = Me.txtp_Id
    criteria = "PId = " & p_id
    ' Έλεγχος εάν έχει επιλεγθεί ένα ραντεβού από τη λίστα.
    If IsNull(DLookup("PId", "Appointment", criteria)) Then
        MsgBox "Δεν υπάρχει καταχωρημένο ραντεβού για αυτόν τον ασθενή, για επεξεργασία.", _
            vbCritical + vbOKOnly, "Σφάλμα | Ραντεβού ασθενή"
```



```

Exit Sub
Else
' Άνοιγμα της φόρμας Επεξεργασία Ραντεβού.
DoCmd.OpenForm "frmEditAppointment", acNormal, , , , acWin-
dowNormal
End If
Exit_Here:
Exit Sub
Err_Handler:
DoCmd.Hourglass False
MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub

```

Τέλος, ο χρήστης μπορεί να διαγράψει ένα ραντεβού από έναν χρήστη μέσω του κουμπιού «Ακύρωση επίσκεψης», αρκεί να έχει επιλέξει από την θυγατρική φόρμα το ραντεβού που θέλει. Οι εντολές κώδικα αυτής της λειτουργίας είναι:

```

' Συμβάν που ενεργοποιείται με το κουμπί Ακύρωση ραντεβού
Private Sub btnCancelApp_Click()
On Error GoTo Err_Handler
Dim sql, criteria As String
p_id = Me.txtp_Id
criteria = "PId = " & p_id
' Σε περίπτωση που πατηθεί το κουμπί 'Ακύρωση' χωρίς να υπάρχει
' κάποιο ραντεβού στον ασθενή.
If IsNull(DLookup("PId", "Appointment", criteria)) Then
MsgBox "Δεν υπάρχει καταχωρημένο ραντεβού για αυτόν τον ασθε-
νή για ακύρωση!", _
vbCritical + vbOKOnly, "Σφάλμα | Ραντεβού ασθενή"
Exit Sub
Else
' Διαφορετικά, εμφανίζεται κατάλληλο μήνυμα επιβεβαίωσης και
' ενεργοποιείται η διαγραφή του από τον πίνακα Appointments.
Response = MsgBox("Θέλετε να ακυρώσετε/διαγράψετε το ραντεβού
με A/A " & app_id & " ;", _
vbYesNo + vbQuestion, "medApp | Ακύρωση Ραντε-
βού")
If Response = vbYes Then
Set db = CurrentDb
sql = "DELETE * FROM Appointment WHERE Id = " & app_id
CurrentDb.Execute sql, dbFailOnError
DoCmd.SetWarnings True
db.Close
MsgBox "Το ραντεβού ακυρώθηκε!", vbInformation + vbOKOn-
ly, "medApp | Ακύρωση Ραντεβού"
[Forms]![frmPatient]![frmSubAppointment].Requery
GoTo Exit_Here
End If
End If
Exit_Here:

```

```

Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

4.4.4 Η φόρμα «frmAddPatient» - Νέος Ασθενής

Η φόρμα «**Νέος Ασθενής**» ενεργοποιείται είτε μέσα από την φόρμα «Επισκέψεις Ασθενών» είτε μέσω της αρχικής φόρμας στην καρτέλα «Ασθενείς» και δίνει την δυνατότητα στον χρήστη να καταχωρεί στη βάση δεδομένων έναν νέο ασθενή συμπληρώνοντας τα απαραίτητα πεδία. Εμφανίζεται στην οθόνη ως **παράθυρο διαλόγου**.

Το πεδίο **ΑΜΚΑ** είναι υποχρεωτικό πεδίο και κατά την διάρκεια της πληκτρολόγησης του ΑΜΚΑ γίνεται έλεγχος εάν πρόκειται για 11-ψηφία ακολουθία αριθμών. Εφόσον συμπληρωθεί το πεδίο της ημερομηνίας γέννησης συμπληρώνεται αυτόματα.

Εικόνα 30: Η φόρμα «Νέος Ασθενής» (frmAddPatient)

Εφόσον ο χρήστης συμπληρώσει κατάλληλα όλα τα πεδία πατάει το κουμπί «ΑΠΟΘΗΚΕΥΣΗ» και ο ασθενής αποθηκεύεται στην βάση δεδομένων, ενώ πατώντας το κουμπί «ΑΚΥΡΩΣΗ» η διαδικασία ακυρώνεται και η φόρμα κλείνει. Τα στοιχεία ελέγχου που χρησιμοποιήσαμε στη φόρμα είναι:

Πίνακας 30: Τα στοιχεία ελέγχου της φόρμας «Νέος Ασθενής»

Στοιχεία Ελέγχου (Components)	Ονομασία Στοιχείου	Χρήση
Κουμπί (Button)	btnAddPatient	Αποθήκευση ασθενή
	btnCancel	Ακύρωση αποθήκευσης
Πλαίσιο Κειμένου (Textbox)	txtAMKA	ΑΜΚΑ ασθενή
	txtLName	Επώνυμο
	txtFName	Όνομα
	txtDoB	Ημ/νια γέννησης
	txtAddress	Διεύθυνση
	txtCity	Πόλη
	txtPhoneA	Σταθερό τηλέφωνο
	txtPhoneB	Κινητό τηλέφωνο
txtNotes	Παρατηρήσεις	
Σύνθετο Πλαίσιο (Combobox)	cboGender	Φύλο

Για την αποθήκευση ενός νέου ασθενή στη βάση συντάξαμε μια ενσωματωμένη διαδικασία η οποία καλείται με το πάτημα του κουμπιού «**ΑΠΟΘΗΚΕΥΣΗ**». Η διαδικασία παίρνει ως είσοδο τις τιμές των πεδίων και καταχωρεί στον πίνακα «Patient» της ΒΔ μια νέα εγγραφή και ο κώδικας δίνεται παρακάτω:

```
Private Sub addPatient(ByVal pAMKA As Variant _
    , ByVal plname As Variant _
    , ByVal pfname As Variant _
    , ByVal pdob As Variant _
    , ByVal pGender As Variant _
    , ByVal pAddress As Variant _
    , ByVal pCity As Variant _
    , ByVal pPhoneA As Variant _
    , ByVal pPhoneB As Variant _
    , ByVal pnotes As Variant _
)
On Error GoTo Err_Handler
Dim db As DAO.Database
Set db = CurrentDb

sql = "INSERT INTO Patient"
sql = sql + " (AMKA, LName, FName, DOB, Gender, Address, City,
PhoneA, PhoneB, Notes)"
sql = sql + " VALUES('" & pAMKA & "' , '" & plname & "' , '" &
pfname & "' , '" & pdob & "' , '"
sql = sql + " '" & pGender & "' , '" & pAddress & "' , '" & pCity
& "' , '"
sql = sql + " '" & pPhoneA & "' , '" & pPhoneB & "' , '" & pnotes
& "' )"
CurrentDb.Execute sql, dbFailOnError
CurrentDb.Close
```

```

Result = MsgBox("Τα στοιχεία του ασθενούς καταχωρήθηκαν με επιτυχία!", _
vbInformation + vbOKOnly, "Στοιχεία Ασθενούς")
If Result = vbOK Then
DoCmd.Close acForm, "frmAddPatient"
End If
Exit_Here:
Exit Sub
Err_Handler:
'DoCmd.Hourglass False
MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub

```

4.4.5 Η φόρμα «frmEditPatient» - Επεξεργασία Ασθενή

Η φόρμα «Επεξ/σια Ασθενή» ενεργοποιείται μέσα από την φόρμα «Επισκέψεις Ασθενών» με το πάτημα του κουμπιού «Επεξ/σια Ασθενή» και δίνει την δυνατότητα στον χρήστη να αλλάξει τα στοιχεία ενός ασθενή στη βάση δεδομένων τροποποιώντας τις τιμές στα πεδία που επιθυμεί.

Σχεδιαστικά η φόρμα είναι παρόμοια με την προηγούμενη φόρμα της καταχώρησης ενός ασθενή. Όταν ενεργοποιηθεί, εμφανίζεται ως πλαίσιο διαλόγου. Η μόνη διαφορά είναι η προσθήκη ενός επιπλέον πεδίου (πλαίσιο κειμένου), στο οποίο περιέχεται η τιμή του πρωτεύοντος κλειδιού του ασθενή (**Id**). Το πλαίσιο κειμένου ονομάζεται «**txtId**» και κατά την εκτέλεση της εφαρμογής παραμένει κρυφό (hidden).

Εικόνα 31: Η φόρμα «Επεξ/σία στοιχείων ασθενή» (frmEditPatient)

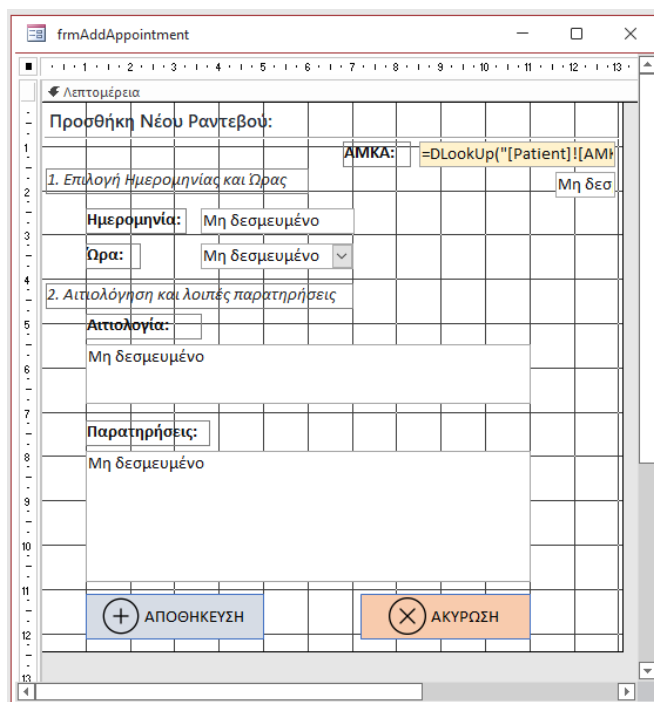
Βασική προϋπόθεση είναι ο χρήστης να έχει επιλέξει από την κύρια φόρμα (master) την εγγραφή του ασθενή που θέλει να αλλάξει. Έπειτα, πατώντας το παραπάνω κουμπί, η φόρμα ανοίγει και τα πεδία περιέχουν τις τιμές της εγγραφής.

Τα στοιχεία ελέγχου που χρησιμοποιούνται στην φόρμα είναι **παρόμοια** με αυτά της προηγούμενης φόρμας. Εφόσον ο χρήστης ολοκληρώσει τις αλλαγές πατάει το κουμπί «**ΕΝΗΜΕΡΩΣΗ**» και ενημερώνεται ο πίνακας «Patient», ενώ με το πάτημα του κουμπιού «**ΚΛΕΙΣΙΜΟ**» η φόρμα κλείνει. Ο κώδικας που χρησιμοποιείται για την ενημέρωση του ασθενή δίνεται παρακάτω:

```
Private Sub btnUpdatePatient_Click()
On Error GoTo Err_Handler
    Response = MsgBox("Θέλετε να αποθηκεύσετε τις αλλαγές ;", _
        vbYesNo + vbInformation, "medApp | Επεξεργασία Ασθενή")
    If Response = vbYes Then
        With Me.Form.Recordset
            .Edit
            .Fields("LName") = Me.txtLName
            .Fields("FName") = Me.txtFName
            .Fields("DOB") = Me.txtDoB
            .Fields("Gender") = Me.cboGender
            .Fields("Address") = Me.txtAddress
            .Fields("City") = Me.txtCity
            .Fields("PhoneA") = Me.txtPhoneA
            .Fields("PhoneB") = Me.txtPhoneB
            .Fields("Notes") = Me.txtNotes
            .Fields("UpdDate") = Now()
            .Update
        End With
        Response = MsgBox("Τα στοιχεία ενημερώθηκαν με επιτυχία!",
vbOKOnly + vbInformation, "medApp | Επεξεργασία Ασθενή")
        [Forms]![frmPatient]![frmSubPatient].Requery
        If Response = vbOK Then
            DoCmd.Close acForm, "frmEditPatient"
        End If
        rst.Close
        Set rst = Nothing
        Set db = Nothing
    Else
        Cancel = True
        btnCancel_Click
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub
```

4.4.6 Η φόρμα «frmAddAppointment» - Νέο Ραντεβού

Η φόρμα «**Νέος Ασθενής**» ενεργοποιείται μέσα από την φόρμα «Επισκέψεις Ασθενών» και δίνει την δυνατότητα στον χρήστη να καταχωρεί στη βάση δεδομένων μια νέα επίσκεψη για έναν μόνο ασθενή και εμφανίζεται στην οθόνη ως **παράθυρο διαλόγου**. Ασφαλώς βασική προϋπόθεση για την σωστή λειτουργία της φόρμας είναι η επιλογή ενός ασθενή από την κύρια (Master) φόρμα



Εικόνα 32: Η φόρμα «Νέο Ραντεβού» (frmAddAppointment)

Εφόσον επιλεγθεί ο ασθενής, τότε στη φόρμα το πεδίο **ΑΜΚΑ** γεμίζει με την αντίστοιχη τιμή και το κρυφό πεδίο **txtPIId** κρατάει το **Id** του ασθενή. Για να πάρει τιμή το πεδίο ΑΜΚΑ, ως τιμή της ιδιότητας «Προέλευση στοιχείου ελέγχου» γράφουμε την εντολή:

```
=DLookup("[Patient]![ΑΜΚΑ]";"[Patient]";" [Patient]![Id] = " & [txtPIId])
```

Έπειτα ο χρήστης επιλέγει την ημερομηνία που θέλει ο ασθενής να κλείσει το ραντεβού και στη συνέχεια επιλέγει την ώρα από το σύνθετο πλαίσιο «**Ώρα**». Οι τιμές του πλαισίου που εμφανίζονται σε δυο στήλες, όπως φαίνονται στην παρακάτω εικόνα.

Εικόνα 33: Οι τιμές του πεδίου «Ώρα» με την ένδειξη εάν είναι διαθέσιμη ή όχι στον ασθενή.

Στην πρώτη στήλη εμφανίζονται οι **ώρες** του ιατρού που μπορεί να δεχθεί ο γιατρός μια επίσκεψη και στην δεύτερη στήλη εμφανίζεται η πληροφορία (**Διαθέσιμο / ΜΗ Διαθέσιμο**) για το εάν η ώρα είναι διαθέσιμη ή όχι. Με αυτό τον τρόπο ενημερώνεται ο χρήστης για την διαθεσιμότητα των ωρών που έχει μια μέρα και αποφεύγεται η καταχώρηση διπλότυπων τιμών στο σχεσιακό πίνακα «AppDetails». Η προέλευση των τιμών στο σύνθετο πλαίσιο τιμών (combobox) «cboAppHour» δίνεται από το παρακάτω ερώτημα:

SELECT

```

Format (AppHours.AppHour, "Short Time") AS Ώρες,
IIf (Exists (
    SELECT *
    FROM AppDetails
    INNER JOIN Appointment ON
        Appointment.Id = AppDetails.Appid
    WHERE Appointment.AppDate = AppDays.AppDate
        AND AppDetails.AppHour = AppHours.AppHour
), "Μη ΔΙΑΘΕΣΙΜΟ", "Διαθέσιμο") AS Κατάσταση
FROM AppDays, AppHours
WHERE AppDays.AppDate = [Forms]![frmAddAppointment]![txtAppDate];

```

Εφόσον ο χρήστης συμπληρώσει σωστά τα πεδία, προχωράει στην καταχώρηση της επίσκεψης, πατώντας το κουμπί «ΑΠΟΘΗΚΕΥΣΗ». Διαφορετικά, κλείνει το παράθυρο πατώντας το κουμπί «ΑΚΥΡΩΣΗ». Τα στοιχεία ελέγχου που χρησιμοποιούνται στην φόρμα είναι τα παρακάτω:

Πίνακας 31: Τα στοιχεία ελέγχου της φόρμας «Νέο Ραντεβού»

Στοιχεία Ελέγχου (Components)	Ονομασία Στοιχείου	Χρήση
Κουμπί (Button)	btnAddAppointment	Αποθήκευση ραντεβού
	btnCancel	Ακύρωση ενέργειας
Πλαίσιο Κειμένου (Textbox)	txtPAMKA	ΑΜΚΑ ασθενή

	txtAppDate	Ημέρα ραντεβού
	txtSubject	Θέμα επίσκεψης
	txtNotes	Παρατηρήσεις
	txtPid	Κωδ. Ασθενή (Κρυφό)
Σύνθετο Πλαίσιο (Combobox)	cboAppHour	Ώρα ραντεβού

Για την αποθήκευση ενός ραντεβού για έναν ασθενή στη βάση συντάξαμε μια ενσωματωμένη διαδικασία η οποία καλείται με το πάτημα του κουμπιού «**ΑΠΟΘΗΚΕΥΣΗ**». Η διαδικασία παίρνει ως είσοδο τις τιμές των πεδίων (παράμετροι) και καταχωρεί στους πίνακες «**Appointment**» και «**AppDetails**» της ΒΔ από μια νέα εγγραφή και ο κώδικας δίνεται παρακάτω:

```
Private Sub addAppointment(ByVal PID As Integer _
    , ByVal pdate As Variant _
    , ByVal phour As Variant _
    , ByVal psubject As String _
    , ByVal pnotes As Variant)

Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim sql, sql2 As String
Dim l_appid As Integer

On Error GoTo Err_Handler

Set db = CurrentDb
' Εισαγωγή στον πίνακα Appointment.
sql = "INSERT INTO Appointment"
sql = sql + "(Pid, AppDate, Subject, Notes)"
sql = sql + " VALUES('" & PID & "' , '" & pdate & "' , '" & psub-
ject & "' , '" & pnotes & "' )"

CurrentDb.Execute sql, dbFailOnError
' Εύρεση του τελευταίου πρωτεύοντος κλειδιού
' από τον πίνακα Appointment, πεδίο Id
Set rs = db.OpenRecordset("Appointment")
rs.MoveLast
l_appid = rs.Fields("Id")
rs.Close

' Εισαγωγή στον πίνακα AppDetails, σχεσιακό πίνακας (detail ta-
ble)
sql2 = "INSERT INTO AppDetails"
sql2 = sql2 + "(AppId, AppHour) VALUES('" & l_appid & "' , '" &
phour & "' )"

CurrentDb.Execute sql2, dbFailOnError

Result = MsgBox("Το ραντεβού καταχωρήθηκε με επιτυχία!", _
vbInformation + vbOKOnly, "Ραντεβού Ασθενούς")
```



```

If Result = vbOK Then
    DoCmd.Close acForm, "frmAddAppointment"
End If
CurrentDb.Close
Exit_Here:
Exit Sub
Err_Handler:
'DoCmd.Hourglass False
MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub

```

Τα πεδία «**Ημερομηνία**» και «**Ωρα**» είναι **υποχρεωτικά**. Σε περίπτωση που ο χρήστης δεν ορίσει τιμές στα συγκεκριμένα πεδία και πατήσει το κουμπί «**ΑΠΟΘΗΚΕΥΣΗ**» ενεργοποιούνται μηνύματα ενημέρωσης και σφάλματος. Επίσης, σε περίπτωση που επιλεγθεί ένα δεσμευμένο ραντεβού εμφανίζεται μήνυμα προειδοποίησης και η διαδικασία καταχώρησης σταματάει.

Εφόσον η καταχώρηση είναι επιτυχής, η φόρμα κλείνει και τα δεδομένα στην θυγατρική φόρμα των ραντεβού ανανεώνεται αυτόματα μέσω της εντολής:

```
Forms![frmPatient]![frmSubAppointment].Requery
```

4.4.7 Η φόρμα «frmEditAppointment» - Επεξεργασία Ραντεβού

Εικόνα 34: Η φόρμα «Επεξ/σια Ραντεβού» (frmEditAppointment)

Η φόρμα «**Επεξ/σια Ραντεβού**» ενεργοποιείται μέσα από την φόρμα «Επισκέψεις Ασθενών» με το πάτημα του κουμπιού «Επεξ/σία Επίσκεψης» και δίνει την δυνατότητα στον χρήστη να αλλάξει τα στοιχεία του ραντεβού ενός ασθενή στη βάση δεδομένων τροποποιώντας τις τιμές στα πεδία που επιθυμεί.

Ομοίως με την φόρμα «Επεξ/σια Ασθενή», σχεδιαστικά η φόρμα είναι παρόμοια με την προηγούμενη φόρμα της καταχώρησης ενός ραντεβού. Όταν ενεργοποιηθεί, εμφανίζεται ως **πλαίσιο διαλόγου** και τα πεδία περιέχουν τις τιμές της εγγραφής του ραντεβού που έχει επιλέξει ο χρήστης.

Τα στοιχεία της φόρμας γεμίζουν με τις τιμές της εγγραφής με την βοήθεια των πεδίων «txtPRAMKA» και «txtPId» όπως είδαμε και στην προηγούμενη φόρμα και ισχύουν οι ίδιοι κανόνες της συμπλήρωσης των υποχρεωτικών πεδίων. Εφόσον ο χρήστης κάνει τις αλλαγές στις τιμές των πεδίων που επιθυμεί πατάει το κουμπί «**ΕΝΗΜΕΡΩΣΗ**» και ολοκληρώνεται η ενημέρωση στους πίνακες «**Appointment**» και «**AppDetails**». Διαφορετικά, με το κουμπί «**ΑΚΥΡΩΣΗ**» κλείνει η φόρμα και ακυρώνεται η διαδικασία.

Για την ενημέρωση της εγγραφής στον πίνακα «Appointment» και «AppDetails», συντάξαμε τον παρακάτω κώδικα στο συμβάν «**Με το κλικ**» (OnClick) του κουμπιού «**ΕΝΗΜΕΡΩΣΗ**» (btnUpdateAppointment):

```
Private Sub btnUpdateAppointment_Click()
    Dim l_appdate as String
    Dim l_apphour as String
    Dim l_subject as String
    Dim l_notes as string
    Dim strCriteria As String
On Error GoTo Err_Handler

    Response = MsgBox("Θέλετε να αποθηκεύσετε τις αλλαγές ;", vbYesNo
+ vbQuestion, "medApp | Επεξεργασία Ραντεβού")

    If Response = vbYes Then
        l_appdate = Me.txtAppDate.Value
        l_apphour = Me.cboAppHour.Value
        l_subject = Me.txtSubject.Value
        l_notes = Me.txtNotes.Value

        strCriteria = " AppDate = #" & Format(l_appdate, "yyyy-mm-
dd") & "# And AppHour = #" & l_apphour & "#"
        ' Έλεγχος εάν τα πεδία ημερομηνίας και ώρας είναι συμπληρωμέ-
να.
        If IsNull(Me.txtAppDate) Or IsNull(Me.cboAppHour) Then
            MsgBox "Τα πεδία Ημερομηνία και Ωρα πρέπει να είναι συ-
μπληρωμένα κατάλληλα.", _
                vbCritical + vbOKOnly, "Καταχώρηση Ραντεβού"
```

```

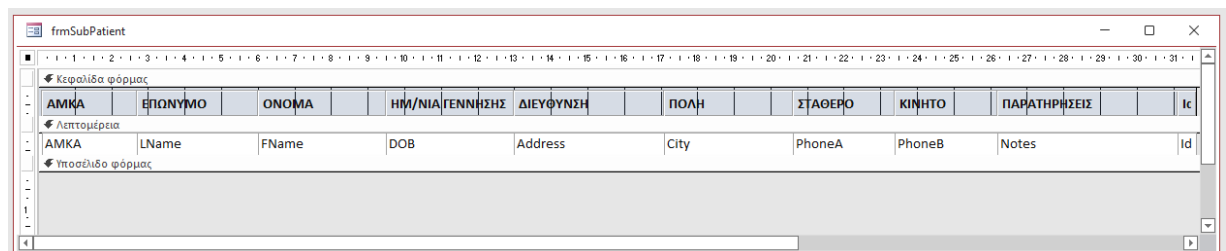
        GoTo Exit_Here

        ' Έλεγχος εάν υπάρχει ήδη καταχωρημένο ραντεβού στην ίδια ώρα
        και μέρα.
        ElseIf Not IsNull(DLookup("AppDate", "qryAppointmentHours",
strCriteria)) Then
            MsgBox "Το ραντεβού είναι κλεισμένο για άλλο πελάτη. Επι-
λέξτε από τη λίστα μόνο τα Διαθέσιμα!", _
                vbCritical + vbOKOnly, "Καταχώρηση Ραντεβού"
            GoTo Exit_Here
        Else
            sql = "UPDATE Appointment "
            sql = sql + "LEFT JOIN AppDetails ON Appointment.Id = Ap-
pDetails.AppId "
            sql = sql + "SET Appointment.AppDate = '" & l_appdate &
            "' "
            sql = sql + ", Appointment.Subject = '" & l_subject & "' "
            "
            sql = sql + ", Appointment.Notes = '" & l_notes & "' "
            sql = sql + ", AppDetails.AppHour = '" & l_apphour & "' "
            sql = sql + "WHERE Appointment.Id = " & app_id

            CurrentDb.Execute sql, dbFailOnError
            db.Close
            [Forms]![frmPatient]![frmSubAppointment].Requery
            MsgBox "Οι αλλαγές αποθηκεύτηκαν.", vbInformation +
vbOKOnly, "medApp | Επεξεργασία Ραντεβού"
            GoTo Exit_Here
        End If
    End If
Exit_Here:
    DoCmd.Close
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

4.4.8 Η φόρμα «frmSubPatient» - Προβολή Ασθενών (Κύρια)



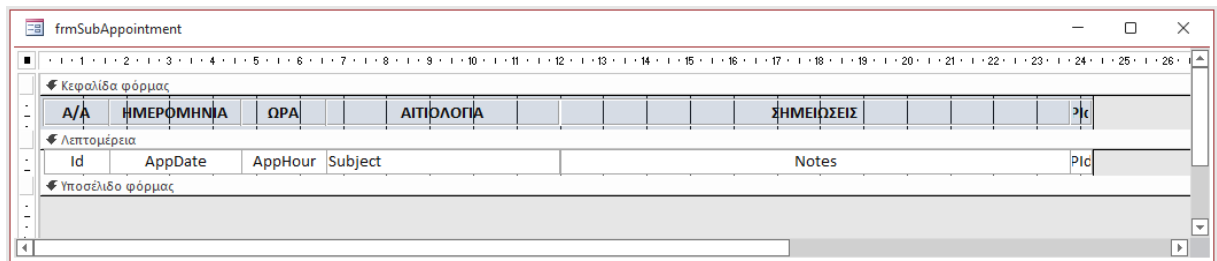
Εικόνα 35: Η υπο-φόρμα «Προβολή Ασθενών» (frmSubPatient)

Στη φόρμα «**Προβολή Ασθενών**» προβάλλονται τα στοιχεία των ασθενών και λειτουργεί ως η κύρια (**Master**) φόρμα προβολής της σχέσης ασθενής-ραντεβού/επισκέψεις. Έχει σχεδιαστεί ως φόρμα-πίνακας με **δεδομένα προέλευσης** το παρακάτω δυναμικό ερώτημα:

```
SELECT * FROM Patient ORDER BY Id;
```

Η φόρμα αυτή είναι το αντικείμενο δευτερεύουσας φόρμας (**frmSubPatient**) που χρησιμοποιήσαμε στην φόρμα «**Ραντεβού Ασθενών**». Το πεδίο «**Id**» έχει οριστεί να είναι κρυφό για καθαρά σχεδιαστικούς λόγους.

4.4.9 Η φόρμα «frmSubAppointment» - Προβολή Επισκέψεων (Θυγατρική)

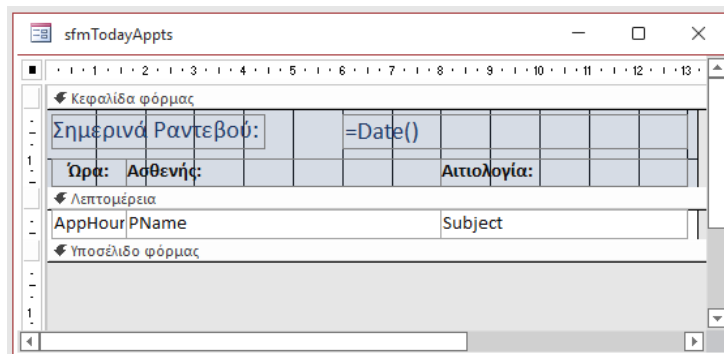


Εικόνα 36: Η υπο-φόρμα «Προβολή Ραντεβού Ασθενών» (frmSubAppointment)

Στη φόρμα «**Προβολή Επισκέψεων**» προβάλλονται οι λεπτομέρειες της επίσκεψης ενός ασθενή και λειτουργεί ως η θυγατρική-λεπτομέρεια (**detail**) φόρμα προβολής της σχέσης ασθενής-ραντεβού/επισκέψεις. Έχει σχεδιαστεί ως φόρμα-πίνακας με **δεδομένα προέλευσης** το ερώτημα «**qryAppointments**» που αναλύσαμε στην παράγραφο 4.2.3

Η φόρμα αυτή είναι το αντικείμενο δευτερεύουσας φόρμας (**frmSubAppointment**) που χρησιμοποιήσαμε στην φόρμα «**Ραντεβού Ασθενών**» και συνδέεται με την κύρια φόρμα «**frmSubPatient**» μέσω του πεδίου **PId**, όπως περιγράψαμε στην παράγραφο 4.4.3.

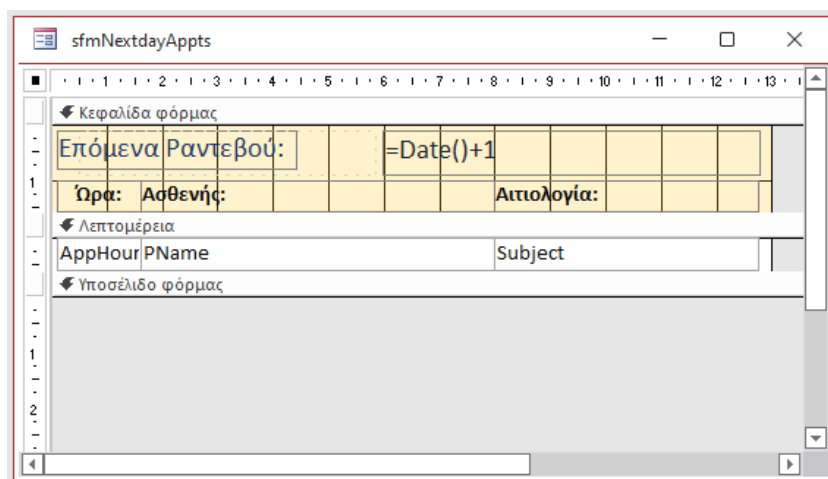
4.4.10 Η φόρμα «sfmTodayAppts» - Σημερινά Ραντεβού



Εικόνα 37: Η υπο-φόρμα «Σημερινά Ραντεβού» (sfmTodayAppts)

Στη φόρμα «**Σημερινά Ραντεβού**» προβάλλονται τα ραντεβού της τρέχουσας ημέρας. Έχει σχεδιαστεί ως φόρμα-πίνακας, είναι αντικείμενο-φόρμα στην κεντρική φόρμα επιλογών και προβάλλει τα δεδομένα του ερωτήματος «**qryTodayAppointments**», ταξινομημένα ανά ώρα επισκέψεως σε αύξουσα σειρά. Τα δεδομένα της φόρμας ανανεώνονται αυτόματα με την καταχώρηση μιας νέας επίσκεψης.

4.4.11 Η φόρμα «**sfmNextdayAppts**» - Επόμενα Ραντεβού



Εικόνα 38: Η υπο-φόρμα «Επόμενα Ραντεβού» (sfmNextdayAppts)

Στη φόρμα «**Επόμενα Ραντεβού**» προβάλλονται τα ραντεβού της επόμενης ημέρας. Έχει σχεδιαστεί ως φόρμα-πίνακας, είναι αντικείμενο-φόρμα στην κεντρική φόρμα επιλογών και προβάλλει τα δεδομένα του ερωτήματος «**qryNextDayAppointments**», ταξινομημένα ανά ώρα επισκέψεως σε αύξουσα σειρά. Τα δεδομένα της φόρμας ανανεώνονται αυτόματα με την καταχώρηση μιας νέας επίσκεψης.

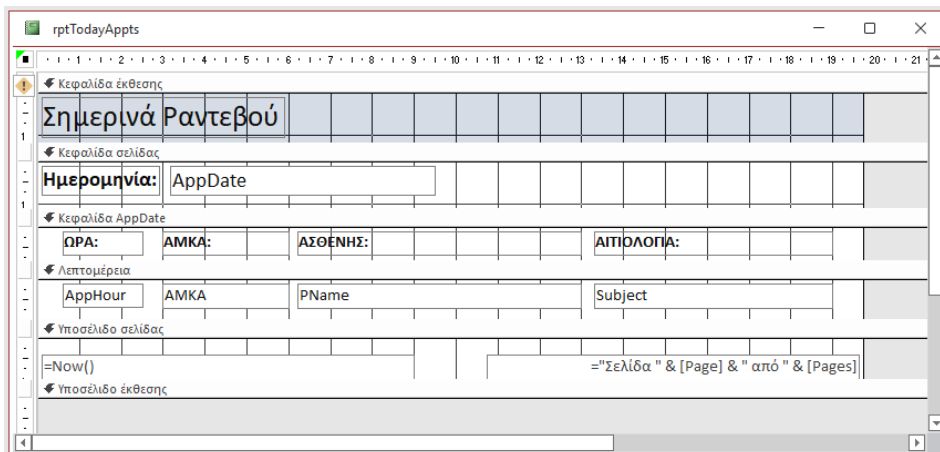
4.5 Οι Εκθέσεις (Αναφορές)

Οι εκθέσεις ή αναφορές (Reports) είναι ένα χρήσιμο εργαλείο που προσφέρεται σε ένα ΣΔΒΔ που μας βοηθούν στην έντυπη παρουσίαση των δεδομένων της ΒΔ. Στην MS Access, οι εκθέσεις μοιάζουν πολύ με τις φόρμες, καθώς αποτελούνται και αυτές από στοιχεία ελέγχου, με τη διαφορά που δεν επιτρέπεται στο χρήστη να κάνει ενημερώσεις στις εγγραφές.

Όπως και μια φόρμα, μια έκθεση μπορεί να έχει ως προέλευση δεδομένων έναν πίνακα, ένα ερώτημα ή ακόμα μια πρόταση SQL και μπορεί να προβάλλει αυτά τα δεδομένα σε συνοπτική ή αναλυτική περιγραφή. Επίσης μια έκθεση μπορεί να περιέχει και διαγραμματικές απεικονίσεις των δεδομένων.

Στην εφαρμογή μας θα παρουσιάσουμε ενδεικτικά 3 βασικές εκθέσεις οι οποίες προκύπτουν εύκολα από ερωτήματα που έχουμε περιγράψει στις προηγούμενες παραγράφους και οι οποίες είναι χρήσιμες για τον χρήστη στη λήψη αποφάσεων που αφορούν την λειτουργία του ιατρού. Για την δημιουργία των εκθέσεων χρησιμοποιήσαμε τον ενσωματωμένο οδηγό εκθέσεων της MS Access, από την καρτέλα «**Δημιουργία**» → «**Εκθέσεις**» → «**Οδηγός εκθέσεων**» και σχεδιάσαμε τις παρακάτω εκθέσεις.

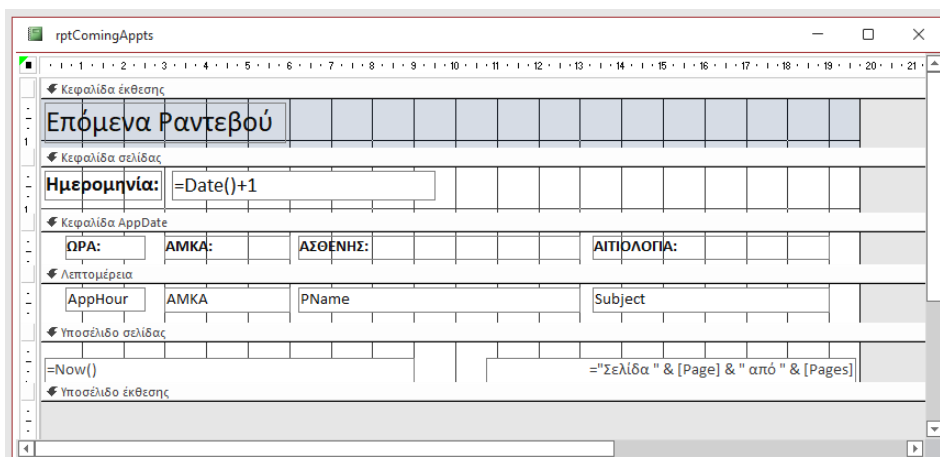
4.5.1 Η έκθεση «rptTodayAppts» - Σημερινά Ραντεβού



Εικόνα 39: Η έκθεση «Σημερινά Ραντεβού» (rptTodayAppts)

Στην έκθεση «**Σημερινά Ραντεβού**» εκτυπώνονται τα ραντεβού της τρέχουσας ημέρας, ταξινομημένα ανά ώρα επίσκεψης σε αύξουσα σειρά. Τα δεδομένα της έκθεσης προέρχονται από το ερώτημα «**qryTodayAppointments**». Είναι διαθέσιμη **α)** στην αρχική φόρμα, στην καρτέλα «**Ραντεβού**» και ενεργοποιείται από το κουμπί «Εκτύπωση αναφοράς» και **β)** στην καρτέλα «Εκτυπώσεις» πατώντας την ετικέτα «**Ημερήσια Ραντεβού**».

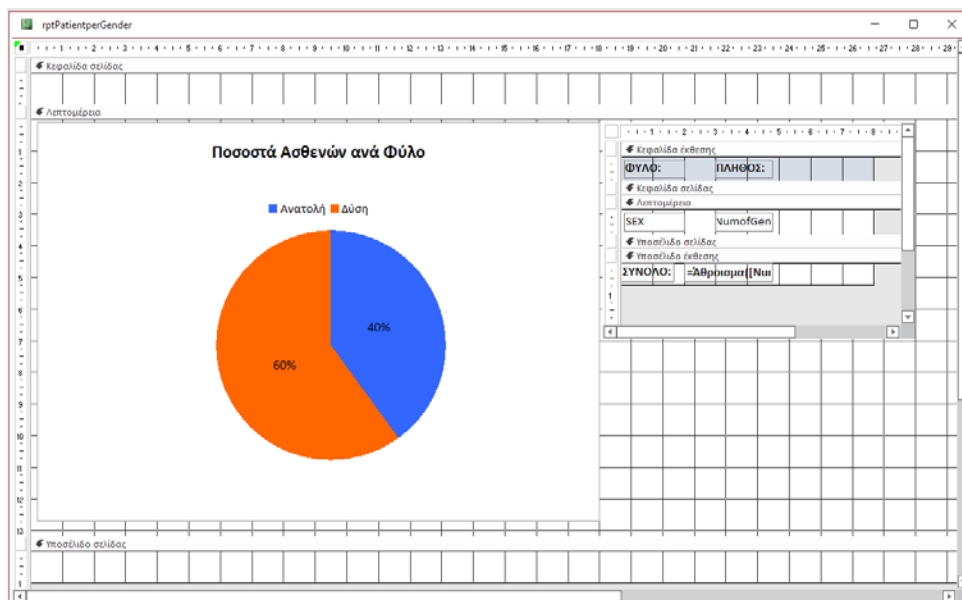
4.5.2 Η έκθεση «rptComingAppts» - Επόμενα Ραντεβού



Εικόνα 40: Η αναφορά «Επόμενα Ραντεβού» (rptNextdayAppt)

Στην έκθεση «**Επόμενα Ραντεβού**» εκτυπώνονται τα ραντεβού της επόμενης ημέρας, ταξινομημένα ανά ώρα επίσκεψης σε αύξουσα σειρά. Τα δεδομένα της έκθεσης προέρχονται από το ερώτημα «**qryNextDayAppointments**». Είναι διαθέσιμη στην αρχική φόρμα, στην καρτέλα «Ραντεβού» και ενεργοποιείται από το κουμπί «Εκτύπωση αναφοράς».

4.5.3 Η έκθεση «rptPatientGender» - Ασθενείς ανά Φύλο

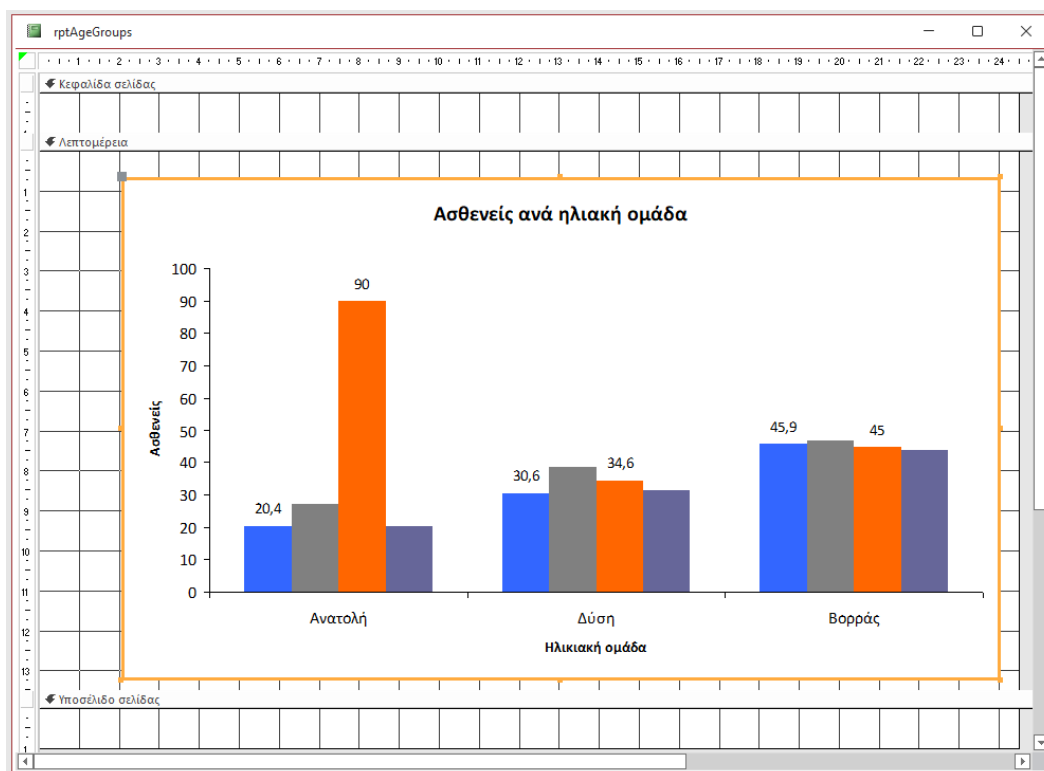


Εικόνα 41: Η αναφορά «Ασθενείς ανά Φύλο» (rptPatientperGender)

Στην έκθεση «**Ασθενείς ανά φύλο**» αποτυπώνονται σε μορφή γραφήματος πίτας τα ποσοστά των ασθενών του ιατρείου ανά φύλο. Τα δεδομένα του γραφήματος προέρχονται από το ερώτημα «**qryPatientGender**» και σχεδιάστηκε με την βοήθεια του στοιχείου ελέγχου «Γράφημα» (Graph), ενώ παράλληλα παρέχεται και η προέλευση δεδομένων του γραφήμα-

τος σε μορφή πίνακα-έκθεση. Η αναφορά αυτή είναι διαθέσιμη στην Αρχική φόρμα, στην καρτέλα «Εκτυπώσεις» και ενεργοποιείται από το κουμπί «Ασθενείς ανά φύλο».

4.5.4 Η έκθεση «rptPatientAgeGroups» - Ηλικιακές ομάδες ασθενών



Εικόνα 42: Η αναφορά «Ηλικιακές ομάδες ασθενών» (rptPatientAgeGroups)

Στην έκθεση «**Ηλικιακές ομάδες ασθενών**» αποτυπώνονται σε γράφημα τύπου ράβδου ο αριθμός των ασθενών ανά ηλικιακή ομάδα. Τα δεδομένα του γραφήματος προέρχονται από το ερώτημα «**qryAgeGroups**» και σχεδιάστηκε με την βοήθεια του στοιχείου ελέγχου «Γράφημα» (Graph). Η αναφορά αυτή είναι διαθέσιμη στην Αρχική φόρμα, στην καρτέλα «Εκτυπώσεις» και ενεργοποιείται από το κουμπί «**Ηλικιακές ομάδες**».

4.6 Οι Λειτουργικές Μονάδες

Όπως περιγράψαμε και σε προηγούμενο κεφάλαιο, οι λειτουργικές μονάδες (Modules) της MS Access είναι εντολές κώδικα, συναρτήσεις και διαδικασίες (υπο-ρουτίνες) που αποθηκεύονται σε ένα ενιαίο πακέτο εντολών. Μοιάζουν με τις μακροεντολές, αλλά παρέχουν περισσότερη λειτουργικότητα στην εφαρμογή και μπορούν να κληθούν από οποιοδήποτε αντικείμενο της εφαρμογής

Στην εφαρμογή μας, χρησιμοποιούμε τρεις λειτουργικές μονάδες. Οι δύο από αυτές αφορούν συναρτήσεις και εντολές της οντότητας «Ασθενής» και «Ραντεβού», αντίστοιχα ενώ η Τρίτη περιέχει μεταβλητές απεριόριστης εμβέλειας, χρήσιμες για την λειτουργία της εφαρμογής.

4.6.1 Η μονάδα «mdlPatient»

Περιέχει την συνάρτηση «**validateAMKA**» που επιστρέφει μια λογική τιμή (TRUE ή FALSE). Δέχεται ως είσοδο τον AMKA ενός ασθενή και ελέγχει εάν ο αριθμός αυτός περιέχει **μόνο ψηφία** και εάν έχει μέγεθος **11 χαρακτήρων**. Καλείται στο συμβάν «Μετά από ενημέρωση» (AfterUpdate event) του πεδίου κειμένου (textbox) «AMKA» στις φόρμες «Νέος Ασθενής» και «Επεξ/σια Ασθενή». Ο κώδικας της συνάρτησης δίνεται παρακάτω:

```
Public Function validateAMKA(pAMKA As String) As Boolean
On Error GoTo Err_Handler
    If Len(pAMKA) <> 11 Then
        validateAMKA = False
        Exit Function
    ElseIf IsNumeric(pAMKA) = False Then
        validateAMKA = False
        Exit Function
    Else
        validateAMKA = True
        Exit Function
    End If
Exit_Here:
    Exit Function
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
    Resume Exit_Here
End Function
```

4.6.2 Η μονάδα «mdlAppointment»

Περιέχει την διαδικασία «loadCalendar» που δέχεται ως είσοδο το τρέχον έτος ή την τιμή ενός έτους που επιθυμεί ο χρήστης και προσθέτει εγγραφές στον πίνακα «Appdays», δηλαδή τις ημέρες του έτους, το οποίο όρισε ο χρήστης ως τιμή παραμέτρου στην φόρμα «Ημέρες εργασίας» (frmWorkdays). Η διαδικασία ελέγχει εάν υπάρχουν ήδη καταχωρημένες οι ημέρες του έτους πριν προχωρήσει στην προσθήκη των εγγραφών. Εάν υπάρχουν εμφανίζεται κατάλληλο μήνυμα και η διαδικασία διακόπτεται. Ο κώδικας της συνάρτησης δίνεται παρακάτω:

```
Public Sub loadCalendar(pyear As Integer)
    Dim db As DAO.Database
    Dim dte As Date
    Dim dteLast As Date
    Dim intYear As Integer
    Dim rs As DAO.Recordset
    Dim strMsg As String
On Error GoTo ErrorHandler
    intYear = IIf(pyear = 0, Year(Date), pyear)
    dte = DateSerial(intYear, 1, 1)
    dteLast = DateSerial(intYear, 12, 31)

    DoCmd.SetWarnings True

    Set db = CurrentDb
    Set rs = db.OpenRecordset("AppDays", dbOpenDynaset, dbAppendOnly)

    Do While dte <= dteLast
        rs.AddNew
        rs.Fields("AppDate").Value = dte
        rs.Update
        dte = dte + 1
    Loop
    MsgBox "Οι ημέρες για το έτος " & pyear & " προστέθηκαν με επιτυχία!", vbOKOnly + vbInformation, "medApp | Διαχείριση Ημερών"
    rs.Close
Exit_Here:
    Set rs = Nothing
    Set db = Nothing
    Exit Sub
ErrorHandler:
    Select Case Err.Number
        Case 3022
            MsgBox "Υπάρχουν ήδη καταχωρημένες ημέρες για αυτό το έτος! " & vbNewLine & _
                "Για να συνεχίσετε αλλάξτε την τιμή του έτους στο πεδίο.", _
                vbCritical + vbOKOnly, "medApp | Σφάλμα Καταχώρησης"
            Resume Exit_Here
        Case Else
    End Select
End Sub
```

```

        MsgBox Err.Number & " " & Err.Description, _
            vbExclamation, "medApp | Σφάλμα Εφαρμογής"
        Resume Exit_Here
    End Select
End Sub

```

4.6.3 Η μονάδα «gblVariables»

Περιέχει τις δηλώσεις των μεταβλητών της εφαρμογής που έχουν απεριόριστη εμβέλεια και βοηθούν στην ομαλή λειτουργία της εφαρμογής. Οι δηλώσεις των μεταβλητών αυτών είναι οι παρακάτω:

```

Option Compare Database
Option Explicit
Public userID As Variant
Public userType As Variant
Public userName As Variant
Public l_amka As String
Public p_id As Variant
Public app_id As Variant

```

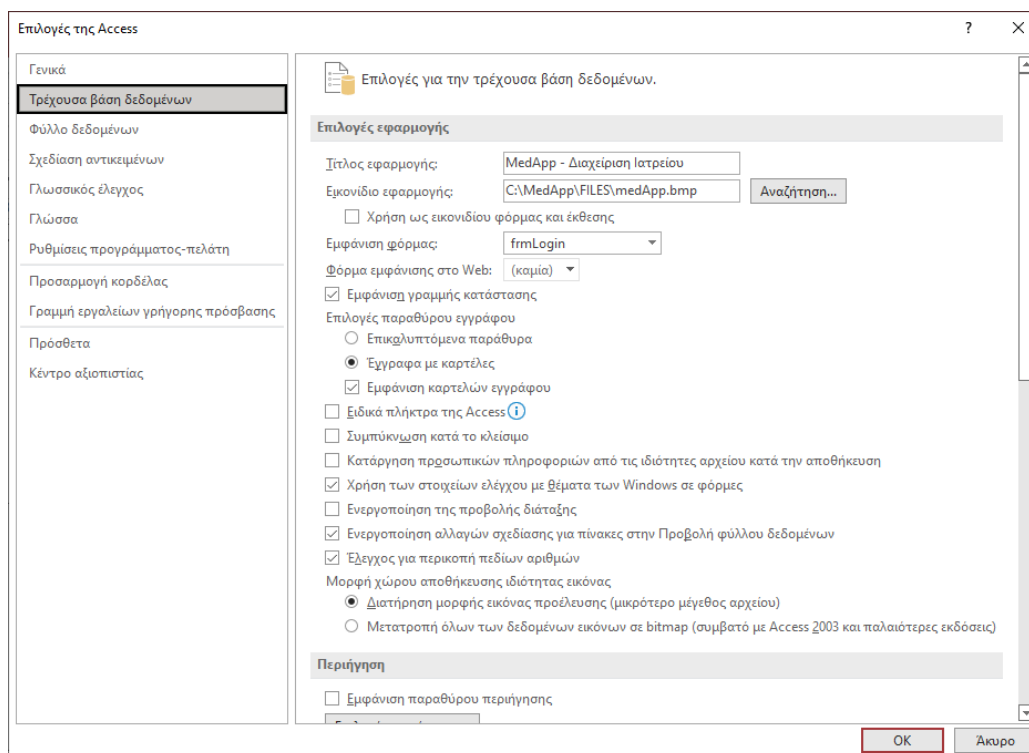
4.7 Παραγωγή του εκτελέσιμου αρχείου

Το εκτελέσιμο αρχείο της MS Access (*.accde) είναι ουσιαστικά ένα αρχείο βάσης δεδομένων της Access που χρησιμοποιείται για την προστασία ενός αρχείου *.accdb ώστε να μην είναι δυνατή η επεξεργασία του σε επίπεδο κώδικα VBA και αντικειμένων (φόρμες, πίνακες, κτλ.). Με αυτό τον τρόπο εξασφαλίζεται η εύρυθμη εκτέλεση της εφαρμογής χωρίς να απαιτείται η εγκατάσταση της MS Access στον υπολογιστή που θα εκτελείται η εφαρμογή παρά μόνο το πακέτο εκτέλεσης **Microsoft Access Runtime**.

Για να δημιουργήσουμε το τελικό εκτελέσιμο αρχείο της εφαρμογής, πρέπει αρχικώς να ρυθμίσουμε την εφαρμογή έτσι ώστε να απενεργοποιηθούν οποιοσδήποτε ενέργειες μπορεί να βλάψουν το αρχείο εφαρμογής, όπως οι καρτέλες μενού, το παράθυρο περιήγησης, τα μενού επιλογών, κτλ. επομένως ακολουθούμε τα παρακάτω βήματα

Έχοντας ανοιχτό το αρχείο «medApp.accdb» επιλέγουμε:

- Από την καρτέλα «**Αρχείο**» → «**Επιλογές**».
- Από το παράθυρο διαλόγου «**Επιλογές της Access**» επιλέγουμε από τη δεξιά στήλη την επιλογή «**Τρέχουσα βάση δεδομένων**» (Εικόνα 43).



Εικόνα 43: Οι ρυθμίσεις επιλογών ενός αρχείου MS Access

- Στο παράθυρο διαλόγου από την ενότητα «Επιλογές εφαρμογής» ρυθμίζουμε τα πεδία με τις παρακάτω τιμές:

Πίνακας 32: Οι ρυθμίσεις των πεδίων «Επιλογές εφαρμογής»

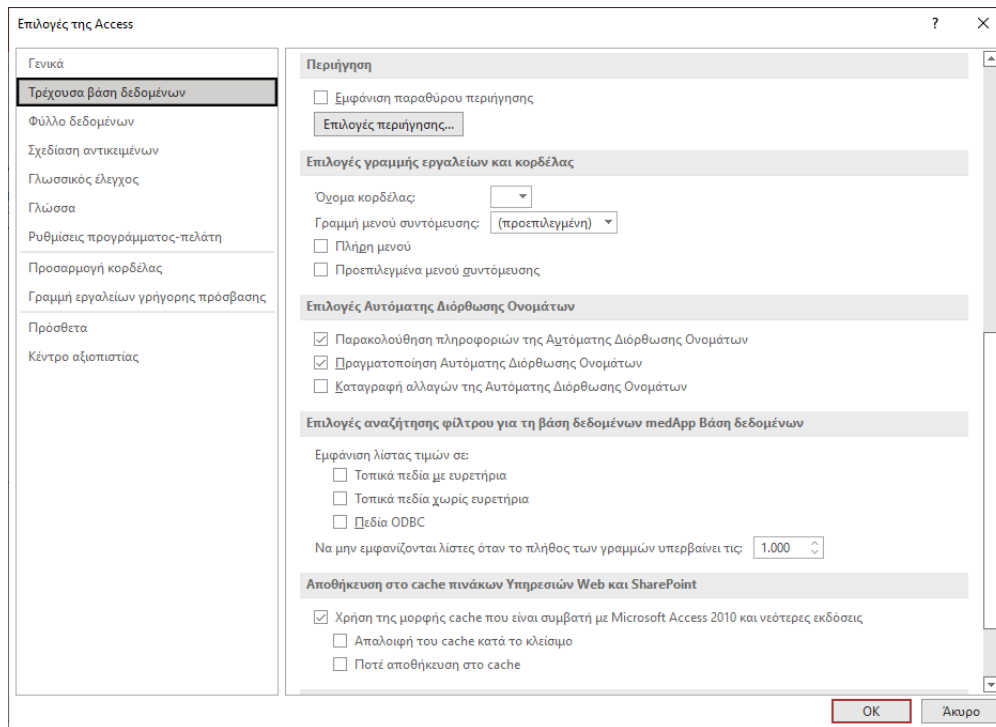
Τίτλος εφαρμογής:	MedApp – Διαχείριση Ιατρού
Εμφάνιση φόρμας:	frmLogin Η πρώτη φόρμα που εμφανίζεται κατά την εκτέλεση της εφαρμογής.
Ειδικά πλήκτρα Access:	Απενεργοποιημένο
Ενεργοποίηση της προβολής διάταξης:	Απενεργοποιημένο

- Έπειτα, στις ενότητες «Περιήγηση» και «Επιλογές γραμμής εργαλείων και κορδέλα» (Εικόνα 44) ρυθμίζουμε τα πεδία με τις παρακάτω τιμές:

Πίνακας 33: Οι ρυθμίσεις που αφορούν την Περιήγηση και τις επιλογές των γραμμών εργαλείων

Εμφάνιση παραθύρου περιήγησης:	Απενεργοποιημένο
Πλήρη μενού:	Απενεργοποιημένο
Προεπιλεγμένα μενού συντόμευσης:	Απενεργοποιημένο
Ενεργοποίηση της προβολής διάταξης:	Απενεργοποιημένο

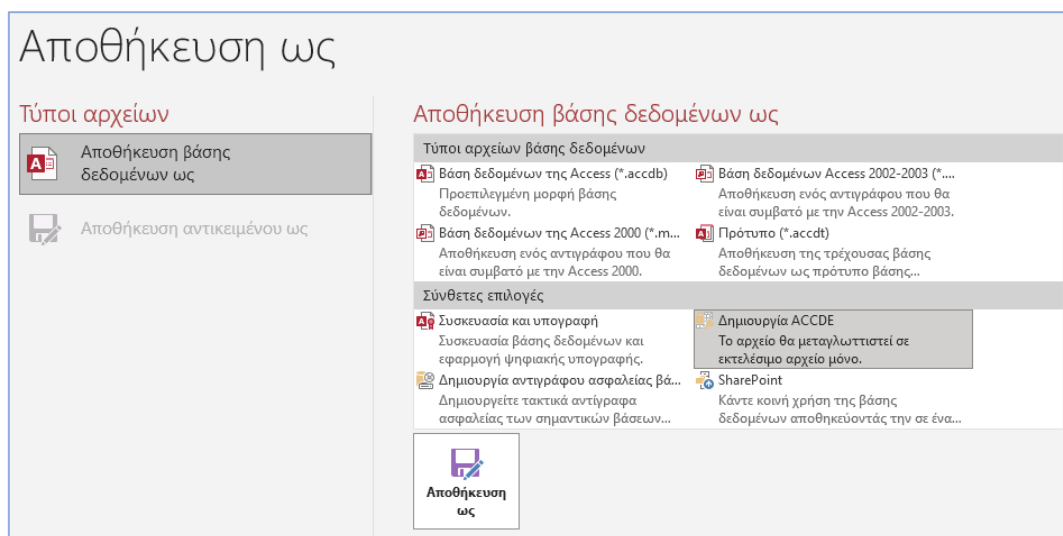
- Οι παραπάνω ρυθμίσεις θα πρέπει να ολοκληρωθούν όπως φαίνονται στην παρακάτω εικόνα:



Εικόνα 44: Οι ρυθμίσεις επιλογών για τις ενότητες «Περιήγηση» και «Επιλογές γραμμής εργαλείων και κορδέλας»

- Ολοκληρώνουμε τις απαραίτητες ρυθμίσεις πατώντας το κουμπί «OK». Θα μας ζητηθεί να επανεκκινήσουμε την MS Access ώστε οι ρυθμίσεις να εφαρμοστούν, το οποίο το παραβλέπουμε.

Έπειτα από το μενού «Αρχείο» επιλέγουμε: «Αποθήκευση Ως» → «Δημιουργία ACCDE» και δημιουργούμε το εκτελέσιμο αρχείο στο φάκελο «C:\medApp» με το ίδιο όνομα που έχουμε το αρχείο των φορμών (**medApp**), όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 45: Η δημιουργία του εκτελέσιμου αρχείου γίνεται μέσω της επιλογής «Αποθήκευση Ως»

Στο κεφάλαιο που ακολουθεί θα περιγράψουμε τις βασικές λειτουργίες της εφαρμογής, παραδίνοντας κατά κάποιον τρόπο τις οδηγίες χρήσεως της εφαρμογής για τον χρήστη, πως θα κάνει την σύνδεση, πώς θα καταχωρεί τα δεδομένα στη βάση, τα μηνύματα και τις προειδοποιήσεις που θα εμφανίζονται, κτλ.

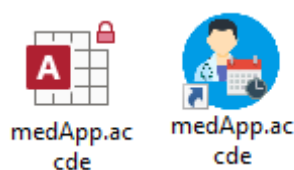
Για περισσότερα στοιχεία σχετικά με τον τρόπο λειτουργιών των στοιχείων ελέγχου που χρησιμοποιήθηκαν στις φόρμες της εφαρμογής, τα συμβάντα κώδικα που ενεργοποιούνται σε ορισμένα και γενικά ο συνολικός κώδικας VBA της εφαρμογής με την λειτουργία του κάθε αντικειμένου, παρατίθενται στο κεφάλαιο «ΠΑΡΑΡΤΗΜΑΤΑ».

5 Τρόπος λειτουργίας της εφαρμογής

Κεφάλαιο 5: Τρόπος λειτουργίας της εφαρμογής

Θεωρούμε ότι το ιατρείο διαθέτει έναν υπολογιστή (επιτραπέζιο ή φορητό) και ο φάκελος που περιέχει τα αρχεία της εφαρμογής (δεδομένα, εφαρμογή και εκτελέσιμο αρχείο) βρίσκεται εγκατεστημένος στην τοποθεσία που αναφέραμε στο προηγούμενο κεφάλαιο.

Το εκτελέσιμο αρχείο μπορούμε να το εμφανίσουμε στην επιφάνεια εργασίας ως συντόμευση της εφαρμογής. Επίσης μπορούμε να αλλάξουμε και το εικονίδιο της εφαρμογής μέσα από τις ιδιότητες του αρχείου των Windows. Το τελικό εικονίδιο του εκτελέσιμου αρχείου (αριστερά) με το εικονίδιο της συντόμευσης (δεξιά) είναι το παρακάτω:



5.1 Σύνδεση στην εφαρμογή

Κάνουμε διπλό κλικ στο εικονίδιο και ξεκινάμε την εφαρμογή. Ανοίγει το περιβάλλον της MS Access και εμφανίζεται η φόρμα «Σύνδεση Χρήστη», όπως φαίνεται παρακάτω:

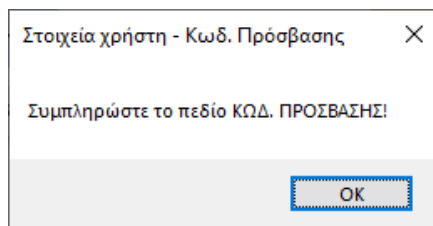
MedApp - Σύνδεση Χρήστη	
Σύνδεση Χρήστη:	
Χρήστης:	<input type="text" value="admin"/>
Κωδ. Πρόσβασης:	<input type="password" value="*****"/>
<input type="button" value="ΣΥΝΔΕΣΗ"/>	<input type="button" value="ΕΞΟΔΟΣ"/>
2022 MedApp	Δευτέρα, 9 Μαΐου 2022

Εικόνα 46: Η φόρμα Σύνδεση Χρήστη είναι η πρώτη φόρμα που εμφανίζεται κατά την εκτέλεση της εφαρμογής.

Εάν δεν συμπληρωθούν τα πεδία «Χρήστης» ή «Κωδ. Πρόσβασης» και ο χρήστης πατήσει «ΣΥΝΔΕΣΗ» εμφανίζονται τα παρακάτω μηνύματα προειδοποίησης αντίστοιχα:

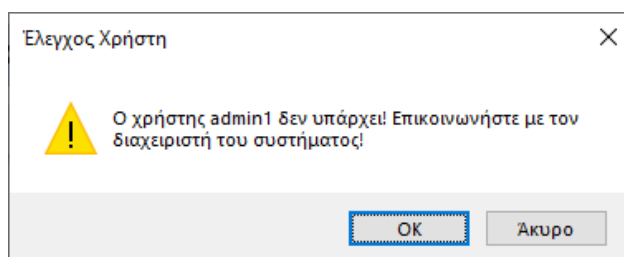
Στοιχεία χρήστη - Χρήστης	
Συμπληρώστε το πεδίο ΧΡΗΣΤΗΣ!	
<input type="button" value="OK"/>	

Εικόνα 47: Μήνυμα προειδοποίησης για κενή τιμή στο πεδίο «Χρήστης»



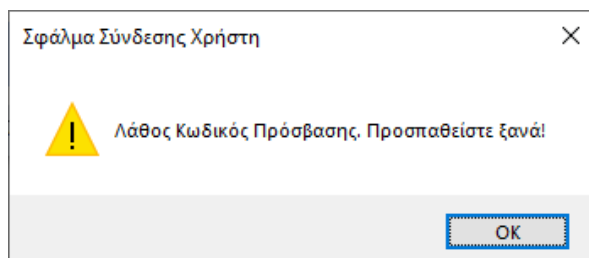
Εικόνα 48: Μήνυμα προειδοποίησης για κενή τιμή στο πεδίο «Κωδ. Πρόσβασης»

Επίσης στο πεδίο «Χρήστης» γίνεται έλεγχος για το εάν υπάρχει το όνομα χρήστη καταχωρημένο μέσα στην βάση. Σε περίπτωση που δεν υπάρχει εμφανίζεται το παρακάτω μήνυμα:




Εικόνα 49: Μήνυμα ελέγχου ταυτότητας χρήστη (username) στη ΒΔ

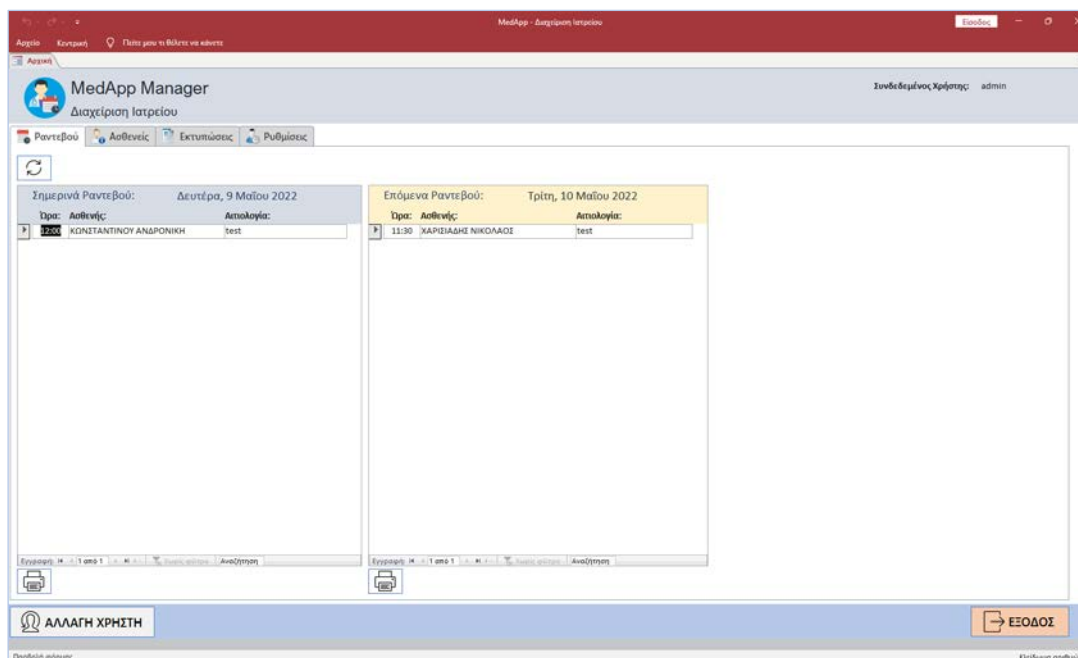
Εφόσον έχουν συμπληρωθεί κατάλληλα όλα τα παραπάνω πεδία ο χρήστης πατάει το κουμπί «ΣΥΝΔΕΣΗ». Σε περίπτωση που πληκτρολόγησε λάθος κωδικό πρόσβασης η εφαρμογή τον ενημερώνει κατάλληλα με σχετικό μήνυμα:



Εικόνα 50: Μήνυμα προειδοποίησης λάθους κωδικού πρόσβασης.

Διαφορετικά εάν έχουν εισαχθεί σωστά τα στοιχεία του χρήστη στα πεδία, η φόρμα κλείνει και ανοίγει η κεντρική φόρμα επιλογών (Αρχική) της εφαρμογής. Η Αρχική φόρμα φορτώνεται εξ' ολοκλήρου σε όλη την οθόνη προσφέροντας έτσι στον χρήστη έναν εύχρηστο περιβάλλον εργασίας όπως φαίνεται στην παρακάτω εικόνα.

Η πρώτη καρτέλα «**Ραντεβού**» προβάλλει τα ραντεβού της τρέχουσας και της επόμενης ημέρας. Δίπλα από κάθε προβολή δίνεται στο χρήστη η **δυνατότητα εκτύπωσης** τους μέσω του κουμπιού 



Εικόνα 51: Η Αρχική οθόνη επιλογών

Πατώντας το κουμπί «**ΑΛΛΑΓΗ ΧΡΗΣΤΗ**» η αρχική φόρμα κλείνει και ανοίγει ξανά η φόρμα «Σύνδεση Χρήστη» για την σύνδεση διαφορετικού χρήστη. Για τον τερματισμό της εφαρμογής ο χρήστης πατάει το κουμπί «**ΕΞΟΔΟΣ**». Είναι οργανωμένη σε 4 καρτέλες, που η κάθε μία περιέχει τις βασικές και δευτερεύουσες λειτουργίες της εφαρμογής. Αυτές είναι:

1. Ραντεβού

- α. Προβολή ραντεβού τρέχουσας και επόμενης ημέρας.
- β. Εκτύπωση ραντεβού τρέχουσας και επόμενης ημέρας.

2. Ασθενείς

- α. Ραντεβού Ασθενών
- β. Νέος Ασθενής

3. Εκτυπώσεις

- α. Ημερήσιες επισκέψεις
- β. Ασθενείς ανά φύλο
- γ. Ηλικιακές ομάδες

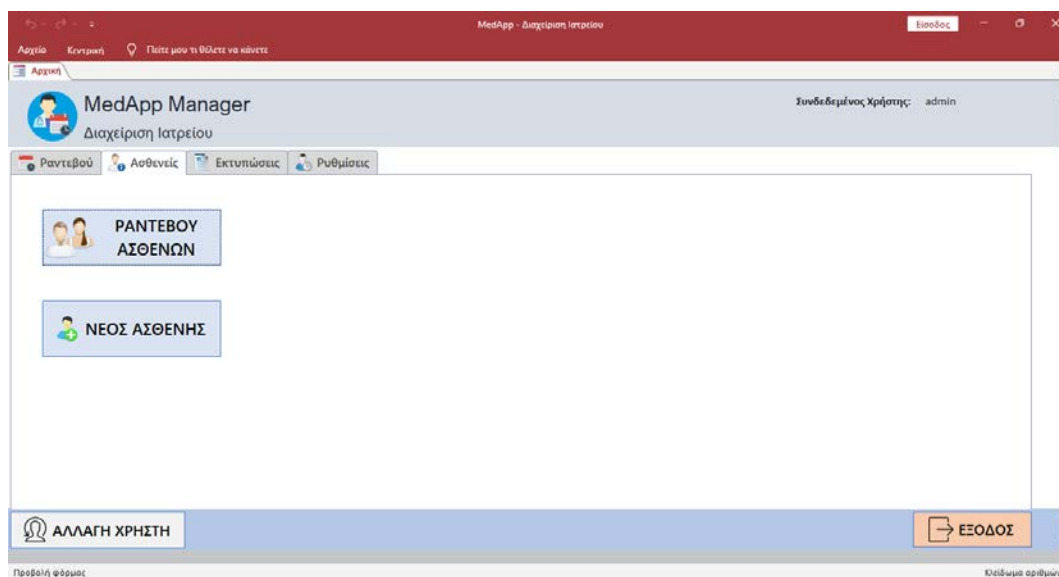
4. Ρυθμίσεις

- α. Ημέρες Εργασίας / Ωρες
- β. Διαχείριση Χρηστών

5.2 Καταχώρηση ασθενή

Η διαδικασία καταχώρησης ενός νέου ασθενή μπορεί να γίνει με δύο τρόπους:

- α. Απευθείας μέσω του κουμπιού «**ΝΕΟΣ ΑΣΘΕΝΗΣ**» στην καρτέλα της Αρχικής φόρμας «**ΑΣΘΕΝΕΙΣ**».
- β. Μέσω της φόρμας «**Ραντεβού Ασθενών**», όπως θα αναφέρουμε παρακάτω.

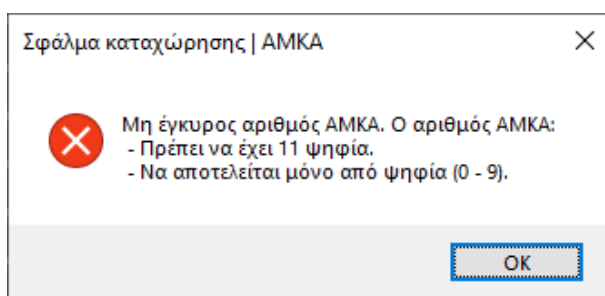


Εικόνα 52: Η καρτέλα «Ασθενείς» της αρχικής φόρμας.

Και στις δυο περιπτώσεις η διαδικασία είναι η ίδια και δεν απαιτείται κάποια ιδιαίτερη προσοχή. Από την καρτέλα «**Ασθενείς**» της αρχικής φόρμας, ο χρήστης πατάει το κουμπί «**ΝΕΟΣ ΑΣΘΕΝΗΣ**» και ανοίγει η φόρμα καταχώρησης νέου ασθενή.

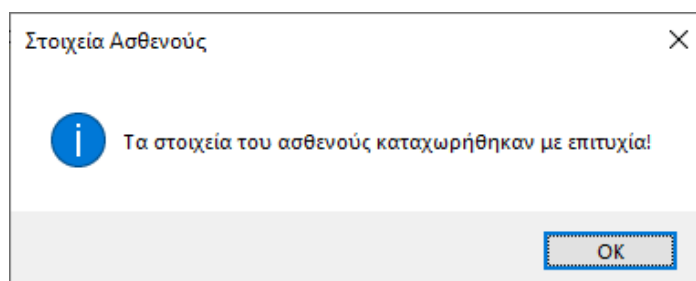
Εικόνα 53: Η φόρμα καταχώρησης ενός νέου ασθενή στη ΒΔ.

Το πεδίο «**ΑΜΚΑ**» είναι υποχρεωτικό και δίνεται η δυνατότητα στο χρήστη να καταχωρήσει τον ασθενή στη ΒΔ μόνο με το ΑΜΚΑ του και να προσθέσει αργότερα τα στοιχεία του (π.χ. κατά την επίσκεψη του στο ιατρείο). Κατά την διάρκεια της πληκτρολόγησης του ΑΜΚΑ γίνεται έλεγχος για την ορθότητα του, δηλαδή εάν είναι **11-ψηφιος αριθμός** ή εάν περιέχει μόνο **αριθμητικούς χαρακτήρες**. Εάν δεν τηρείται ένας από τους δύο κανόνες ή και οι δυο μαζί τότε εμφανίζεται το παρακάτω προειδοποιητικό μήνυμα:



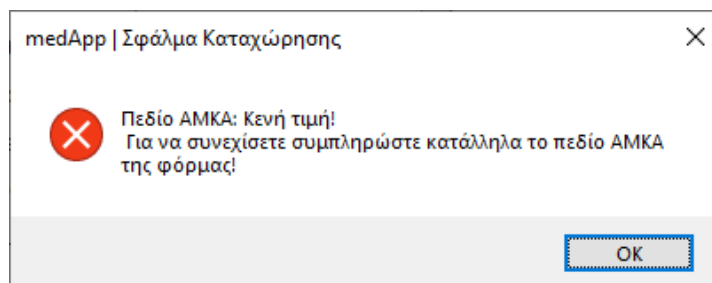
Εικόνα 54: Μήνυμα προειδοποίησης για την ορθότητα του ΑΜΚΑ

Μόλις συμπληρωθεί ένας έγκυρος ΑΜΚΑ, το πεδίο «Ημερομηνία Γέννησης» συμπληρώνεται αυτόματα. Ο χρήστης καταχωρεί τα υπόλοιπα στοιχεία στα πεδία και πατάει το κουμπί «**ΑΠΟΘΗΚΕΥΣΗ**» ενώ εάν επιθυμεί να ακυρώσει την διαδικασία πατάει το κουμπί «**ΑΚΥΡΩΣΗ**». Εφόσον το πεδίο του «ΑΜΚΑ» έχει συμπληρωθεί εμφανίζεται το παρακάτω μήνυμα που ενημερώνει το χρήστη ότι ολοκληρώθηκε επιτυχώς η καταχώρηση.



Εικόνα 55: Μήνυμα επιτυχούς καταχώρησης του ασθενή στη ΒΔ

Διαφορετικά εάν δεν συμπληρωθεί ο ΑΜΚΑ εμφανίζεται το παρακάτω μήνυμα:



Εικόνα 56: Μήνυμα σφάλματος για μη καταχώρηση ΑΜΚΑ στο αντίστοιχο πεδίο

5.3 Καταχώρηση νέου ραντεβού

Η διαδικασία καταχώρησης ενός νέου ραντεβού (επίσκεψης) γίνεται μόνο μέσω της φόρμας «**Επισκέψεις Ασθενών**». Ο χρήστης πατάει το κουμπί «**PANTEBOY ΑΣΘΕΝΩΝ**» στην καρτέλα «**Ασθενείς**» και ανοίγει η παρακάτω φόρμα.

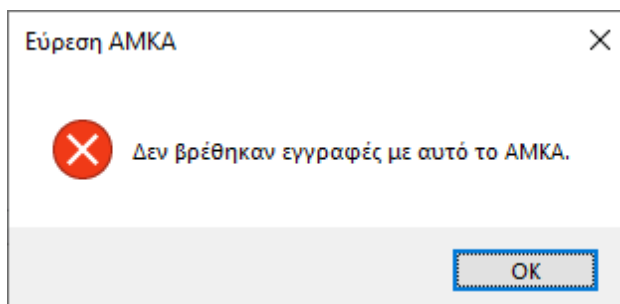
ΑΜΚΑ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΗΜ/ΝΙΑ ΓΕΝΝΗΣΗΣ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΣΤΑΘΕΡΟ	ΚΙΝΗΤΟ	ΠΑΡΑΤΗΡΗΣΙΣ
02128904500	ΤΣΑΚΙΡΗΣ	ΕΥΑΓΓΕΛΟΣ	2/12/1989	ΚΟΛΟΚΟΤΡΩΝΗ 23	2467083210			
30098102331	ΧΑΡΙΣΙΑΔΗΣ	ΝΙΚΟΛΑΟΣ	30/9/1981	ΟΡΕΣΤΟΥ 9	ΑΡΓΟΣ ΟΡΕΣΤΙΚΟ	2467088999		
05089204600	ΠΑΠΑΜΙΧΑΗΛ	ΑΦΡΟΔΙΤΗ	5/8/1992	ΔΙΟΙΚΗΤΗΡΙΟΥ 2	ΚΑΣΤΟΡΙΑ	2467050502		
15077702281	ΚΑΛΥΒΑ	ΣΤΥΛΙΑΝΗ	15/7/1977	ΠΑΠΑΡΕΣΚΑ 8	ΚΑΣΤΟΡΙΑ		6947080907	
20057502098	ΠΟΥΛΙΑΔΗ	ΑΝΔΡΟΝΙΚΗ	20/5/1975	ΓΟΥΝΑΡΑΔΩΝ 8, ΧΛΟΙ	ΚΑΣΤΟΡΙΑ	2467401221		
16038902271	ΠΑΠΑΜΟΣΧΟΥ	ΕΥΑΓΓΕΛΙΑ	16/3/1989	ΑΘ. ΧΡΗΣΤΟΠΟΥΛΟΥ 9	ΚΑΣΤΟΡΙΑ		6937123987	
25089702041	ΑΠΟΣΤΟΛΟΥ	ΜΑΡΙΝΑ	25/8/1997	ΒΑΛΛΑ 3	ΚΑΣΤΟΡΙΑ		6944200301	

Α/Α	ΗΜΕΡΟΜΗΝΙΑ	ΩΡΑ	ΑΙΤΙΟΛΟΓΙΑ	ΣΗΜΕΙΩΣΕΙΣ
47	29/4/2022	12:00	ok	
26	28/4/2022	12:00	test	
27	19/3/2021	13:00	Εξέταση	
27	16/3/2021	12:00	Εξέταση	

Εικόνα 57: Η φόρμα Επισκέψεις (Ραντεβού) Ασθενών


Με το άνοιγμα της φόρμας, προβάλλονται τα δεδομένα όλων των ασθενών με τις επισκέψεις τους στο ιατρείο. Για την καταχώρηση ενός νέου ραντεβού, απαιτείται ο χρήστης να έχει επιλέξει από την λίστα των ασθενών την εγγραφή που επιθυμεί.

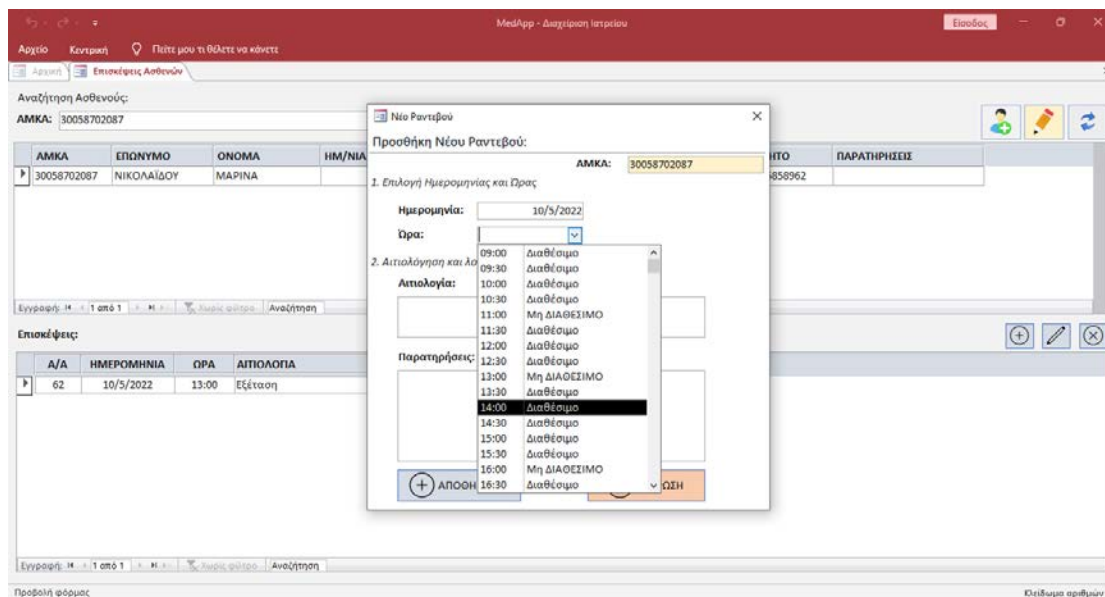
Για να επιλέξει μια εγγραφή μπορεί να το κάνει απευθείας από την λίστα μετακινούμενος στις εγγραφές είτε να τον αναζητήσει μέσω ΑΜΚΑ, πληκτρολογώντας το στο πεδίο «**ΑΜΚΑ**». Εάν δεν υπάρχει καταχωρημένο ΑΜΚΑ στην ΒΔ τότε εμφανίζεται το παρακάτω μήνυμα:



Εικόνα 58: Μήνυμα προειδοποίησης πως δεν βρέθηκε ασθενής με το ΑΜΚΑ που δόθηκε στο πεδίο.

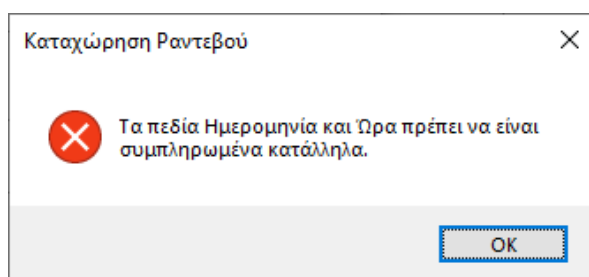
Εφόσον βρεθεί η εγγραφή του ασθενή, εμφανίζονται τα ραντεβού του στην λίστα «**Επισκέψεις**». Η λίστα «**Επισκέψεις**» εμφανίζει τα ραντεβού του κάθε ασθενή, κάθε φορά που ο

χρήστης επιλέγει διαφορετικό ασθενή. Για την καταχώρηση ενός νέου ραντεβού για έναν ασθενή, ο χρήστης πατά το κουμπί «**Νέα Επίσκεψη**»  και εμφανίζεται η παρακάτω φόρμα:



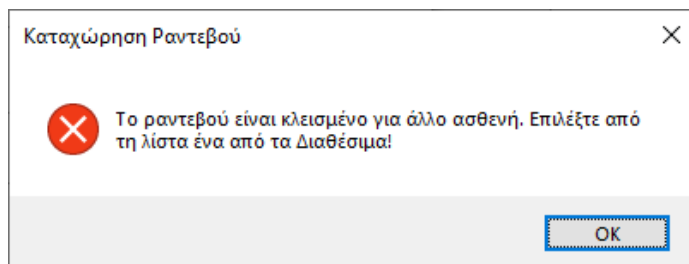
Εικόνα 59: Η φόρμα καταχώρησης ενός νέου ραντεβού ασθενή

Το πεδίο ΑΜΚΑ της φόρμας είναι συμπληρωμένο αυτόματα και δεν μπορεί να αλλάξει τιμή. Ο χρήστης συμπληρώνει τα πεδία «**Ημερομηνία**» και «**Ωρα**», τα οποία είναι υποχρεωτικά για την καταχώρηση του ραντεβού. Σε περίπτωση που δεν συμπληρωθούν και πατηθεί το κουμπί «**ΑΠΟΘΗΚΕΥΣΗ**» εμφανίζεται ανάλογο μήνυμα.



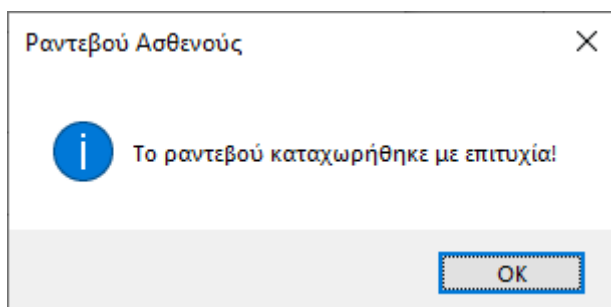
Εικόνα 60: Μήνυμα σφάλματος για κενές τιμές στα πεδία Ημερομηνία και Ωρα

Στο πεδίο «**Ωρα**» εμφανίζονται οι ώρες του ιατρού που μπορεί να δεχθεί ο γιατρός μια επίσκεψη και η ένδειξη διαθεσιμότητας της ώρας. Εάν ο χρήστης επιλέξει μια μη διαθέσιμη ώρα εμφανίζεται σχετικό μήνυμα προειδοποίησης:



Εικόνα 61: Μήνυμα σφάλματος για δεσμευμένο ραντεβού


Εφόσον έχει γίνει σωστά η καταχώρηση ο χρήστης πατάει το κουμπί «**ΑΠΟΘΗΚΕΥΣΗ**» και το ραντεβού καταχωρείται στη ΒΔ επιτυχώς ενώ παράλληλα ανανεώνονται και η λίστα των εγγραφών του ραντεβού του ασθενή στην φόρμα «Επισκέψεις Ασθενών».

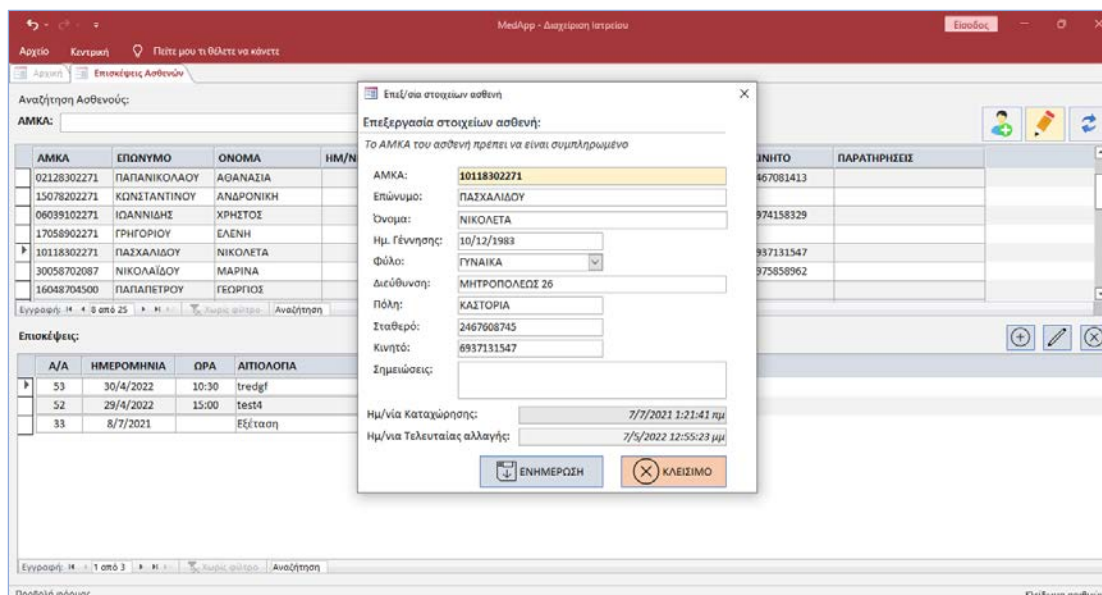


Εικόνα 62: Μήνυμα ενημέρωσης για επιτυχή καταχώρηση του ραντεβού ασθενή

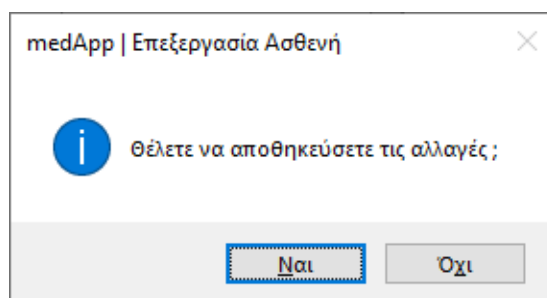
5.4 Αλλαγή στοιχείων ασθενή - επίσκεψης

Η αλλαγή των στοιχείων του ασθενή ή της επίσκεψης του αφορούν την διαδικασία της ενημέρωσης των εγγραφών τους στην ΒΔ. Οι φόρμες που χρησιμοποιούνται για αυτές τις λειτουργίες είναι ίδιες σε επίπεδο στοιχείων ελέγχου και έλεγχου εγκυρότητας των δεδομένων στα απαραίτητα πεδία. Ασφαλώς, οι δυο διαδικασίες είναι ανεξάρτητες μεταξύ τους αφού αφορούν ενημέρωση δεδομένων σε ξεχωριστούς πίνακες.

Για να αλλάξει ο χρήστης τα **στοιχεία** ενός **ασθενή**, αρχικά τον αναζητεί στη λίστα, όπως περιγράψαμε στην προηγούμενη παράγραφο και έπειτα τον επιλέγει. Στη συνέχεια πατάει το κουμπί «**Επεξ/σια Ασθενή**»  και εμφανίζεται η παρακάτω φόρμα.

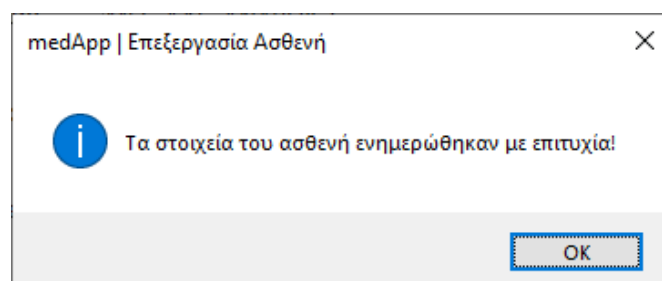


Εικόνα 63: Η φόρμα «Επεξ/σια στοιχείων ασθενή» όπως εμφανίζεται μετά το πάτημα του κουμπιού Ο χρήστης εφ’ όσον ενημερώσει τα πεδία των τιμών που επιθυμεί, πατάει το κουμπί «**Ε-ΝΗΜΕΡΩΣΗ**» και εμφανίζεται μήνυμα επιβεβαίωσης της ενέργειας



Εικόνα 64: Μήνυμα επιβεβαίωσης αλλαγών στον ασθενή

Εάν ο χρήστης πατήσει «Όχι» η φόρμα κλείνει και ο χρήστης επαναλαμβάνει την διαδικασία από την αρχή. Διαφορετικά, η ενημέρωση των στοιχείων είναι επιτυχής και εμφανίζεται το παρακάτω μήνυμα.



Εικόνα 65: Μήνυμα ενημέρωσης αλλαγών στον ασθενή

Με την ενημέρωση των στοιχείων των ασθενών, γίνεται αυτόματη ανανέωση και της λίστας των ασθενών στην προηγούμενη φόρμα. Η εφαρμογή προσφέρει την δυνατότητα στο χρή-

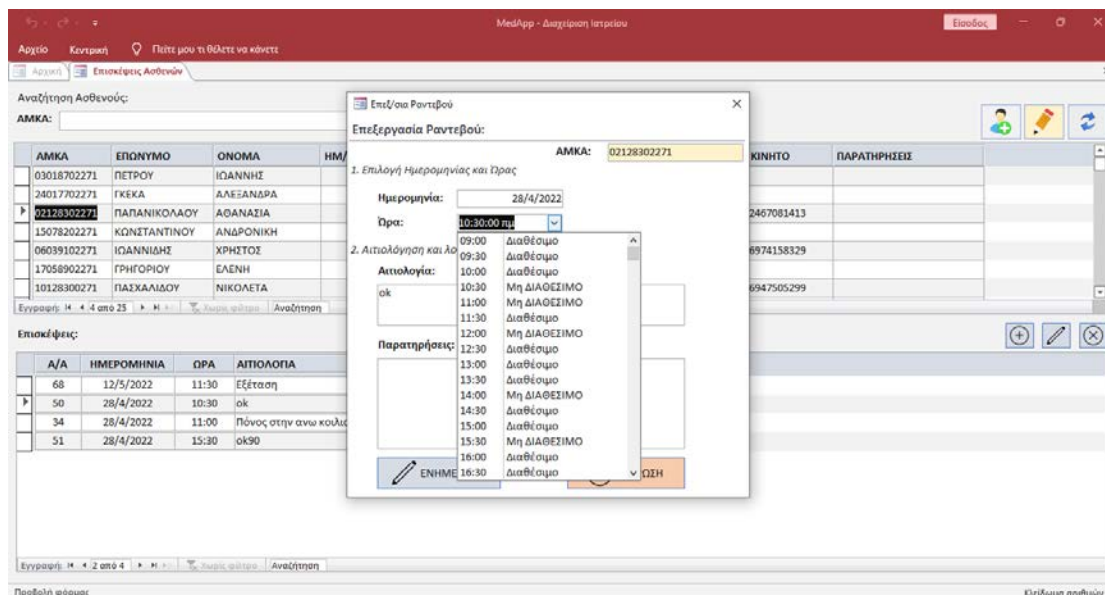
στη να κάνει ανανέωση των λιστών (ασθενείς και επισκέψεις) όποτε αυτός το επιθυμεί από

το κουμπί «Ανανέωση»



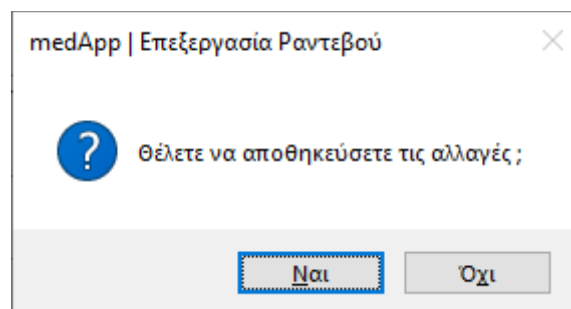
Με παρόμοια διαδικασία γίνεται και η **αλλαγή** των στοιχείων ενός ραντεβού. Για να αλλάξει ο χρήστης τα **στοιχεία** ενός **ραντεβού**, αρχικά επιλέγει το ραντεβού του ασθενή από την λίστα «Επισκέψεις», και πατώντας το κουμπί «Επεξ/σια Επίσκεψης» ανοίγει η επόμενη φόρμα.

Η φόρμα «Επεξ/σια Ραντεβού» περιέχει τα ίδια στοιχεία ελέγχου με την φόρμα «Νέο Ραντεβού» και όταν ανοίγει τα πεδία της έχουν τις τιμές της κατάλληλης εγγραφής της ΒΔ. Κατά την διάρκεια της ενημέρωσης των τιμών των πεδίων ισχύουν οι ίδιοι κανόνες με την διαδικασία της καταχώρησης ενός νέου ραντεβού.



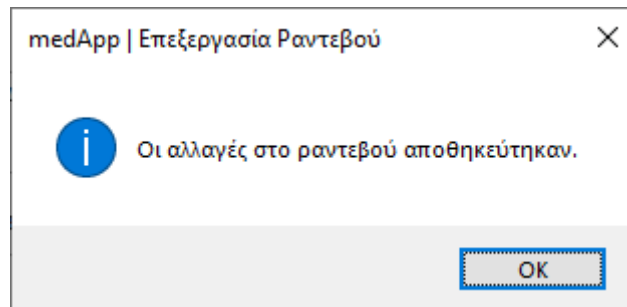
Εικόνα 66: Η φόρμα «Επεξ/σία Ραντεβού» όταν καλείται από τον χρήστη

Ο χρήστης εφ' όσον ενημερώσει τα πεδία των τιμών που επιθυμεί, πατάει το κουμπί «Ε-ΝΗΜΕΡΩΣΗ» και εμφανίζεται μήνυμα επιβεβαίωσης της ενέργειας.




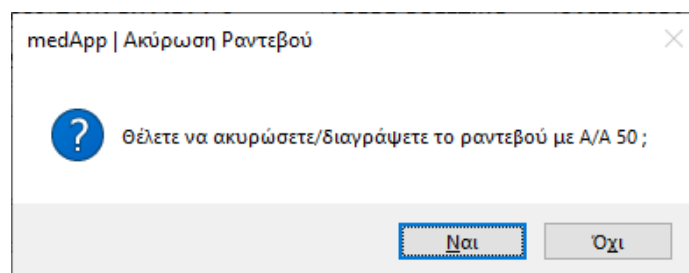
Εικόνα 67: Μήνυμα επιβεβαίωσης αλλαγών στο ραντεβού του ασθενή

Εάν ο χρήστης πατήσει «Όχι» η φόρμα κλείνει και ο χρήστης επαναλαμβάνει την διαδικασία από την αρχή. Διαφορετικά, η ενημέρωση των στοιχείων είναι επιτυχής και εμφανίζεται το παρακάτω μήνυμα.



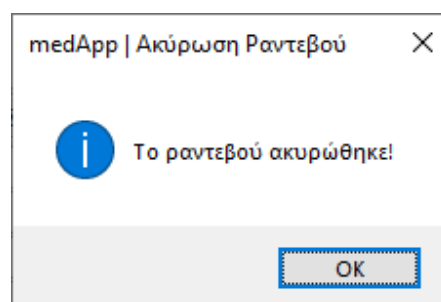
Εικόνα 68: Μήνυμα ενημέρωσης αλλαγών στο ραντεβού του ασθενή

Τέλος στην ίδια φόρμα ο χρήστης μπορεί να διαγράψει ένα ραντεβού από τη λίστα των επισκέψεων. Εφόσον το επιλέξει, πατάει το κουμπί «**Διαγραφή ραντεβού**»  και εμφανίζεται το παρακάτω μήνυμα επιβεβαίωσης της ενέργειας.



Εικόνα 69: Μήνυμα επιβεβαίωσης διαγραφής του ραντεβού ενός ασθενή

Εάν ο χρήστης πατήσει «Όχι» η ενέργεια ακυρώνεται. Διαφορετικά, το ραντεβού διαγράφεται και εμφανίζεται σχετικό μήνυμα.



Εικόνα 70: Μήνυμα ενημέρωσης για τη διαγραφή του ραντεβού ενός ασθενή

5.4 Εκτυπώσεις εκθέσεων

Η εφαρμογή προσφέρει στο χρήστη την δυνατότητα εκτύπωσης χρήσιμων αναφορών που σχετίζονται με τους ασθενείς και τα ραντεβού που κλείνουν στο ιατρείο. Την πληροφορία που παρέχει η κάθε μια από αυτές την περιγράψαμε στο προηγούμενο κεφάλαιο. Ο χρήστης έχει πρόσβαση σε αυτές μέσω της καρτέλας «**Εκτυπώσεις**» από την αρχική φόρμα. Στον χρήστη είναι διαθέσιμες οι εκθέσεις:

- α. Ημερήσιες επισκέψεις
- β. Ασθενείς ανά φύλο
- γ. Ηλικιακές ομάδες

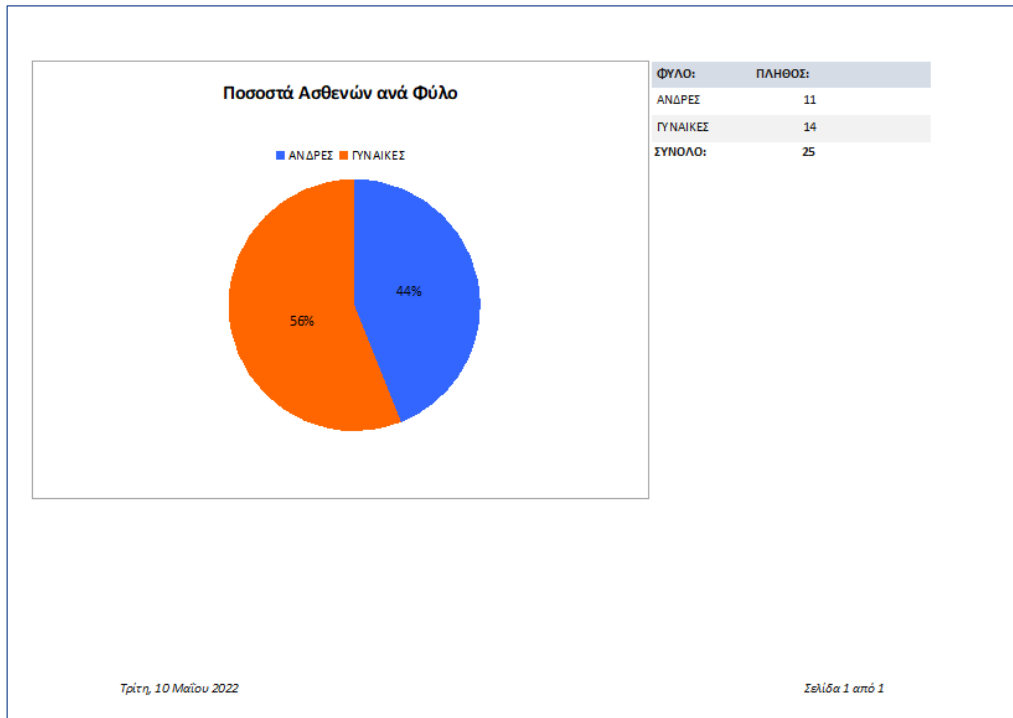
Η έκθεση «**Ημερήσιες επισκέψεις**» εμφανίζεται με το πάτημα του αντίστοιχου κουμπιού και προβάλλεται ως εξής:

Σημερινά Ραντεβού			
Ημερομηνία: 10/5/2022			
ΩΡΑ:	ΑΜΚΑ:	ΑΣΘΕΝΗΣ:	ΑΙΤΙΟΛΟΓΙΑ:
11:00	25089702041	ΑΠΟΣΤΟΛΟΥ ΜΑΡΙΝΑ	Εξέταση
13:00	30038702087	ΝΙΚΟΛΑΪΔΟΥ ΜΑΡΙΝΑ	Εξέταση
16:00	17038902271	ΓΡΗΓΟΡΙΟΥ ΕΛΕΝΗ	Εξέταση
19:00	02129304631	ΤΣΑΚΙΡΗΣ ΕΥΑΓΓΕΛΟΣ	Εξέταση

Τρίτη, 10 Μαΐου 2022 Σελίδα 1 από 1

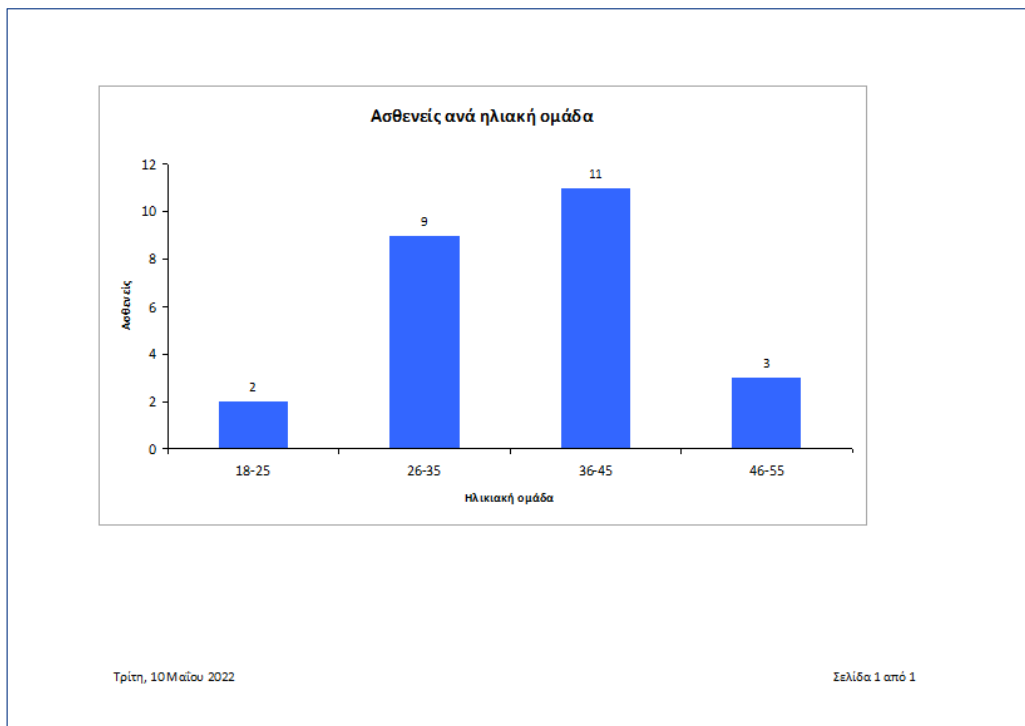
Εικόνα 71: Η έκθεση «Ημερήσιες επισκέψεις»

Η έκθεση «**Ποσοστά Ασθενών ανά Φύλο**» εμφανίζεται με το πάτημα του αντίστοιχου κουμπιού και προβάλλεται ως εξής:



Εικόνα 72: Η έκθεση «Ποσοστά Ασθενών ανά Φύλο»

Η έκθεση «**Ηλικιακές ομάδες**» εμφανίζεται με το πάτημα του αντίστοιχου κουμπιού και προβάλλεται ως εξής:



Εικόνα 73: Η έκθεση «Ηλικιακές ομάδες»

5.5 Πρόσθετες Λειτουργίες

Στις πρόσθετες λειτουργίες της εφαρμογής περιλαμβάνονται:

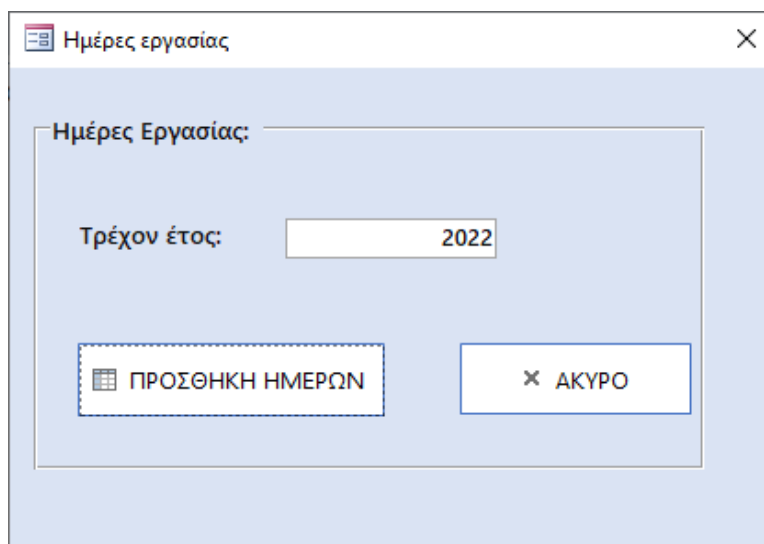
- α. Προσθήκη ημερών εργασίας**, με την οποία προστίθενται οι ημέρες ενός έτους στον πίνακα «AppDays» της βάσης. Η ημέρα εργασίας του ιατρού αποτελεί μοναδική εγγραφή καθώς σε μια μέρα μπορούν να γίνουν πολλές επισκέψεις από πολλούς πελάτες και με αυτό τον τρόπο κρατιέται και μια ιστορικότητα εγγραφών στη βάση.
- β. Διαχείριση χρηστών**, με την οποία ένας εξουσιοδοτημένος χρήστης (administrator) μπορεί να δημιουργεί νέους χρήστες στην βάση. Προς το παρόν είναι διαθέσιμη μόνο η λειτουργία της προσθήκης νέου χρήστη ενώ οποιεσδήποτε αλλαγές που αφορούν τον χρήστη (τύπος, κωδικός πρόσβασης, κτλ) γίνονται απευθείας μέσα από τη ΒΔ.

Στις επόμενες παραγράφους θα παρουσιάσουμε σύντομα αυτές τις πρόσθετες λειτουργίες οι οποίες είναι διαθέσιμες για τον διαχειριστή της ΒΔ από την καρτέλα «**Ρυθμίσεις**» της αρχικής φόρμας.

5.5.1 Προσθήκη Ημερών Εργασίας

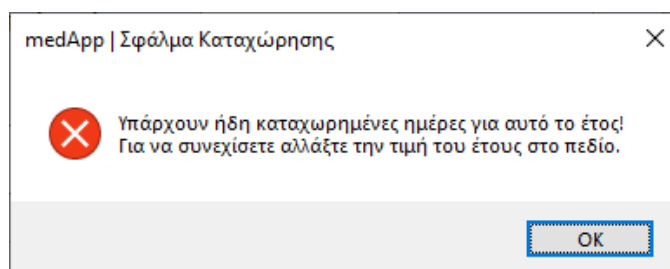
Η λειτουργία αυτή πρέπει να εκτελείται από το χρήστη κατά το τέλος του ημερολογιακού έτους, προκειμένου να καταχωρήσει στη ΒΔ τις ημέρες που λειτουργεί το ιατρείο. Θεωρούμε ότι πρέπει να καταχωρηθούν όλες οι ημέρες του έτους (1/1 – 31/12) για την πιο βέλτιστη διαχείριση των ραντεβού από τον χρήστη.

Η λειτουργία «Προσθήκη Ημερών εργασίας» ενεργοποιείται με το πάτημα του κουμπί «**Ημέρες Εργασίας / Ωρες**» στην καρτέλα «**Ρυθμίσεις**» και εμφανίζεται η παρακάτω φόρμα.



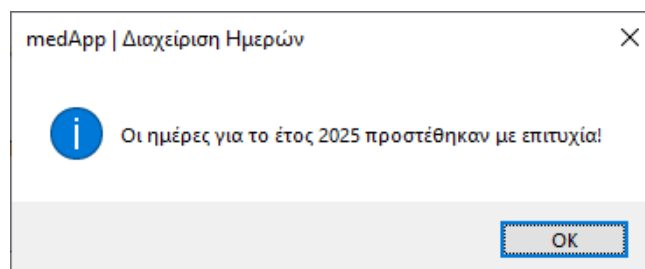
Εικόνα 74: Η φόρμα «Hμέρες εργασίας»

Η φόρμα περιέχει το πεδίο «**Τρέχον έτος**» το οποίο έχει ως προεπιλεγμένη τιμή την τρέχουσα χρονιά. Η τιμή μπορεί να αλλάξει από τον χρήστη ανάλογα το έτος που επιθυμεί. Πατώντας το κουμπί «**Προσθήκη ημερών**», γίνεται έλεγχος για το εάν έχουν καταχωρηθεί ήδη οι ημέρες του έτους στη ΒΔ, που αναγράφεται στο πεδίο. Εάν υπάρχουν τότε εμφανίζεται το παρακάτω μήνυμα:



Εικόνα 75: Μήνυμα προειδοποίησης για ήδη καταχωρημένες ημέρες του έτους που αναγράφεται στο πεδίο «Τρέχον έτος»

Σε αυτή την περίπτωση ο χρήστης επιστρέφει στη φόρμα, διορθώνει την τιμή με το έτος που θέλει και προχωράει στην καταχώρηση. Με το κουμπί «**ΑΚΥΡΟ**» η διαδικασία τερματίζεται. Εφόσον η καταχώρηση είναι επιτυχής, εμφανίζεται το παρακάτω μήνυμα.

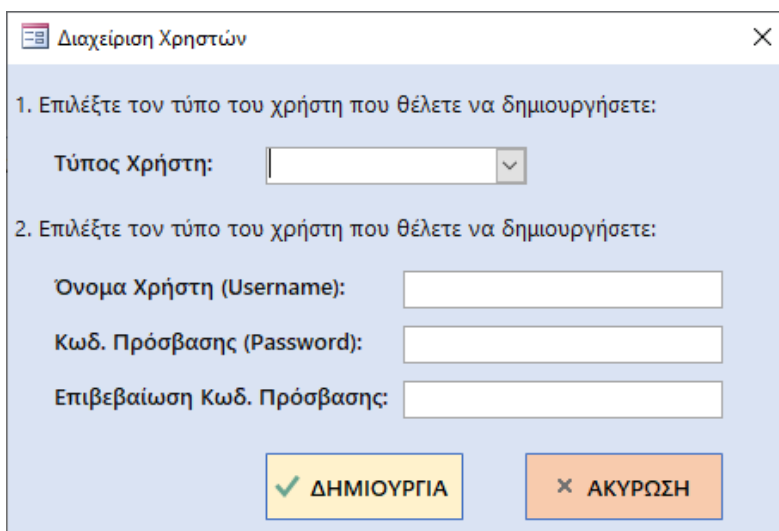


Εικόνα 76: Μήνυμα ενημέρωσης για επιτυχή καταχώρηση των ημερών στη ΒΔ

Προς το παρόν, η λειτουργία προσφέρει μόνο την δημιουργία ημερών ενός έτους. Σε μελλοντική επέκταση της εφαρμογής μπορεί να προστεθεί και η δυνατότητα προσθήκης των ωρών λειτουργίας του ιατρού που ο ασθενής μπορεί να κλείσει ένα ραντεβού. Η διαδικασία αυτή γίνεται μόνο μέσα από την ΒΔ (Πίνακας «AppHours»).

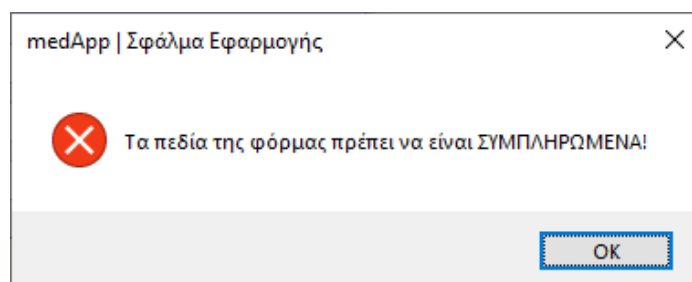
5.5.2 Διαχείριση Χρηστών

Η λειτουργία αυτή είναι χρήσιμη για τον διαχειριστή της εφαρμογής καθώς μπορεί να προσθέσει/δημιουργήσει χρήστες για την εφαρμογή χωρίς να εμπλακεί μέσα από την ΒΔ (π.χ με κώδικα SQL). Προς το παρόν, η λειτουργία αφορά μόνο την δημιουργία χρήστη για λόγους πρακτικότητας, ευκολίας και ασφάλειας. Πατώντας το κουμπί «Διαχείριση Χρηστών» ανοίγει η παρακάτω φόρμα.



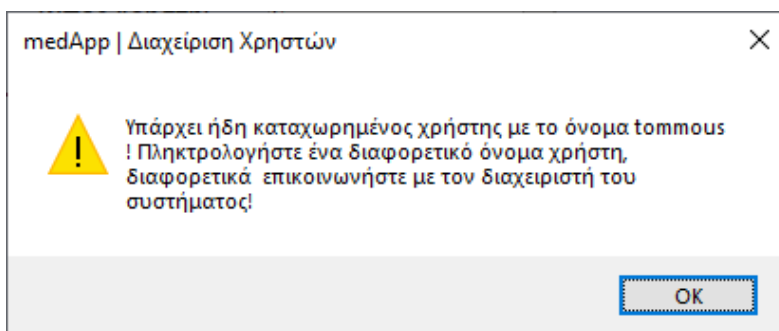
Εικόνα 77: Η φόρμα «Διαχείριση Χρηστών»

Όλα τα πεδία της φόρμας είναι υποχρεωτικά. Εάν ο χρήστης κατά λάθος πατήσει το κουμπί «ΔΗΜΙΟΥΡΓΙΑ» τότε εμφανίζεται το παρακάτω μήνυμα σφάλματος.



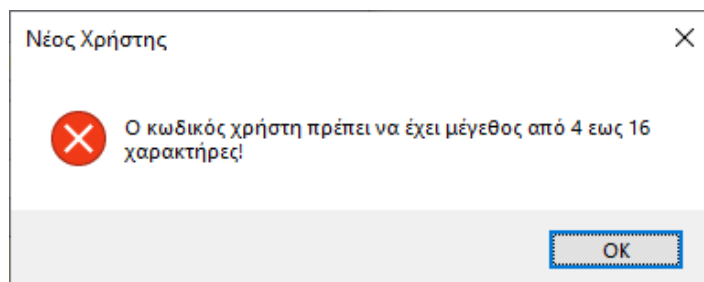
Εικόνα 78: Μήνυμα σφάλματος για μη συμπληρωμένα πεδία της φόρμας

Στην φόρμα υπάρχουν περιορισμοί ασφαλείας για την ορθή καταχώρηση στοιχείων στη ΒΔ. Έτσι με την σειρά εμφάνισης, οι τύποι των χρηστών που μπορεί να επιλέξει ο χρήστης από το πεδίο «**Τύπος Χρήστη**» είναι «Administrator», «Superuser» και «User». Έπειτα στο πεδίο «**Όνομα Χρήστη (Username)**» πληκτρολογεί το όνομα του χρήστη. Σε περίπτωση που υπάρχει ήδη το όνομα χρήστη στη ΒΔ εμφανίζεται το παρακάτω μήνυμα προειδοποίησης (περιορισμός διπλότυπης εγγραφής στον πίνακα «User»)



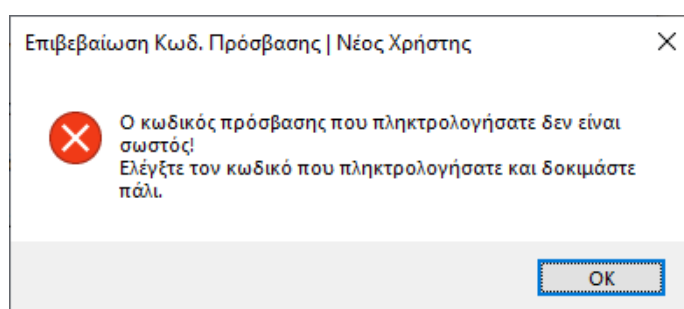
Εικόνα 79: Μήνυμα προειδοποίησης για ύπαρξη ονόματος χρήστη στη ΒΔ

Ο χρήστης αλλάζει το όνομα χρήστη και προχωράει στην καταχώρηση του κωδικού πρόσβασης στο πεδίο «**Κωδ. Πρόσβασης (Password)**». Έχουμε ορίσει ότι ο κωδικός πρόσβασης του χρήστη πρέπει να αποτελείται από 4 έως και 16 χαρακτήρες (αριθμοί, σύμβολα, γράμματα). Σε περίπτωση που ο χρήστης πληκτρολογήσει λιγότερους ή περισσότερους χαρακτήρες στο πεδίο τότε εμφανίζεται το παρακάτω μήνυμα.



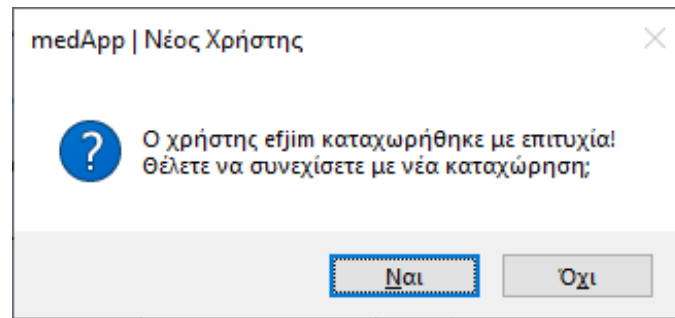
Εικόνα 80: Μήνυμα σφάλματος για το μέγεθος του κωδικού πρόσβασης του χρήστη

Εφόσον έχει πληκτρολογήσει έναν κωδικό πρόσβασης, ο χρήστης πρέπει να τον πληκτρολογήσει ξανά στο πεδίο «**Επιβεβαίωση Κωδ. Πρόσβασης**». Εάν υπάρχει κάποιο λάθος στην πληκτρολόγηση εμφανίζεται το παρακάτω μήνυμα.



Εικόνα 81: Μήνυμα ελέγχου λάθους κωδικού πρόσβασης κατά την επιβεβαίωση της τιμής του στο αντίστοιχο πεδίο

Διαφορετικά, πατώντας το κουμπί «**ΔΗΜΙΟΥΡΓΙΑ**» και εμφανίζεται το μήνυμα επιβεβαίωσης καταχώρησης της εγγραφής στη ΒΔ και εάν ο χρήστης επιθυμεί να καταχωρήσει και άλλον χρήστη.



Εικόνα 82: Μήνυμα ενημέρωσης για την επιτυχή καταχώρηση του χρήστη στη ΒΔ

Εάν επιλέξει «**Ναι**» επαναλαμβάνεται η παραπάνω διαδικασία διαφορετικά κλείνει και τερματίζει την λειτουργία. Εάν ο χρήστης θέλει να διακόψει οριστικά την λειτουργία της καταχώρησης νέου χρήστη πατάει το κουμπί «**ΑΚΥΡΩΣΗ**».

6

Συμπεράσματα Επεκτάσεις

Κεφάλαιο 6: Συμπεράσματα – Επεκτάσεις

Είναι γεγονός ότι η ανάπτυξη εφαρμογών που διαχειρίζονται δεδομένα, μικρού ή μεγάλου όγκου αποτελεί τον κυριότερο κορμό της λειτουργίας μιας επιχείρησης ή ενός οργανισμού. Ειδικότερα, στην εποχή των μεγάλων δεδομένων («big data») μια επιχείρηση φροντίζει να επενδύει σε προϊόντα ή υπηρεσίες Πληροφορικής προκειμένου να **αυξήσουν** την **παραγωγικότητα** τους μέσω ευέλικτων και εύχρηστων διαδικασιών. Αυτό άλλωστε ήταν και το αντικείμενο αυτής της πτυχιακής εργασίας, που εξετάσαμε την περίπτωση ενός ιατρείου που χαρακτηρίζεται από έναν μεγάλο αριθμό ασθενών-επισκεπτών που κλείνουν ραντεβού για εξέταση στο ιατρείο περισσότερες από μια φορές.

Γι' αυτό το λόγο επιλέξαμε την υλοποίηση αυτής της εφαρμογής στο περιβάλλον της **MS Access**. όπως διαπιστώθηκε η MS Access είναι ένα γνώριμο περιβάλλον εργασίας και αποτελεί το **κατάλληλο εργαλείο** για το **σχεδιασμό** και την **ανάπτυξη εφαρμογών κατά παραγγελία**, που ανταποκρίνονται πλήρως στις ανάγκες μιας επιχείρησης και μπορούν να προσαρμόζονται εύκολα στον τρόπο λειτουργίας της επιχείρησης, αφού παρέχει μια μεγάλη γκάμα εργαλείων ανάπτυξης λειτουργικών εφαρμογών.

Όπως είδαμε τα εργαλεία αυτά, είναι πλήρως λειτουργικά με τη γλώσσα προγραμματισμού **Visual Basic** (για την ακρίβεια με την Visual Basic for Applications) μια από τις πιο ισχυρές γλώσσες προγραμματισμού, που αν και δεν είναι πλήρως αντικειμενοστραφής γλώσσα προγραμματισμού, ωστόσο μας επιτρέπει να **δημιουργούμε εύκολα** ενσωματωμένες **λειτουργίες** για την εφαρμογή.

Ασφαλώς, θα ήταν εξαιρετικά δύσκολη η υλοποίηση της εφαρμογής εάν η MS Access δεν υποστήριζε πλήρως το **σχεσιακό μοντέλο**. Αυτό μας βοήθησε να σχεδιάσουμε λεπτομέρεια τη βάση δεδομένων, τις σχέσεις και τους περιορισμούς των εγγραφών σύμφωνα με τις προδιαγραφές που μας δόθηκαν.

Ακόμα με την μέθοδο διαχωρισμού της εφαρμογής στα αρχεία δεδομένων και εφαρμογής, εξασφάλισαμε την **ορθή καταχώρηση** των **δεδομένων**, μειώσαμε τις πιθανότητες εμφάνισης υπολογιστικών και ανθρώπινων λαθών στη ΒΔ και προσφέραμε στον τελικό χρήστη, μέσω φιλικού γραφικού περιβάλλοντος, **άμεσες** και **σωστές** πληροφορίες που θα τον διευκολύνουν στην λήψη αποφάσεων (εκθέσεις, γραφήματα, κτλ.)

Ωστόσο, όπως διαπιστώθηκε, υπάρχουν περιπτώσεις εφαρμογών που η MS Access δεν αποτελεί την ιδανική λύση. Οι εφαρμογές αυτές απαιτούν μεγαλύτερη κλίμακα ασφαλείας, μεγαλύτερο αποθηκευτικό χώρο για τα δεδομένα της εφαρμογής, περισσότερα εργαλεία

σχεδίασης λειτουργιών και υποστήριξη σε μεγαλύτερο αριθμό χρηστών. Λαμβάνοντας υπ' όψη αυτούς τους παράγοντες η εφαρμογή μας μπορεί να **επεκταθεί** με την προσθήκη των παρακάτω λειτουργιών:

- **Κρυπτογράφηση της βάσης δεδομένων.** Τα ευαίσθητα δεδομένα των ασθενών (προσωπικές πληροφορίες, ασθένειες, θεραπείες, κτλ.) πρέπει να προστατεύονται από εξωγενείς παράγοντες. Η MS Access προσφέρει την δυνατότητα προστασίας ενός αρχείου βάσης δεδομένων με την προσθήκη ενός κωδικού πρόσβασης τόσο στο αρχείο του παρασκηνίου (δεδομένα/back-end) αλλά και στο αρχείο του προσκηνίου (εφαρμογή/front-end). Βέβαια η ύπαρξη εφαρμογών τρίτων κατασκευαστών μπορούν να παρακάμψουν την προσθήκη κωδικού πρόσβασης και να αλλοιωθούν έτσι τα δεδομένα ή ο κώδικας VBA της εφαρμογής. Γι' αυτό αποτελεί αναγκαιότητα η παροχή σύνθετων μηχανισμών ασφαλείας από ένα ΣΔΒΔ.
- **Διατήρηση πλήρους ιατρικού φακέλου ασθενή.** Η εφαρμογή μπορεί να επεκταθεί πλήρως στο κομμάτι της αποθήκευσης και παροχής πληροφορίας σχετικά με το ιατρικό ιστορικό ενός ασθενή. Ο χρήστης εκτός από την κράτηση ενός ραντεβού, θα μπορεί να εισάγει στη ΒΔ την πάθηση με την οποία διαγνώστηκε ο ασθενής και την ενδεικτική θεραπεία που προτείνει ο γιατρός ανάλογα με την περίπτωση. Οι παθήσεις και οι διαγνώσεις θα αποτελούν ξεχωριστούς πίνακες στη ΒΔ και θα συσχετίζονται κατάλληλα μεταξύ τους και έτσι θα παρέχεται πλήρη γνώση στον γιατρό σχετικά με την πορεία της υγείας του ασθενή.
- **Προβολή πληροφοριών ασθενών.** Συμπληρωματικά με την παραπάνω επέκταση, θα δίνεται η δυνατότητα στον γιατρό να προβάλλει κατάλληλες εκτυπώσεις (αναφορές, γραφήματα, κτλ.) που αφορούν τον ιατρικό φάκελο του ασθενή, δίνοντας έτσι με γραφικό και συγκεντρωτικό τρόπο χρήσιμες πληροφορίες για το ιστορικό υγείας του κάθε ασθενή. Επιπρόσθετα, ο χρήστης θα μπορεί να αποθηκεύει και αποτελέσματα εξετάσεων των ασθενών σε ηλεκτρονική μορφή.
- **Διάθεση της εφαρμογής για κινητές συσκευές (smartphones, tablets).** Ασφαλώς δεν θα μπορούσε να υπάρξει πλέον επιτραπέζια εφαρμογή χωρίς την διάθεση της για φορητές συσκευές όπως smartphones και tablets. Οι εφαρμογές αυτές έχουν κατακλύσει πλέον την αγορά και προσφέρουν ένα πλήρες περιβάλλον εφαρμογής στο χρήστη παράλληλο με αυτό της επιτραπέζιας έκδοσης. Σε αυτή την έκδοση, ο χρήστης θα μπορεί να διαχειρίζεται όλα τις παραπάνω λειτουργίες με ευκολία και ασφάλεια από οποιοδήποτε μέρος και εάν βρίσκεται.

Για την υλοποίηση των παραπάνω λειτουργιών η MS Access δεν αποτελεί την ιδανικότερη λύση, τόσο σε θέμα αποθηκευτικού χώρου αλλά και σε θέμα διεπαφής χρήστη και αναφορών (εκθέσεων). Σε αυτή την περίπτωση το καταλληλότερο περιβάλλον ανάπτυξης προγράμματος είναι το **Microsoft Visual Studio 2019** και η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί είναι η **C#** για μια επιτραπέζια χρήση της εφαρμογής ενώ για την επέκταση της σε διαδικτυακή εφαρμογή η πιο κατάλληλη γλώσσα προγραμματισμού είναι η **ASP.NET**.

Και οι δυο γλώσσες υποστηρίζονται πλήρως από το MS Visual Studio και από την Microsoft σε επίπεδο επεκτάσεων και εργαλείων. Τέλος, το αρχείο της βάσης δεδομένων μπορεί εύκολα να μεταφερθεί σε ένα ΣΔΒΔ που υποστηρίζει **MS SQL Server**, χωρίς να χαθούν τα δεδομένα των πινάκων και οι σχέσεις μεταξύ τους, προσφέροντας παράλληλα την δυνατότητα ανάπτυξης ενσωματωμένων διαδικασιών και συναρτήσεων, ανεξάρτητες από την κύρια εφαρμογή.

Ακόμα, η **Microsoft** προσφέρει μια νέα υπηρεσία, την «**Power Apps**» που παρέχει ένα ενιαίο περιβάλλον γρήγορης ανάπτυξης προσαρμοσμένων εφαρμογών που τα δεδομένα τους είτε είναι αποθηκευμένα στην νέα cloud-πλατφόρμα «**Microsoft Dataverse**» είτε σε διάφορες διαδικτυακές και εσωτερικές πηγές δεδομένων (SQL Server, SharePoint, Microsoft 365, Dynamics 365, κτλ.). Η υπηρεσία αυτή δεν είναι δωρεάν, έχει κερδίσει το ενδιαφέρον μεγάλων οργανισμών και επιχειρήσεων (IKEA, Coca-Cola, Shell, H&M, κ.α) και μπορεί να αναπτυχθεί και σε μικρότερους οργανισμούς ή επιχειρήσεις.

Κλείνοντας, είναι σημαντικό να αναφερθεί πως η ανάπτυξη εφαρμογών μικρής ή μεγάλης έκτασης (ανεξάρτητα της πλατφόρμας υλοποίησης) είναι και θα εξακολουθήσει να είναι στο μέλλον, ο κυριότερος άξονας ανάπτυξης μιας εταιρείας ή ενός οργανισμού. Στην εποχή της Πληροφορίας και του Ψηφιακού Μετασχηματισμού, αποτελεί μονόδρομο για μια επιχείρηση η ψηφιοποίηση των υπηρεσιών που παρέχει στους πελάτες, καθώς έτσι μειώνονται τα λειτουργικά έξοδα της ενώ παράλληλα βελτιώνεται η αποτελεσματικότητα των διαδικασιών της.

Βιβλιογραφία

Παράρτημα

Βιβλιογραφία

- [1] A. Couch, Microsoft Access 2010, VBA Programming Inside Out, Redmond, Washington: Microsoft Press, 2011.
- [2] M. R. Groh, Access 2010 Bible, New York: John Wiley & Sons Inc., 2010.
- [3] A. Silberschatz, H. F. Corth και S. Sudarshan, Συστήματα Βάσεων Δεδομένων, Αθήνα: Εκδόσεις ΓΚΙΟΥΡΔΑΣ Μ., 2021.
- [4] J. C. Stockman και A. Simpson, Access 2007 VBA Programming For Dummies, Indianapolis: Wiley Publishing, Inc., 2007.
- [5] Θ. Γεωργάκης και Κ. Κατσάμπαλος, Οδηγός Προγραμματισμού στο περιβάλλον της Visual Basic 6, Θεσσαλονίκη: Εκδόσεις ΑΠΘ - Τμήμα Αγρονόμων και Τοπογράφων Μηχανικών, 2005.
- [6] Δ. Γιάτας, Γ. Γώγουλος, Ι. Κοτίνη, Γ. Κυριακάκη, Δ. Μωράκης, Σ. Τζελέπη και Μ. Φραγκονικολάκης, Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές τους στο Διαδίκτυο - Β' ΕΠΑ.Λ, Αθήνα: Ι.Τ.Υ.Ε. ΔΙΟΦΑΝΤΟΣ, 2016.
- [7] Χ. Γουλιτίδης, Βάσεις Δεδομένων: Access 2010, Αθήνα: Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 2013.
- [8] Ε. Κεχρής, Σχεσιακές Βάσεις Δεδομένων, Αθήνα: Εκδόσεις ΚΡΙΤΙΚΗ, 2021.
- [9] Ι. Μανωλόπουλος και Α. Ν. Παπαδόπουλος, Βάσεις Δεδομένων: Θεωρία και Πρακτική Εφαρμογή, Αθήνα: Εκδόσεις ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, 2006.
- [10] Χ. Στασινός, Δουλεύοντας με την Access, Αθήνα: Εκδόσεις ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, 2007.
- [11] Α. Τσιμπήρης, Βάσεις Δεδομένων II (Διδακτικές Σημειώσεις), Σέρρες: ΑΤΕΙ ΣΕΡΡΩΝ/Τμήμα Πληροφορικής & Επικοινωνιών, 2005.
- [12] Γ. Χαριτούδη, Microsoft Visual Basic and Visual Basic for Application, Καλοχώρι, Θεσσαλονίκη: Εκδόσεις ΔΙΣΙΓΜΑ, 2017.
- [13] «Getting started with VBA in Office (Άρθρο),» Microsoft, 22 1 2022. [Ηλεκτρονικό]. Available: <https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office>. [Πρόσβαση 6 3 2022].
- [14] «Access VBA reference (Άρθρο),» Microsoft, 13 9 2021. [Ηλεκτρονικό]. Available: <https://docs.microsoft.com/en-us/office/vba/api/overview/access>. [Πρόσβαση 12 4 2022].
- [15] «DB-Engines Ranking,» DB-Engines, 11 4 2022. [Ηλεκτρονικό]. Available: <https://db-engines.com/en/ranking>. [Πρόσβαση 11 4 2022].
- [16] «Access SQL: basic concepts, vocabulary, and syntax (Online Οδηγός Βοήθειας),» Microsoft, [Ηλεκτρονικό]. Available: <https://support.microsoft.com/en->

- us/office/access-sql-basic-concepts-vocabulary-and-syntax-444d0303-cde1-424e-9a74-e8dc3e460671. [Πρόσβαση 29 3 2022].
- [17] «DevGuru Jet SQL Quick Reference guide,» DevGuru, 2018. [Ηλεκτρονικό]. Available: <https://www.devguru.com/content/technologies/jetsql/home.html>. [Πρόσβαση 9 3 2022].
- [18] «Why use Microsoft Access?,» Access World, 2013. [Ηλεκτρονικό]. Available: <https://www.access-programmers.co.uk/why-use-microsoft-access/>. [Πρόσβαση 6 12 2022].
- [19] «Microsoft Access Databases Are Still Popular (Blog),» Hallam ICS, 5 2021. [Ηλεκτρονικό]. Available: <https://www.hallam-ics.com/blog/microsoft-access-databases-are-still-popular>. [Πρόσβαση 9 11 2021].
- [20] «SQL,» Wikipedia, 2022. [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/SQL>. [Πρόσβαση 29 3 2022].
- [21] «Visual Basic for Applications,» Wikipedia, 2022. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Visual_Basic_for_Applications. [Πρόσβαση 30 3 2022].
- [22] «Microsoft Access,» Wikipedia, 2022. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Microsoft_Access. [Πρόσβαση 5 4 2022].
- [23] «Access Database Engine,» Wikipedia, 2022. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Access_Database_Engine. [Πρόσβαση 15 4 2022].
- [24] «Microsoft Access: Is it still relevant in 2022?,» CompariTech, 2022. [Ηλεκτρονικό]. Available: <https://www.comparitech.com/net-admin/microsoft-access/>. [Πρόσβαση 3 4 2022].

Παράρτημα

A. Κώδικας των φορμών

Φόρμα «frmAddAppointment»

```

' Κώδικας Φόρμας frmAddAppointment
Option Compare Database
' Διαδικασία εισαγωγής ραντεβού στη ΒΔ
Private Sub addAppointment(ByVal PID As Integer _
    , ByVal pdate As Variant _
    , ByVal phour As Variant _
    , ByVal psubject As String _
    , ByVal pnotes As Variant)

    Dim db As DAO.Database
    Dim rs As DAO.Recordset
    Dim sql, sql2 As String
    Dim l_appid As Integer

    On Error GoTo Err_Handler

    Set db = CurrentDb
    ' Εισαγωγή στον πίνακα Appointment.
    sql = "INSERT INTO Appointment"
    sql = sql + "(Pid, AppDate, Subject, Notes)"
    sql = sql + " VALUES('" & PID & "' , '" & pdate & "' , '" &
    psubject & "' , '" & pnotes & "' )"

    CurrentDb.Execute sql, dbFailOnError

    ' Εύρεση του τελευταίου πρωτεύοντος κλειδιού
    ' από τον πίνακα Appointment, πεδίο Id
    Set rs = db.OpenRecordset("Appointment")
    rs.MoveLast
    l_appid = rs.Fields("Id")
    rs.Close

    ' Εισαγωγή στον πίνακα AppDetails, σχεσιακό πίνακας (detail
table)
    sql2 = "INSERT INTO AppDetails"
    sql2 = sql2 + "(AppId, AppHour) VALUES('" & l_appid & "' , '"
& phour & "' )"

    CurrentDb.Execute sql2, dbFailOnError

    Result = MsgBox("Το ραντεβού καταχωρήθηκε με επιτυχία!", _
        vbInformation + vbOKOnly, "Ραντεβού Ασθενούς")
    If Result = vbOK Then
        DoCmd.Close acForm, "frmAddAppointment"
    End If
    CurrentDb.Close

Exit_Here:

```

```

        Exit Sub
    Err_Handler:
        MsgBox Err.Number & " " & Error.Description, _
            vbExclamation, "medApp | Σφάλμα Εφαρμογής"
        Resume Exit_Here
End Sub

' Συμβάν κουμπιού btnAddAppointment
Private Sub btnAddAppointment_Click()
    On Error GoTo Err_Handler

    Dim tpid As Integer
    Dim tdate, thour, tsubject, tnotes, sql As String
    Dim strCriteria As String

    tpid = Me.txtPIId.Value
    tdate = Me.txtAppDate.Value
    thour = Me.cboAppHour.Value
    tsubject = Nz(Me.txtSubject.Value, "Εξέταση")
    tnotes = Me.txtNotes.Value

    strCriteria = " AppDate = #" & Format(tdate, "yyyy-mm-dd") & "#
And AppHour = #" & thour & "#

    ' Έλεγχος εάν τα πεδία ημερομηνίας και ώρας είναι συμπληρωμένα.
    If IsNull(Me.txtAppDate) Or IsNull(Me.cboAppHour) Then
        MsgBox "Τα πεδία Ημερομηνία και Ώρα πρέπει να είναι συμπληρω-
μένα κατάλληλα.", vbCritical + vbOKOnly, "Καταχώρηση Ραντεβού"
        GoTo Exit_Here
    ' Έλεγχος εάν υπάρχει ήδη καταχωρημένο ραντεβού στην ίδια ώρα και
μέρα.
    ElseIf Not IsNull(DLookup("AppDate", "qryAppointmentsHours",
strCriteria)) Then
        MsgBox "Το ραντεβού είναι κλεισμένο για άλλο ασθενή. Επιλέξτε
από τη λίστα ένα από τα Διαθέσιμα!", vbCritical + vbOKOnly, "Καταχώ-
ρηση Ραντεβού"
        GoTo Exit_Here
    Else
        ' Κλήση διαδικασίας addAppointment για την καταχώρηση ραντε-
βού
        Call addAppointment(tpid, tdate, thour, tsubject, tnotes)
    End If
    strCriteria = ""
    [Forms]![frmPatientAppts]![frmSubAppointment].Requery
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

```

' Κλείσιμο φόρμας
Private Sub btnCancel_Click()
    DoCmd.Close
End Sub

' Συμβάν στο άνοιγμα της φόρμας
Private Sub Form_Open(Cancel As Integer)
    ' Αρχικοποίηση τιμών
    Me.txtPIId.Value = Null
    Me.txtPIId.Value = p_id
    Me.txtAppDate.Value = DMin("AppDate", "AppDays", "AppDate >= #" &
Format(Now(), "yyyy-mm-dd") & "#")
    Me.cboAppHour.Requery
End Sub

' Συμβάν ενημέρωσης τιμών της ώρας
' στο combobox
Private Sub txtAppDate_AfterUpdate()
    Me.cboAppHour.Requery
End Sub

' Συμβάν ενημέρωσης τιμών της ώρας
' στο combobox
Private Sub txtAppDate_Change()
    Me.cboAppHour.Requery
End Sub

```

Φόρμα «frmAddPatient»

```

Option Compare Database
' Διαδικασία addPatient για την προσθήκη
' νεου ασθενή
Private Sub addPatient(ByVal pAMKA As Variant _
    , ByVal plname As Variant _
    , ByVal pfname As Variant _
    , ByVal pdob As Variant _
    , ByVal pGender As Variant _
    , ByVal pAddress As Variant _
    , ByVal pCity As Variant _
    , ByVal pPhoneA As Variant _
    , ByVal pPhoneB As Variant _
    , ByVal pnotes As Variant _
)
On Error GoTo Err_Handler

    Dim db As DAO.Database
    Set db = CurrentDb

    sql = "INSERT INTO Patient"
    sql = sql + " (AMKA, LName, FName, DOB, Gender, Address, City,
PhoneA, PhoneB, Notes)"

```

```

        sql = sql + " VALUES('" & pAMKA & "' , '" & plname & "' , '" &
pfname & "' , '" & pdob & "' , "
        sql = sql + " '" & pGender & "' , '" & pAddress & "' , '" & pCity
& "' , "
        sql = sql + " '" & pPhoneA & "' , '" & pPhoneB & "' , '" & pnotes
& "' )"

CurrentDb.Execute sql, dbFailOnError

CurrentDb.Close
Result = MsgBox("Τα στοιχεία του ασθενούς καταχωρήθηκαν με επιτυ-
χία!", _
                vbInformation + vbOKOnly, "Στοιχεία Ασθενούς")
If Result = vbOK Then
    DoCmd.Close acForm, "frmAddPatient"
End If
Exit_Here:
Exit Sub
Err_Handler:
'DoCmd.Hourglass False
MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub
' Συμβάν στο κουμπί για προσθήκη ασθενή
Private Sub btnAddPatient_Click()
On Error GoTo Err_Handler
Dim tAMKA, tLname, tFname, tDOB, tGender, _
    tAddress, tCity, tPhoneA, tPhoneB, tnotes As String

' Έλεγχος εάν το πεδίο AMKA έχει σωστές καταχωρημένες τιμές.
If mdlPatient.validateAMKA(Me.txtAMKA.Value) = False Then
    MsgBox "Το πεδίο AMKA πρέπει να είναι συμπληρωμένο.",
vbCritical + vbOKOnly, "Αριθμός AMKA"
    Me.txtAMKA.SetFocus
Exit Sub
Else
    tAMKA = Me.txtAMKA.Value
    tDOB = Me.txtDoB.Value
    l_amka = Me.txtAMKA.Value
End If

' Ανάθεση των τιμών των Controls της φόρμας στις παραμέτρους της
διαδικασίας addPatient
tLname = Nz(Me.txtLName.Value, "")
tFname = Nz(Me.txtFName.Value, "")
tGender = Nz(Me.cboGender.Value, "")
tAddress = Nz(Me.txtAddress.Value, "")
tCity = Nz(Me.txtCity.Value, "")
tPhoneA = Nz(Me.txtPhoneA.Value, "")
tPhoneB = Nz(Me.txtPhoneB.Value, "")
tnotes = Nz(Me.txtNotes.Value, "")
' Κλήση της διαδικασίας addPatient

```

```

    Call addPatient(tAMKA, tLname, tFname, tDOB, tGender, tAddress,
tCity, tPhoneA, tPhoneB, tnotes)
    If CurrentProject.AllForms("frmsubPatient").IsLoaded Then
        ' Ανανέωση λίστας ασθενών
        [Forms]![frmPatientAppts]![frmsubPatient].Requery
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    Select Case Err.Number
        Case 94
            MsgBox "Πεδίο AMKA: Κενή τιμή! " & vbCrLf & " Για να
συνεχίσετε συμπληρώστε κατάλληλα το πεδίο AMKA της φόρμας!", _
vbCritical + vbOKOnly, "medApp | Σφάλμα Καταχώρησης"
            Resume Next
        Case Else
            MsgBox Err.Number & " " & Err.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
            Resume Exit_Here
    End Select
End Sub

' Συμβάν κουμπιού Άκυρο/Κλείσιμο φόρμας
Private Sub btnCancel_Click()
    DoCmd.Close
End Sub

' Έλεγχος της τιμής του πεδίου AMKA
Private Sub txtAMKA_AfterUpdate()
On Error GoTo Err_Handler

    Dim l_amka As String
    Dim strMsg As String
    Dim dvalue As Variant

    strMsg = "Μη έγκυρος αριθμός AMKA. Ο αριθμός AMKA:" & vbCrLf
    strMsg = strMsg + " - Πρέπει να έχει 11 ψηφία." & vbCrLf & " -
Να αποτελείται μόνο από ψηφία (0 - 9)."

    If validateAMKA(Me.txtAMKA.Value) = False Then
        Me.txtAMKA.SetFocus
        MsgBox strMsg, vbCritical + vbOKOnly, "Σφάλμα καταχώρησης |
AMKA"
    Exit Sub
    Else
        l_amka = Me.txtAMKA.Value
        dvalue = DateSerial(CInt(Mid(l_amka, 5, 2)), CInt(Mid(l_amka,
3, 2)), CInt(Left(l_amka, 2)))
        Me.txtDoB.Value = dvalue
    End If

Exit_Here:

```

```

Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

Φόρμα «frmEditAppointment»

```

Option Compare Database
' Δηλώσεις δημόσιων μεταβλητών
Public db As Database
Public rst As Recordset
Public sql As String

' Συμβάν ακύρωσης ενημέρωσης εγγραφών
Private Sub btnCancel_Click()
On Error GoTo Err_Handler
    sql = ""
    If Not (rst Is Nothing) Then
        rst.Close
        db.Close
        Set rst = Nothing
        Set db = Nothing
    End If
    DoCmd.Close
Exit_Here:
    Set db = Nothing
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν κατά το πάτημα του κουμπιού Ενημέρωση
Private Sub btnUpdateAppointment_Click()
    Dim l_appdate, l_apphour, l_subject, l_notes, strCriteria As
String
On Error GoTo Err_Handler

    Response = MsgBox("Θέλετε να αποθηκεύσετε τις αλλαγές ;", _
        vbYesNo + vbQuestion, "medApp | Επεξεργασία
Ραντεβού")

    If Response = vbYes Then
        l_appdate = Me.txtAppDate.Value
        l_apphour = Me.cboAppHour.Value
        l_subject = Me.txtSubject.Value
        l_notes = Me.txtNotes.Value

```

```

    strCriteria = " AppDate = #" & Format(l_appdate, "yyyy-mm-
dd") & "# And AppHour = #" & l_apphour & "#"
    ' Έλεγχος εάν τα πεδία ημερομηνίας και ώρας είναι συμπληρωμέ-
να.
    If IsNull(Me.txtAppDate) Or IsNull(Me.cboAppHour) Then
        MsgBox "Τα πεδία Ημερομηνία και Ώρα πρέπει να είναι συ-
μπληρωμένα κατάλληλα.", _
            vbCritical + vbOKOnly, "Καταχώρηση Ραντεβού"
        GoTo Exit_Here
    ' Έλεγχος εάν υπάρχει ήδη καταχωρημένο ραντεβού στην ίδια ώρα
και μέρα.
    ElseIf Not IsNull(DLookup("AppDate", "qryAppointmentsHours",
strCriteria)) Then
        MsgBox "Το ραντεβού είναι κλεισμένο για άλλο ασθενή. Επι-
λέξτε από τη λίστα ένα από τα Διαθέσιμα!", _
            vbCritical + vbOKOnly, "Καταχώρηση Ραντεβού"
        GoTo Exit_Here
    Else
        sql = "UPDATE Appointment "
        sql = sql + "LEFT JOIN AppDetails ON Appointment.Id = Ap-
pDetails.AppId "
        sql = sql + "SET Appointment.AppDate = '" & l_appdate &
"'"
        sql = sql + ", Appointment.Subject = '" & l_subject & "'
"
        sql = sql + ", Appointment.Notes = '" & l_notes & "' "
        sql = sql + ", AppDetails.AppHour = '" & l_apphour & "' "
        sql = sql + "WHERE Appointment.Id = " & app_id

        CurrentDb.Execute sql, dbFailOnError
        db.Close
        [Forms]![frmPatientAppts]![frmSubAppointment].Requery
        MsgBox "Οι αλλαγές στο ραντεβού αποθηκεύτηκαν.", _
            vbInformation + vbOKOnly, "medApp | Επεξεργασία
Ραντεβού"
        GoTo Exit_Here
    End If
End If
Exit_Here:
DoCmd.Close
Exit Sub
Err_Handler:
DoCmd.Hourglass False
MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub
' Κατά τη φόρτωση δεδομένων της φόρμας, τα πεδία φορτώνουν
' με τις τιμές της εγγραφής που επέλεξε ο χρήστης
Private Sub Form_Load()
On Error GoTo Err_Handler
Set db = CurrentDb
sql = "SELECT * FROM qryAppointments WHERE Id = " & app_id

```



```

Set rst = CurrentDb.OpenRecordset(sql)
Set Me.Form.Recordset = rst
With rst
    Me.txtAppDate = .Fields("AppDate")
    Me.cboAppHour = .Fields("AppHour")
    Me.txtSubject = .Fields("Subject")
    Me.txtNotes = .Fields("Notes")
End With
Exit_Here:
Exit Sub
Err_Handler:
DoCmd.Hourglass False
MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
Resume Exit_Here
End Sub
' Κατά το άνοιγμα της φόρμας, η παράμετρος p_id
' παίρνει τον αριθμό του ασθενή και την ημερομηνία
' της επίσκεψης του ασθενή
Private Sub Form_Open(Cancel As Integer)
    Me.txtPId.Value = p_id
    Me.txtAppDate.Value = DMin("AppDate", "AppDays", "AppDate >= #" &
Format(Now(), "yyyy-mm-dd") & "#")
    Me.cboAppHour.Requery
End Sub

' Ενημέρωση των τιμών της ώρας των ραντεβού
Private Sub txtAppDate_AfterUpdate()
    Me.cboAppHour.Requery
End Sub

' Ενημέρωση των τιμών της ώρας των ραντεβού
' κατά την αλλαγή της ημερομηνίας
Private Sub txtAppDate_Change()
    Me.cboAppHour.Requery
End Sub

```

Φόρμα «frmEditPatient»

```

Option Compare Database
Public db As Database
Public rst As Recordset
Public sql As String
' Συμβάν με το πάτημα του κουμπιού ΑΚΥΡΩΣΗ
Private Sub btnCancel_Click()
On Error GoTo Err_Handler
    sql = ""
    If Not (rst Is Nothing) Then
        rst.Close
        db.Close
        Set rst = Nothing

```

```

        Set db = Nothing
    End If
    DoCmd.Close
Exit_Here:
    Set db = Nothing
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν με την φόρτωση δεδομένων στη φόρμα
' Όταν ανοίγει τα πεδία περιέχουν τις τιμές της
' εγγραφής του ασθενή
Private Sub Form_Load()
On Error GoTo Err_Handler
    Set db = CurrentDb
    sql = "SELECT * FROM Patient WHERE Id = " & p_id

    Set rst = CurrentDb.OpenRecordset(sql)
    Set Me.Form.Recordset = rst
    ' Αντιστοιχίση τιμών του recordset στα πεδία
    ' της φόρμας
    With rst
        Me.txtId = .Fields("Id")
        Me.txtAMKA = .Fields("AMKA")
        Me.txtLName = .Fields("LName")
        Me.txtFName = .Fields("FName")
        Me.txtDoB = .Fields("DOB")
        Me.cboGender = .Fields("Gender")
        Me.txtAddress = .Fields("Address")
        Me.txtCity = .Fields("City")
        Me.txtPhoneA = .Fields("PhoneA")
        Me.txtPhoneB = .Fields("PhoneB")
        Me.txtNotes = .Fields("Notes")
        Me.txtCrtDate = .Fields("CreationDate")
        Me.txtUpdDate = .Fields("UpdDate")
    End With
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    Select Case Err.Number
        Case 94, 3075
            MsgBox "Κενή τιμή AMKA: " & vbNewLine & "Πρέπει να επιλέ-
ξετε έναν ασθενή για να επεξεργαστείτε τα στοιχεία του!", _
                vbCritical + vbOKOnly, "medApp | Σφάλμα Επε-
ξεργασίας"
            Resume Next
        Case Else

```

```

        MsgBox Err.Number & " " & Err.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
        Resume Exit_Here
    End Select
End Sub
' Ενεργοποίηση του κουμπιού ΑΠΟΘΗΚΕΥΣΗ
' και ενημερωση της εγγραφής στον πίνακα Patient
Private Sub btnUpdatePatient_Click()
On Error GoTo Err_Handler
    Response = MsgBox("Θέλετε να αποθηκεύσετε τις αλλαγές ;", _
        vbYesNo + vbInformation, "medApp | Επεξεργασία
Ασθενή")
    If Response = vbYes Then
        With Me.Form.Recordset
            .Edit
            .Fields("AMKA") = Me.txtAMKA
            .Fields("LName") = Me.txtLName
            .Fields("FName") = Me.txtFName
            .Fields("DOB") = Me.txtDoB
            .Fields("Gender") = Me.cboGender
            .Fields("Address") = Me.txtAddress
            .Fields("City") = Me.txtCity
            .Fields("PhoneA") = Me.txtPhoneA
            .Fields("PhoneB") = Me.txtPhoneB
            .Fields("Notes") = Me.txtNotes
            .Fields("UpdDate") = Now()
            .Update
        End With

        Response = MsgBox("Τα στοιχεία του ασθενή ενημερώθηκαν με ε-
πιτυχία!", _
            vbOKOnly + vbInformation, "medApp | Επεξεργασία
Ασθενή")
        [Forms]![frmPatientAppts]![frmsubPatient].Requery
        If Response = vbOK Then
            DoCmd.Close acForm, "frmEditPatient"
        End If
        rst.Close
        Set rst = Nothing
        Set db = Nothing
    Else
        Cancel = True
        btnCancel_Click
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

```

' Έλεγχος εγκυρότητας δεδομένων στο πεδίο AMKA
Private Sub txtAMKA_AfterUpdate()
On Error GoTo Err_Handler
    Dim l_amka As String
    Dim strMsg As String
    Dim dvalue As Variant

    strMsg = "Μη έγκυρος αριθμός AMKA. Ο αριθμός AMKA:" & vbCrLf
    strMsg = strMsg + " - Πρέπει να έχει 11 ψηφία." & vbCrLf & " -
Να αποτελείται μόνο από ψηφία (0 - 9)."

    If validateAMKA(Me.txtAMKA.Value) = False Then
        Me.txtAMKA.SetFocus
        MsgBox strMsg, vbCritical + vbOKOnly, "Σφάλμα καταχώρησης |
AMKA"
    Exit Sub
    Else
        l_amka = Me.txtAMKA.Value
        dvalue = DateSerial(CInt(Mid(l_amka, 5, 2)), CInt(Mid(l_amka,
3, 2)), CInt(Left(l_amka, 2)))
        Me.txtDoB.Value = dvalue
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

Φόρμα «frmLogin»

```

Option Compare Database
' Κώδικας κουμπιού ΣΥΝΔΕΣΗ
' Συμβάν με το κλικ
Private Sub btnLogin_Click()
On Error GoTo Err_Handler
    ' Ελέγχουμε εάν έχει συμπληρωθεί το πεδίο Χρήστης (username)
    If IsNull(Me.txtUsername) Or Me.txtUsername = "" Then
        MsgBox "Συμπληρώστε το πεδίο ΧΡΗΣΤΗΣ!", vbOKOnly, "Στοιχεία
χρήστη - Χρήστης"
        Me.txtUsername.SetFocus
    Exit Sub
    ' Ελέγχουμε εάν έχει συμπληρωθεί το πεδίο Κωδ. Πρόσβασης (pass-
word)
    ElseIf IsNull(Me.txtPassword) Or Me.txtPassword = "" Then
        MsgBox "Συμπληρώστε το πεδίο ΚΩΔ. ΠΡΟΣΒΑΣΗΣ!", vbOKOnly,
"Στοιχεία χρήστη - Κωδ. Πρόσβασης"
        Me.txtPassword.SetFocus
    Exit Sub
    Else

```

```

        ' Ελέγχουμε εάν ο Κωδ. Πρόσβασης είναι σωστός. Εάν είναι σω-
        στός, ανοίγει η κύρια φόρμα
        ' και η φόρμα login κλείνει.
        If DLookup("Password", "Users", "[Id] = " & userID) =
Me.txtPassword.Value Then
            DoCmd.Close acForm, "frmLogin", acSaveNo
            DoCmd.OpenForm "frmMain"
        Else
            MsgBox "Λάθος Κωδικός Πρόσβασης. Προσπαθείστε ξανά!",
vbOKOnly + vbExclamation, "Σφάλμα Σύνδεσης Χρήστη"
            Me.txtPassword.SetFocus
        End If
    End If
Exit_Here:
    CurrentDb.Close
    Exit Sub
Err_Handler:
    'DoCmd.Hourglass False
    MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

Private Sub Form_Open(Cancel As Integer)
    ' Μόλις ανοίξει η φόρμα, ο κέρσορας
    ' βρίσκεται στο πεδίο Χρήστης
    Me.txtUsername.SetFocus
End Sub

Private Sub txtPassword_AfterUpdate()
    btnLogin_Click
End Sub

Private Sub btnExit_Click()
    ' Τερματισμός της εφαρμογής.
    Application.Quit
End Sub

' Συμβάν κατά την πληκτρολόγηση του ονόματος χρήστη
' για ύπαρξη στη ΒΔ
Private Sub txtUsername_AfterUpdate()
On Error GoTo Err_Handler
    ' Ελέγχουμε κατά την διάρκεια της πληκ/σης εάν το Username υπάρ-
    χει καταχωρημένο στη βάση
    ' μόλις αλλάξει πεδίο
    Dim criteria As String
    criteria = "[Username] = '" & Me.txtUsername.Value & "'"

    ' Σε περίπτωση που δεν βρεθεί ο χρήστης της εφαρμογής
    ' εμφανίζεται σχετικό μήνυμα λάθους, διαφορετικά
    ' εκχωρούμε στην public μεταβλητή userID την τιμή του πίνακα.
    If DCount("Id", "Users", criteria) = 0 Then

```

```

        MsgBox "Ο χρήστης " & Me.txtUsername.Value & " δεν υπάρχει!"
    -
        + " Επικοινωνήστε με τον διαχειριστή του συστήμα-
        τος!", vbExclamation + vbOKCancel, _
        "Έλεγχος Χρήστη"
        Me.txtUsername.SetFocus
    Exit Sub
Else
    userID = DLookup("Id", "Users", criteria)
    userType = DLookup("UserType", "Users", criteria)
    userName = DLookup("Username", "Users", criteria)
    Me.txtPassword.SetFocus
End If
Exit_Here:
    CurrentDb.Close
Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Error.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

Φόρμα «frmMain»

```

Option Compare Database
' Με τη φόρτωση των δεδομένων της φόρμας,
' γίνεται έλεγχος του τύπου χρήστη.
' 01: Πλήρης πρόσβαση στις λειτουργίες της εφαρμογής
' 02: Μερική πρόσβαση.
' 03: Περιορισμένη πρόσβαση
Private Sub Form_Load()
    ' Ανάλογα με τον τύπο χρήστη, ορισμένες λειτουργίες των φορμών
    ' δεν είναι διαθέσιμες (Visible/Enable False)
    If userType = "02" Then
        Me.btnUserMgr.Enabled = False
    ElseIf userType = "03" Then
        Me.btnPrintAgeGroups.Enabled = False
        Me.btnPrintPatientGender.Enabled = False
        Me.btnCreateCalendar.Enabled = False
        Me.btnUserMgr.Enabled = False
    End If
    ' Εμφάνιση ονόματος χρήστη στην οθόνη
    Me.lblUsername.Caption = userName
End Sub
' Συμβάν με το πάτημα του κουμπιού ΕΞΟΔΟΣ
Private Sub btnTerminate_Click()
    If (MsgBox("Η εφαρμογή θα τερματιστεί. Θέλετε να συνεχίσετε;", _
vbYesNo + vbExclamation, "medApp | Διαχείριση Ιατρεί-
ου") = vbYes) Then
        If Me.Dirty Then
            RunCommand acCmdSaveRecord

```

```

        End If
        Application.Quit
    Else
        DoCmd.CancelEvent
    End If
End Sub

' Άνοιγμα φόρμας Σύνδεση χρήστη από
' κουμπί ΑΛΛΑΓΗ ΧΡΗΣΤΗ
Private Sub btnOpenLogin_Click()
    DoCmd.Close
    DoCmd.OpenForm "frmLogin", acNormal, , , , acWindowNormal
End Sub

' Κουμπί ΑΝΑΝΕΩΣΗ ΔΕΔΟΜΕΝΩΝ
Private Sub btnRefreshAll_Click()
    Me.sfmDailyAppointments.Requery
    Me.sfmNextdayAppointments.Requery
End Sub

' Κουμπί ΕΚΤΥΠΩΣΗ ΣΗΜΕΡΙΝΑ ΡΑΝΤΕΒΟΥ
Private Sub btnPrintToday_Click()
    DoCmd.OpenReport "rptTodayAppts", acViewPreview, , , acWindowNormal
End Sub

' Κουμπί ΕΚΤΥΠΩΣΗ ΡΑΝΤΕΒΟΥ ΕΠΟΜΕΝΗΣ ΗΜΕΡΑΣ
Private Sub btnPrintNextday_Click()
    DoCmd.OpenReport "rptComingAppts", acViewPreview, , , acWindowNormal
End Sub

' Κουμπί ΝΕΟΣ ΑΣΘΕΝΗΣ
Private Sub btnOpenNewPatient_Click()
    DoCmd.OpenForm "frmAddPatient", acNormal, , , , acWindowNormal
End Sub

' Κουμπί ΡΑΝΤΕΒΟΥ ΑΣΘΕΝΩΝ
Private Sub btnOpenPatient_Click()
    DoCmd.OpenForm "frmPatientAppts", acNormal, , , , acWindowNormal
End Sub

' Κουμπί ΕΚΘΕΣΗΣ ΗΛΙΚΙΑΚΕΣ ΟΜΑΔΕΣ
Private Sub btnPrintAgeGroups_Click()
    DoCmd.OpenReport "rptPatientAgeGroups", acViewPreview, , , acWindowNormal
End Sub

' Κουμπί ΕΚΘΕΣΗΣ ΑΣΘΕΝΗΣ ΑΝΑ ΦΥΛΟ
Private Sub btnPrintPatientGender_Click()
    DoCmd.OpenReport "rptPatientperGender", acViewPreview, , , acWindowNormal
End Sub

```

```

' Κουμπί ΕΚΘΕΣΗΣ ΣΗΜΕΡΙΝΑ ΡΑΝΤΕΒΟΥ
Private Sub btnPrintTodayAppts_Click()
    DoCmd.OpenReport "rptTodayAppts", acViewPreview, , , acWindowNormal
End Sub

' Κουμπί ΗΜΕΡΕΣ ΕΡΓΑΣΙΑΣ
Private Sub btnCreateCalendar_Click()
    DoCmd.OpenForm "frmWorkdays", acNormal, , , , acWindowNormal
End Sub

' Κουμπί ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ
Private Sub btnUserMgr_Click()
    DoCmd.OpenForm "frmUserManager", acNormal, , , , acWindowNormal
End Sub

```

Φόρμα «frmPatientAppts»

```

Option Compare Database
Dim db As Database

' Αναζήτηση Ασθενή με βάση το ΑΜΚΑ του
Private Sub txtSearchAMKA_AfterUpdate()
On Error GoTo Err_Handler
    Dim criteria As String
    ' Εάν δεν δοθούν τιμές στο πεδίο αναζήτησης, εμφανίζονται
    ' όλες οι εγγραφές του πίνακα Patient
    If (Trim(Me.txtSearchAMKA.Value & "") = vbNullString) Then
        Forms!frmPatientAppts!frmSubPatient.Form.RecordSource = "SE-
LECT * FROM Patient ORDER BY Id"
    Else
        ' Διαφορετικά, εμφανίζεται ο ασθενής βάσει το ΑΜΚΑ του.
        criteria = " LIKE '*' & Me.txtSearchAMKA.Value & '*'"
        Forms!frmPatientAppts!frmSubPatient.Form.RecordSource = "SE-
LECT * FROM Patient WHERE AMKA " & criteria
    End If
    ' Σε περίπτωση που δεν βρεθεί ασθενής, εμφανίζεται κατάλληλο μή-
    νυμα
    If
Forms!frmPatientAppts!frmSubPatient.Form.RecordsetClone.RecordCount =
0 Then
        Cancel = True
        Response = MsgBox("Δεν βρέθηκαν εγγραφές με αυτό το ΑΜΚΑ.",
vbCritical + vbOKOnly, "Εύρεση ΑΜΚΑ")
        Forms!frmPatientAppts!frmSubPatient.Form.RecordSource = "SE-
LECT * FROM Patient ORDER BY Id"
        Me.txtSearchAMKA.SetFocus
    End If
Exit_Here:
    Exit Sub
Err_Handler:

```



```

        MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
        Resume Exit_Here
End Sub

' Ανοιγμα φόρμας Νέος Ασθενής
Private Sub btnOpenAddPatient_Click()
On Error GoTo Err_Handler
    DoCmd.OpenForm "frmAddPatient", acNormal, , , , acWindowNormal
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Ανοιγμα φόρμας Επεξ/σια Ασθενή
Private Sub btnOpenEditPatient_Click()
On Error GoTo Err_Handler
    DoCmd.OpenForm "frmEditPatient", acNormal, , , , acWindowNormal
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν που ενεργοποιείται με το κουμπί Νέο ραντεβού
Private Sub btnAddApp_Click()
On Error GoTo Err_Handler
    DoCmd.OpenForm "frmAddAppointment", acNormal, , , , acWindowNor-
mal
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Επεξεργασία Ραντεβού
' Συμβάν που ενεργοποιείται με το κουμπί Αλλαγή ραντεβού
Private Sub btnEditApp_Click()
On Error GoTo Err_Handler
    Dim criteria As String
    p_id = Me.txtp_Id
    criteria = "PIId = " & p_id
    ' Έλεγχος εάν έχει επιλεχθεί ένα ραντεβού από τη λίστα.

```

```

    If IsNull(DLookup("PId", "Appointment", criteria)) Then
        MsgBox "Δεν υπάρχει καταχωρημένο ραντεβού για αυτόν τον ασθενή, για επεξεργασία.", _
            vbCritical + vbOKOnly, "Σφάλμα | Ραντεβού ασθενή"
        Exit Sub
    Else
        ' Άνοιγμα της φόρμας Επεξεργασία Ραντεβού.
        DoCmd.OpenForm "frmEditAppointment", acNormal, , , acWindowNormal
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp | Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν που ενεργοποιείται με το κουμπί Διαγραφή/Ακύρωση ραντεβού
Private Sub btnCancelApp_Click()
On Error GoTo Err_Handler
    Dim sql, criteria As String

    p_id = Me.txtp_Id
    criteria = "PId = " & p_id
    ' Σε περίπτωση που πατηθεί το κουμπί 'Ακύρωση' χωρίς να υπάρχει κάποιον ραντεβού στον ασθενή.
    If IsNull(DLookup("PId", "Appointment", criteria)) Then
        MsgBox "Δεν υπάρχει καταχωρημένο ραντεβού για αυτόν τον ασθενή για ακύρωση!", _
            vbCritical + vbOKOnly, "Σφάλμα | Ραντεβού ασθενή"
        Exit Sub
    Else
        ' Διαφορετικά, εμφανίζεται κατάλληλο μήνυμα επιβεβαίωσης και ενεργοποιείται η διαγραφή του από τον πίνακα Appointments.
        Response = MsgBox("Θέλετε να ακυρώσετε/διαγράψετε το ραντεβού με A/A " & app_id & " ;", _
            vbYesNo + vbQuestion, "medApp | Ακύρωση Ραντεβού")
        If Response = vbYes Then
            Set db = CurrentDb

            sql = "DELETE * FROM Appointment WHERE Id = " & app_id
            CurrentDb.Execute sql, dbFailOnError
            DoCmd.SetWarnings True
            db.Close

            MsgBox "Το ραντεβού ακυρώθηκε!", vbInformation + vbOKOnly, "medApp | Ακύρωση Ραντεβού"
            [Forms]![frmPatientAppts]![frmSubAppointment].Requery
            GoTo Exit_Here
        End If
    End If

```

```

        End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν κουμπιού ανανέωσης δεδομένων στις φόρμες
Private Sub btnRefresh_Click()
    Forms!frmPatientAppts!frmsubPatient.Requery
    Forms!frmPatientAppts!frmSubAppointment.Requery
End Sub

' Συμβάν κουμπιού ΑΝΑΖΗΤΗΣΗ
Private Sub btnSearch_Click()
    txtSearchAMKA_AfterUpdate
End Sub

' Συμβάν κουμπιού ΕΜΦΑΝΙΣΗ ΟΛΩΝ
Private Sub btnShowAll_Click()
On Error GoTo Err_Handler
    Me.txtSearchAMKA.Value = ""
    txtSearchAMKA_AfterUpdate
Exit_Here:
    Exit Sub
Err_Handler:
    MsgBox Err.Number & " " & Err.Description, vbExclamation, "medApp
| Σφάλμα Εφαρμογής"
    Resume Exit_Here
End Sub

```

Φόρμα «frmSubAppointment»

```

Option Compare Database
' Κάθε φορά που αλλάζει ο ασθενής στη λίστα, αλλάζει
' και το id του ραντεβού στη λίστα των επισκέψεων
Private Sub Form_Current()
    app_id = Null
    app_id = Me.Id.Value
End Sub

```

Φόρμα «frmSubPatient»

```

Option Compare Database
' Κάθε φορά που αλλάζει ο ασθενής στη λίστα, αλλάζει
' και το id του ραντεβού στη λίστα των επισκέψεων
Private Sub Form_Current()
    p_id = Null

```

```

    p_id = Me.Id.Value
End Sub

```

Φόρμα «frmUserManager»

```

Option Compare Database
' Συμβάν μετά την πληκτρολόγηση του username
Private Sub txtUsername_AfterUpdate()
On Error GoTo Err_Handler
    ' Ελέγχουμε κατά την διάρκεια της πληκ/σης εάν το Username υπάρ-
χει
    ' καταχωρημένο στη βάση μόλις αλλάξει πεδίο.
    ' Εάν υπάρχει, ο χρήστης πρέπει να πληκτρολογήσει διαφορετικό
username.
    Dim criteria As String
    criteria = "[Username] = '" & Me.txtUsername.Value & "'"
    ' Έλεγχος εάν υπάρχει εγγραφή με το δοσμένο όνομα χρήστη
    If DCount("Id", "Users", criteria) <> 0 Then
        Response = MsgBox("Υπάρχει ήδη καταχωρημένος χρήστης με το
όνομα " & Me.txtUsername.Value & " !" _
            + " Πληκτρολογήστε ένα διαφορετικό όνομα χρήστη, διαφορε-
τικά " _
            + " επικοινωνήστε με τον διαχειριστή του συστήματος!",
vbExclamation + vbOKOnly, _
            "medApp | Διαχείριση Χρηστών")
        If Response = vbOKOnly Then
            Me.txtUsername.SetFocus
        End If
    Exit Sub
    Else
        Me.txtPassword.SetFocus
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
    Resume Exit_Here
End Sub

' Συμβάν μετά την πληκτρολόγηση του Password
Private Sub txtPassword_AfterUpdate()
On Error GoTo Err_Handler
    ' Ελέγχουμε μετά την πληκτρολόγηση εάν το μέγεθος του Password
    ' είναι μικρότερο από 4 χαρακτήρες και μεγαλύτερο από 16.
    ' Μέγιστος αριθμός χαρακτήρων password: 16 στοιχεία
    Dim pwd As String
    pwd = Me.txtPassword.Value

    If (Len(pwd)) < 4 Or (Len(pwd)) > 16 Then

```

```

        Response = MsgBox("Ο κωδικός χρήστη πρέπει να έχει μέγεθος
από 4 έως 16 χαρακτήρες!" _
        , vbCritical + vbOKOnly, "Νέος Χρήστης")
        If Response = vbOKOnly Then
            Me.txtPassword.SetFocus
        End If
    Exit Sub
End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
    Resume Exit_Here
End Sub

' Κουμπί ΔΗΜΙΟΥΡΓΙΑ
Private Sub btnCreateUser_Click()
On Error GoTo Err_Handler
    Dim usrtype, user, pwd As String
    user = Me.txtUsername.Value
    pwd = Me.txtPassword.Value
    usrtype = Me.cboUserType.Value
    ' Έλεγχος συμπλήρωσης απαραίτητων πεδίων
    If (Trim(usrtype & "") = vbNullString) Or (Trim(user & "") =
vbNullString) Or (Trim(pwd & "") = vbNullString) Then
        Response = MsgBox("Παρακαλώ συμπληρώστε κατάλληλα όλα τα πε-
δία!", vbCritical + vbOKOnly, "Σφάλμα Καταχώρησης")
        Exit Sub
    ElseIf Me.txtValidPwd.Value <> Me.txtPassword.Value Then
        Response = MsgBox("Ο κωδικός πρόσβασης που πληκτρολογήσατε
δεν είναι σωστός! " & vbCrLf & _
            "Ελέγξτε τον κωδικό που πληκτρολογήσατε και δοκιμά-
στε πάλι.", _
            vbCritical + vbOKOnly, "Επιβεβαίωση Κωδ. Πρόσβασης |
Νέος Χρήστης")
        Me.txtValidPwd.SetFocus
    Exit Sub
    Else
        ' Κλήση ενσωματωμένης διαδικασίας addUser
        Call addUser(user, pwd, usrtype)
    End If
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    Select Case Err.Number
        Case 94
            MsgBox "Τα πεδία της φόρμας πρέπει να είναι ΣΥΜΠΛΗΡΩΜΕ-
NA!", vbCritical + vbOKOnly, "medApp | Σφάλμα Εφαρμογής"
            Resume Next
        Case Else

```

```

        MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
        Resume Exit_Here
    End Select
End Sub

' Διαδικασία δημιουργίας νέου χρήστη στη ΒΔ
Private Sub addUser(ByVal luser As String, _
                    ByVal lpwd As String, _
                    ByVal ltype As String)
On Error GoTo Err_addUserError
    Dim db As Database
    Dim sql As String

    Set db = CurrentDb
    sql = "INSERT INTO Users"
    sql = sql + " (Username, Password, UserType)"
    sql = sql + " VALUES ('" & luser & "' , '" & lpwd & "' , '" &
ltype & "' )"

    CurrentDb.Execute sql, dbFailOnError
    Response = MsgBox("Ο χρήστης " & luser & " καταχωρήθηκε με επιτυ-
χία! " & vbCrLf & _
                    "Θέλετε να συνεχίσετε με νέα καταχώρηση;", _
                    vbQuestion + vbYesNo, "medApp | Νέος Χρήστης")
    If Response = vbYes Then
        Cancel = True
        clrControls
    Else
        db.Close
        DoCmd.Close
    End If
Exit_addUserError:
    Exit Sub
Err_addUserError:
    strMsg = "Error Number " & Err.Number & vbCrLf & Err.Description
    MsgBox strMsg, vbCritical, "Σφάλμα | Νέος Χρήστης"
    Resume Exit_addUserError
End Sub

' Διαδικασία διαγραφής/καθαρισμού των πεδίων
' της φόρμας από τις τιμές που πληκτρολογήθηκαν
Private Sub clrControls()
    Dim C As Control
    For Each C In Me.Controls
        If TypeOf C Is TextBox Then
            If C.ControlSource = "" Then
                C.Value = Null
            End If
        End If
    End For
    If TypeOf C Is ComboBox Then
        If C.ControlSource = "" Then
            C.Value = Null
        End If
    End If
End Sub

```

```

        End If
    End If
Next
End Sub

' Κουμπί ΑΚΥΡΩΣΗ
Private Sub btnCancel_Click()
    clrControls
    DoCmd.Close
End Sub

```

Φόρμα «frmWorkdays»

```

' Κουμπί ΠΡΟΣΘΗΚΗ ΗΜΕΡΩΝ
Private Sub btnAddWorkdays_Click()
    Dim pyear As Integer
    Dim db As DAO.Database
On Error GoTo Err_Handler
    pyear = CInt(Me.txtCurrentYear.Value)
    ' Κλήση της ενσωματωμένης διαδικασίας loadCalendar
    ' της λειτουρ. μονάδας mdlAppt
    Call mdlAppt.loadCalendar(pyear)
Exit_Here:
    Exit Sub
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
    Resume Exit_Here
End Sub

' Κουμπί ΑΚΥΡΩΣΗ
Private Sub btnCancel_Click()
    DoCmd.Close
End Sub

```

B. Κώδικας των λειτουργικών μονάδων

Λειτουργική μονάδα «mdlAppt»

```

Option Compare Database
' Ενσωματωμένη διαδικασία της λειτουργικής μονάδας
' mdlAppt.
' Παράμετροι εισόδου: pyear (έτος)
Public Sub loadCalendar(pyear As Integer)
    Dim db As DAO.Database
    Dim dte As Date
    Dim dteLast As Date
    Dim intYear As Integer
    Dim rs As DAO.Recordset
    Dim strMsg As String
On Error GoTo ErrorHandler
    ' Ορισμός τιμών μεταβλητών:
    ' Τρέχον έτος: intYear, τιμή παραμέτρου pyear
    ' Πρώτη μέρα έτους: dteFirst
    ' Τελευταία μέρα έτους: dteLast
    intYear = IIf(pyear = 0, Year(Date), pyear)
    dteFirst = DateSerial(intYear, 1, 1)
    dteLast = DateSerial(intYear, 12, 31)

    DoCmd.SetWarnings True
    ' Άνοιγμα του πίνακα AppDays με χρήση recordset
    Set db = CurrentDb
    Set rs = db.OpenRecordset("AppDays", dbOpenDynaset, dbAppendOnly)
    ' Όσο η πρώτη μέρα είναι μικρότερη της τελευταίας μέρας του έτους
    ' προσθέτουμε στον πίνακα μία εγγραφή ημερομηνίας
    Do While dteFirst <= dteLast
        rs.AddNew
        rs.Fields("AppDate").Value = dte
        rs.Update
        dteFirst = dteFirst + 1
    Loop
    MsgBox "Οι ημέρες για το έτος " & pyear & " προστέθηκαν με επιτυχία!", _
        vbOKOnly + vbInformation, "medApp | Διαχείριση Ημερών"
    rs.Close
Exit_Here:
    On Error GoTo 0
    Set rs = Nothing
    Set db = Nothing
    Exit Sub
ErrorHandler:
    Select Case Err.Number
        Case 3022
            MsgBox "Υπάρχουν ήδη καταχωρημένες ημέρες για αυτό το έτος! " & vbNewLine & _
                "Για να συνεχίσετε αλλάξτε την τιμή του έτους στο πεδίο.", _

```



```

        vbCritical + vbOKOnly, "medApp | Σφάλμα Καταχώρησης"
        Resume Exit_Here
    Case Else
        MsgBox Err.Number & " " & Err.Description, vbExclamation,
"medApp | Σφάλμα Εφαρμογής"
        Resume Exit_Here
    End Select
End Sub

```

Λειτουργική μονάδα «mdlPatient»

```

Option Compare Database
' Συνάρτηση Boolean. Έλεγχος εαν το AMKA:
' α) αποτελείται από 11 χαρακτήρες, επιστρέφει FALSE
' β) είναι αριθμητικά ψηφία, επιστρέφει FALSE
' σε οποιαδήποτε άλλη περίπτωση επιστρέφει TRUE
Public Function validateAMKA(pAMKA As String) As Boolean
On Error GoTo Err_Handler
    If Len(pAMKA) <> 11 Then
        validateAMKA = False
        Exit Function
    ElseIf IsNumeric(pAMKA) = False Then
        validateAMKA = False
        Exit Function
    Else
        validateAMKA = True
        Exit Function
    End If
Exit_Here:
    Exit Function
Err_Handler:
    DoCmd.Hourglass False
    MsgBox Err.Description, vbExclamation, "medApp | Σφάλμα
Εφαρμογής"
    Resume Exit_Here
End Function

```