

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΝΟΣΗΛΕΥΤΙΚΗΣ

ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΕΙΔΙΚΕΥΣΗ: ΠΛΗΡΟΦΟΡΙΚΗ ΥΓΕΙΑΣ

**ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΟΔΟΝΤΙΑΤΡΙΚΗΣ ΚΛΙΝΙΚΗΣ ΜΕ
ΔΥΝΑΤΟΤΗΤΕΣ ΠΟΛΥΜΕΣΙΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

ΜΑΚΡΗ Γ. ΧΡΗΣΤΟΥ
ΟΔΟΝΤΙΑΤΡΟΥ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΗΝΑ 2017

**ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΟΔΟΝΤΙΑΤΡΙΚΗΣ ΚΛΙΝΙΚΗΣ ΜΕ
ΔΥΝΑΤΟΤΗΤΕΣ ΠΟΛΥΜΕΣΙΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΝΟΣΗΛΕΥΤΙΚΗΣ

ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΕΙΔΙΚΕΥΣΗ: ΠΛΗΡΟΦΟΡΙΚΗ ΥΓΕΙΑΣ

**ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΟΔΟΝΤΙΑΤΡΙΚΗΣ ΚΛΙΝΙΚΗΣ ΜΕ
ΔΥΝΑΤΟΤΗΤΕΣ ΠΟΛΥΜΕΣΙΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

ΜΑΚΡΗ Γ. ΧΡΗΣΤΟΥ
ΟΔΟΝΤΙΑΤΡΟΥ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΗΝΑ 2017

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

ΕΠΙΚ. ΚΑΘΗΓΗΤΗΣ Α. ΠΙΚΡΑΚΗΣ (ΕΠΙΒΛΕΠΩΝ)

ΚΑΘΗΓΗΤΗΣ Ι. ΜΑΝΤΑΣ

ΚΑΘΗΓΗΤΡΙΑ Α. ΚΑΛΟΚΑΙΡΙΝΟΥ

Copyright © Μακρής Χρήστος, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο των απαιτήσεων του Μεταπτυχιακού Προγράμματος Ειδίκευσης στην Πληροφορική Υγείας του Εθνικού Καποδιστριακού Πανεπιστημίου Αθηνών του τμήματος Νοσηλευτικής.

Βεβαιώνω ότι η παρούσα διπλωματική εργασία είναι αποτέλεσμα δικής μου δουλειάς και δεν αποτελεί προϊόν αντιγραφής. Στις δημοσιευμένες ή μη δημοσιευμένες πηγές που αναφέρω έχω χρησιμοποιήσει εισαγωγικά όπου απαιτείται και έχω παραθέσει τις πηγές τους στο τμήμα της βιβλιογραφίας.

Υπογραφή:.....

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω όλους τους καθηγητές και καθηγήτριες του μεταπτυχιακού προγράμματος σπουδών της Πληροφορικής Υγείας. Ιδιαίτερος θα ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Πικράκη Άγγελο, χωρίς την καταλυτική υποστήριξη του οποίου δε θα ήταν δυνατή η ολοκλήρωση του πονήματος αυτού.

ΠΕΡΙΕΧΟΜΕΝΑ

1.	Εισαγωγή	1
1.1.	Ηλεκτρονικός φάκελος υγείας οδοντιατρικού ασθενή	1
1.2.	Νομικό πλαίσιο	1
1.3.	Πρόσθετες λειτουργίες	3
1.4.	Πληροφοριακό σύστημα υποστήριξης οδοντιατρικής κλινικής	5
1.5.	Σχεσιακές βάσεις δεδομένων	6
1.6.	Αντικειμενοστρεφής Προγραμματισμός σε C#	6
1.7.	Η γλώσσα προγραμματισμού C#	7
2.	Η Εφαρμογή	8
2.1.	Γενικές απαιτήσεις	8
2.2.	Ειδικές απαιτήσεις	8
2.2.1.	Πρόγραμμα λειτουργίας κλινικής	8
2.2.2.	Ηλεκτρονικός φάκελος ασθενή	8
3.	Παρουσίαση Της Εφαρμογής	10
3.1.	Εγχειρίδιο χρήσης	10
3.1.1.	Εισαγωγική Φόρμα	10
3.1.2.	Φόρμα «ΑΣΘΕΝΕΙΣ»	11
3.1.3.	Φόρμα Οδοντιατρικού Ιστορικού	13
3.1.4.	Φόρμα Μελέτης – Επεξεργασίας Ακτινογραφίας	15
3.1.5.	Φόρμα Ραντεβού	17
3.1.6.	Φόρμα Γενικού Ιστορικού Υγείας	21
4.	Εργαλεία ανάπτυξης της εφαρμογής	23
4.1.	Ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών	23
4.2.	Τεχνολογίες .NET της Microsoft	27

4.3.	Σύστημα Διαχείρισης Βάσης Δεδομένων Microsoft Access	29
5.	Ανάπτυξη Εφαρμογής Πληροφοριακού Συστήματος Οδοντιατρικής Κλινικής	31
5.1.	Ανάπτυξη Βάσης Δεδομένων	31
5.1.1.	Πίνακας «PatientInfo»	32
5.1.2.	Πίνακας «GeneralHealth»	34
5.1.3.	Πίνακας Diagnosis	35
5.1.4.	Πίνακας Treatments	37
5.1.5.	Πίνακας Multimedia	39
5.1.6.	Πίνακας Doctors	40
5.1.7.	Πίνακας Appointments	42
5.1.8.	Σχέσεις πινάκων	44
5.2.	Ανάπτυξη διεπαφής χρήστη της εφαρμογής	46
5.2.1.	Αρχικοποίηση εφαρμογής	46
5.2.2.	Εισαγωγική Φόρμα	47
5.2.3.	Φόρμα «ΑΣΘΕΝΕΙΣ»	51
5.2.4.	Φόρμα Γενικού Ιστορικού Υγείας	56
5.2.5.	Φόρμα Οδοντιατρικής Εξέτασης	58
5.2.6.	Φόρμα Μελέτης Και Επεξεργασίας Ακτινογραφιών	63
5.2.7.	Κλάση ImageProcessing	66
5.2.8.	Φόρμα Ραντεβού	70
5.2.9.	Εξειδικευμένα Query και παράγωγοι πίνακες στο Dataset	73
6.	Συμπεράσματα - Συζήτηση	79
7.	Μελλοντική Εργασία	80
8.	Βιβλιογραφία	82

9.	Ευρετήριο Εικόνων	84
10.	Περίληψη.....	86
11.	Dissertation.....	88

1. Εισαγωγή

Κατά την καθημερινή οδοντιατρική πράξη είναι επιβεβλημένη, για λόγους επιστημονικούς και νομικούς, η τήρηση φακέλου υγείας κάθε ασθενή. Η επιστήμη των υπολογιστών βρίσκει, φυσικά, εφαρμογή και στον τομέα της οδοντιατρικής, διευκολύνοντας λειτουργίες που αλλιώς απαιτούν σημαντικούς πόρους για τη χειροκίνητη επιτέλεσή τους. Ο σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος οδοντιατρικής κλινικής, που να καλύπτει τις βασικές ανάγκες μιας κλινικής σχετικά με την οργάνωση του προγράμματος ραντεβού και την τήρηση του ηλεκτρονικού φακέλου υγείας των ασθενών.

1.1. Ηλεκτρονικός φάκελος υγείας οδοντιατρικού ασθενή

Το περιεχόμενο του ηλεκτρονικού φακέλου υγείας για έναν οδοντιατρικό ασθενή, ορίζεται αφ' ενός από το νομικό πλαίσιο του κράτους, όσον αφορά στις ελάχιστες απαιτήσεις. Αφ' ετέρου η έκταση και οι εξειδικευμένες λειτουργίες του ορίζονται από την οδοντιατρική επιστήμη και πρακτική. Κατ' αρχάς θα αναφερθούν οι βασικές απαιτήσεις που προκύπτουν από την κείμενη νομοθεσία. Οι ειδικές απαιτήσεις ενός σύγχρονου οδοντιατρείου είναι ιδιαίτερα εκτεταμένες, ιδίως αν πρέπει να καλυφθούν οι ανάγκες οδοντιατρικών ειδικοτήτων, όπως γναθοχειρουργικής, ορθοδοντικής, περιοδοντολογίας, οδοντοπροσθετικής κ.α. Στην παρούσα εργασία θα εστιάσουμε στον κοινό παρονομαστή της κλινικής οδοντιατρικής, και στην κάλυψη των βασικών αναγκών αυτής.

1.2. Νομικό πλαίσιο

Η τήρηση οδοντιατρικού φακέλου υγείας επιβάλλεται από τον ισχύοντα «Κώδικα Οδοντιατρικής Δεοντολογίας» (1). Πιο συγκεκριμένα, στο Άρθρο 12 αναφέρονται τα εξής:

«1. Ο οδοντίατρος υποχρεούται να τηρεί οδοντιατρικό αρχείο, σε ηλεκτρονική, ή μη μορφή, το οποίο περιέχει δεδομένα που συνδέονται αρρήκτως, ή αιτιωδώς με την ασθένεια, ή την υγεία των ασθενών του. Για την τήρηση του αρχείου αυτού και την επεξεργασία των δεδομένων του εφαρμόζονται οι διατάξεις του ν. 2472/1997 (ΦΕΚ 50 Α').

2. Τα οδοντιατρικά αρχεία πρέπει να περιέχουν το ονοματεπώνυμο, το πατρώνυμο, το φύλο, την ηλικία, το επάγγελμα, τη διεύθυνση του ασθενή, τις ημερομηνίες της επίσκεψης, καθώς και κάθε άλλο ουσιώδες στοιχείο, που συνδέεται με την παροχή φροντίδας στον ασθενή, τα ενοχλήματα της υγείας του και το λόγο της επίσκεψης, την πρωτογενή και δευτερογενή διάγνωση, ή την αγωγή, που ακολουθήθηκε.»

Ο νομοθέτης, συνεπώς, ορίζει τη βασική δομή του οδοντιατρικού φακέλου ασθενή. Τα βασικά χαρακτηριστικά του φακέλου που περιγράφει ο νόμος, αποτελούν και ουσιαστικά χαρακτηριστικά ενός πλήρους ιστορικού και φακέλου υγείας ασθενή. Σε άλλη παράγραφο αναφέρεται η υποχρέωση τήρησης του αρχείου για δέκα χρόνια, όσον αφορά σε ιδιωτικά ιατρεία, ή είκοσι χρόνια, όσον αφορά σε κλινικές ή νοσοκομεία, διάστημα που μετρά πάντα από την τελευταία επίσκεψη του ασθενούς. (1)

Στο άρθρο 13 του ιδίου κώδικα, αναφέρονται οι βασικές προϋποθέσεις ασφάλειας δεδομένων, στην περίπτωση που ο οδοντιατρικός φάκελος είναι ηλεκτρονικός.

«Άρθρο: 13

Εάν για την αρχειοθέτηση των προσωπικών δεδομένων του ασθενή γίνεται χρήση ηλεκτρονικού υπολογιστή, ή άλλης παρόμοιας μεθόδου αποθήκευσης, η αρχειοθέτηση είναι επιτρεπτή μόνον εφόσον λαμβάνονται τα κατάλληλα οργανωτικά και τεχνικά μέτρα για την ασφάλεια των δεδομένων και την προστασία τους από τυχαία, ή αθέμιτη καταστροφή, απώλεια, αλλοίωση, απαγορευμένη διάδοση, ή πρόσβαση και κάθε άλλη μορφή αθέμιτης επεξεργασίας. Τηρουμένων των ως άνω προϋποθέσεων, οι οδοντίατροι μπορούν να συνεργάζονται για τη δημιουργία ηλεκτρονικών τραπεζών δεδομένων. Κάθε

τράπεζα πληροφορικής δεδομένων πρέπει να τίθεται υπό τον έλεγχο και την εποπτεία ειδικά διορισμένου οδοντιάτρου.» (1)

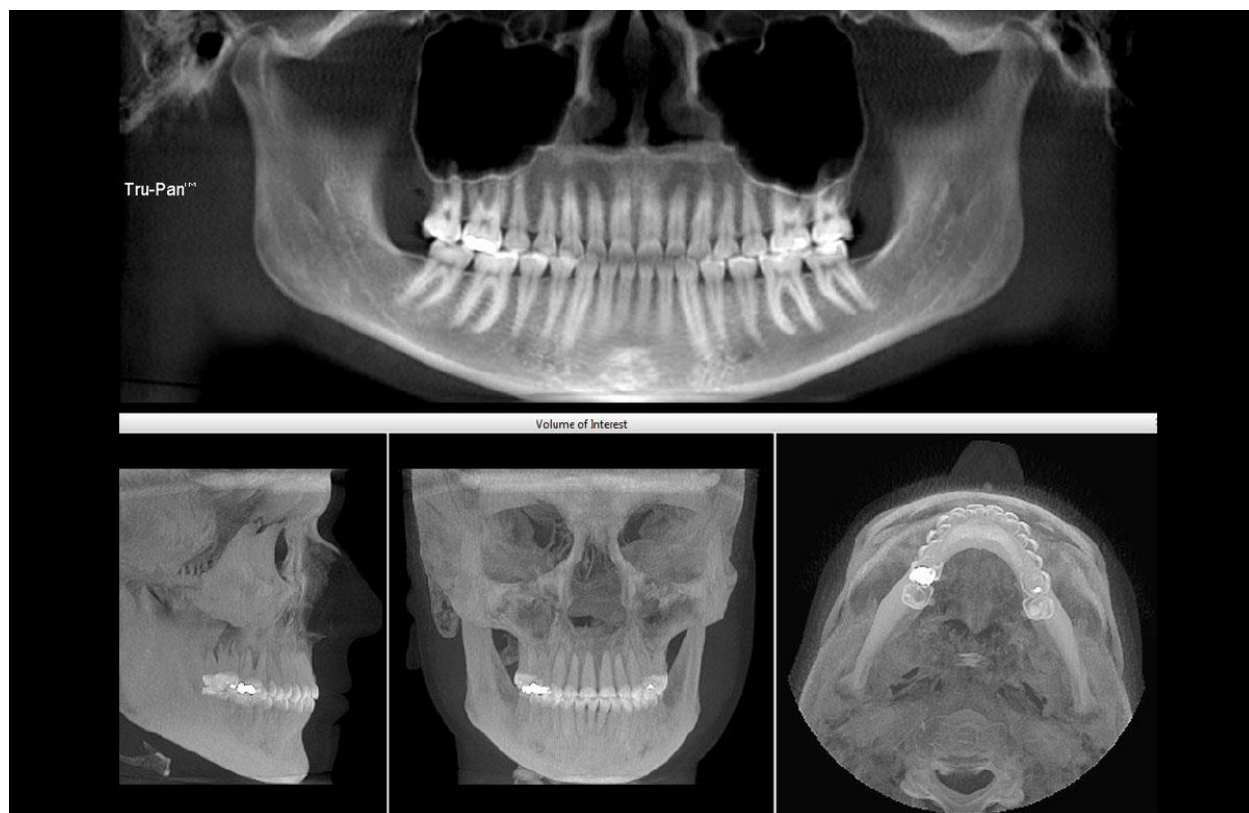
Από το παραπάνω άρθρο, αλλά και σε όλη την έκταση του Κώδικα, γίνεται αντιληπτή η μέριμνα του νομοθέτη για την τήρηση του ιατρικού απορρήτου, καθώς και των λοιπών χαρακτηριστικών ασφαλείας του ηλεκτρονικού αρχείου.

1.3. Πρόσθετες λειτουργίες

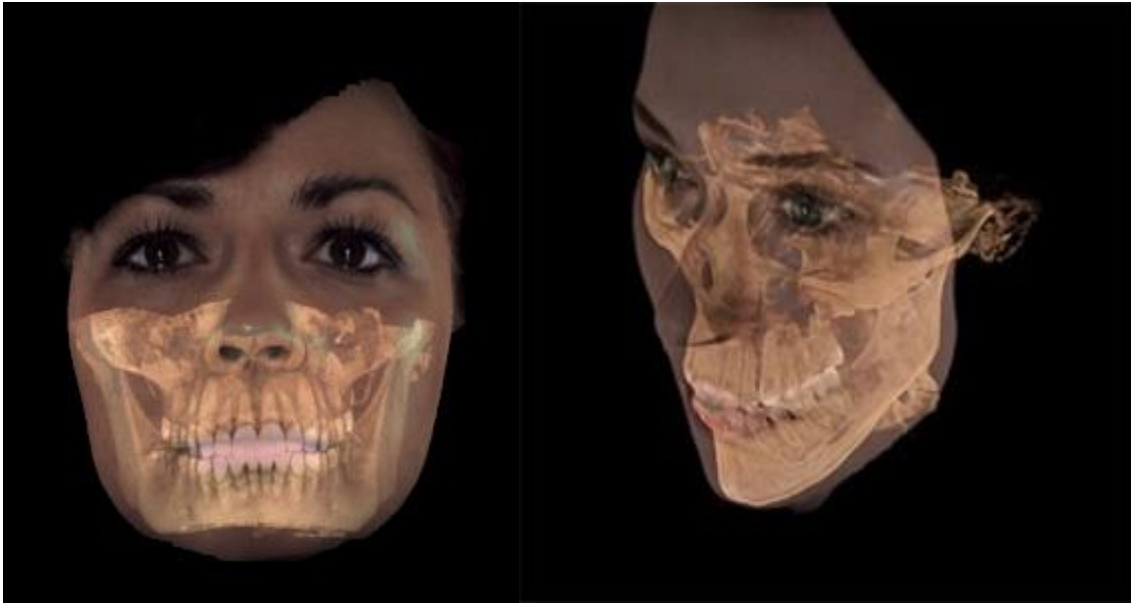
Η αυξημένη χρήση ψηφιακών μέσων διάγνωσης και θεραπείας του οδοντιατρικού ασθενή, δημιουργεί μεγάλο αριθμό ψηφιακών αρχείων δεδομένων τα οποία συνδέονται άρρηκτα με τον ασθενή. Αυτά τα αρχεία μπορούν να περιλαμβάνουν ψηφιακές φωτογραφίες του ασθενούς (ενδοστοματικές και εξωστοματικές), ψηφιακές ακτινογραφίες των δοντιών (ενδοστοματικές κ.α) ή και ολόκληρων των γνάθων (πανοραμικές), ψηφιακές αξονικές τομογραφίες του γναθοπροσωπικού κρανίου, καθώς και τις τρισδιάστατες ανασυνθέσεις που προκύπτουν από αυτές. Συνεχώς προκύπτουν νέοι τύποι δεδομένων, για παράδειγμα τα αρχεία τρισδιάστατων ανασυνθέσεων των οδοντοστοιχιών με ρεαλιστικές χρωματικές υφές. Η ομαλή αποθήκευση και σύνδεση των παραπάνω αρχείων με τον οδοντιατρικό φάκελο του ασθενή, προϋποθέτει τη χρήση ενός εξελεγμένου πληροφοριακού συστήματος, με δυνατότητες αποθήκευσης, ανάκλησης και επεξεργασίας των πολυμεσικών αυτών αρχείων.



Εικόνα 1. Σύγχρονη ψηφιακή πανοραμική ακτινογραφία γνάθων - ορθοπαντομογραφία



Εικόνα 2. Ψηφιακή αξονική τομογραφία CBCT με 3d ανασύσταση κρανίου



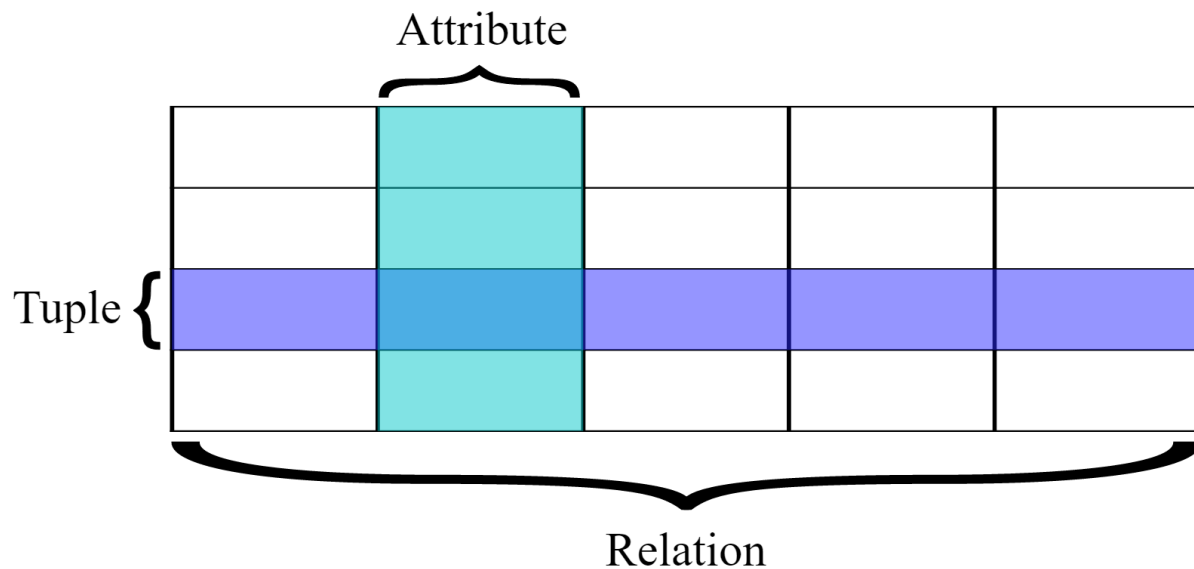
Εικόνα 3. Υπέρθεση 3d ανασύστασης προσωπικού κρανίου σε 3d μοντέλο προσώπου ασθενούς

1.4. Πληροφοριακό σύστημα υποστήριξης οδοντιατρικής κλινικής

Μια σύγχρονη οδοντιατρική κλινική έχει ιδιαίτερες απαιτήσεις οργάνωσης του προγράμματος λειτουργίας της. Σε ένα τυπικό σενάριο λειτουργίας μιας ημέρας, πολλοί οδοντίατροι καλούνται να προσφέρουν υπηρεσίες υγείας σε ένα πλήθος ασθενών, χρησιμοποιώντας μια πλειάδα δομών όπως εξεταστήρια, οδοντιατρικές έδρες, απεικονιστικά – ακτινολογικά συστήματα, χώρους επικοινωνίας και ενημέρωσης, και άλλων. Το πληροφοριακό σύστημα της οδοντιατρικής κλινικής μπορεί να ανταποκριθεί σε όλες τις παραπάνω απαιτήσεις, εξασφαλίζοντας ομαλή ροή εργασίας και αποτελεσματικό χρονικό προγραμματισμό για όλα τα εμπλεκόμενα μέρη: ασθενείς, ιατρούς και δομές της κλινικής.

1.5. Σχεσιακές βάσεις δεδομένων

Οι σχεσιακές βάσεις δεδομένων βασίζονται στο σχεσιακό μοντέλο δεδομένων, το οποίο πρότεινε ο Ε. Κοντ το 1970. (2) Σύμφωνα με το εν λόγω μοντέλο, τα δεδομένα οργανώνονται σε πίνακες που περιλαμβάνουν εγγραφές, ιδιότητες και τις μεταξύ τους συσχετίσεις. (3)



Εικόνα 4 Το σχεσιακό μοντέλο δεδομένων κατά Κοντ (3)

Απαραίτητο για τη λειτουργία της ΒΔ είναι ένα σύστημα διαχείρισης βάσης δεδομένων, το οποίο αναλαμβάνει τη διαχείριση των δεδομένων της ΒΔ, με κυριότερες λειτουργίες την προσθήκη, διαγραφή, τροποποίηση και αναζήτηση των εγγραφών. Το ΣΔΒΔ παρέχει την αναγκαία διεπαφή μεταξύ των εφαρμογών, των χρηστών και των δεδομένων της βάσης.

1.6. Αντικειμενοστρεφής Προγραμματισμός σε C#

Ο αντικειμενοστρεφής προγραμματισμός (object oriented programming) είναι η προγραμματιστική μεθοδολογία που βασίζεται στη χρήση των αντικειμένων (objects),

δηλαδή ενιαίων δομών που περιέχουν δεδομένα, και διαδικασίες που επενεργούν στα συγκεκριμένα δεδομένα. Κεντρικά στοιχεία του αντικειμενοστρεφούς προγραμματισμού αποτελούν οι κλάσεις (classes), τα αντικείμενα που αυτές παράγουν, οι μέθοδοι (methods) και τα πεδία (fields). Σημαντικές λειτουργίες κατά τον αντικειμενοστρεφή προγραμματισμό είναι η ενθυλάκωση δεδομένων (data encapsulation), η αφαίρεση δεδομένων (data abstraction), η κληρονομικότητα (inheritance) και η υπερφόρτωση ή υποσκέλιση μεθόδων (method overloading, method overriding). (4)

1.7. Η γλώσσα προγραμματισμού C#

Η γλώσσα προγραμματισμού C# είναι μια γλώσσα προγραμματισμού γενικής χρήσεως, αντικειμενοστρεφούς προγραμματισμού που αναπτύχθηκε από την εταιρία Microsoft το 2001. Παρόλο που εστιάζει στο αντικειμενοστρεφές υπόδειγμα, είναι πολύ-υποδειγματική, ενσωματώνοντας χαρακτηριστικά λειτουργικού, γενικού και δομημένου προγραμματισμού, μεταξύ άλλων. Ο κώδικας που προκύπτει από τον μεταγλωττιστή (compiler), είναι ενδιάμεσος κώδικας, ο οποίος στοχεύει στην Common Language Infrastructure (CLI) για τη μετατροπή του σε εκτελέσιμο κώδικα. Στα πιο συνηθισμένα σενάρια, η CLI υλοποιείται μέσω της υποδομής “.NET”. Η γλώσσα συνεχίζει να εξελίσσεται συνεχώς, αποκλίνοντας βαθμιαία από παρόμοιες γλώσσες του χώρου όπως η Java. (5)

2. Η Εφαρμογή

2.1. Γενικές απαιτήσεις

Ο ρόλος που έχει να καλύψει το πληροφοριακό σύστημα οδοντιατρικής κλινικής είναι διπλός. Αφ' ενός θα αναλάβει την οργάνωση του προγράμματος λειτουργίας κλινικής (ραντεβού) της ημέρας, συσχετίζοντας τους ασθενείς, τους ιατρούς και τις διαθέσιμες δομές παροχής φροντίδας (ιατρεία με οδοντιατρικές έδρες). Αφ' ετέρου θα τηρεί τον ηλεκτρονικό φάκελο υγείας του οδοντιατρικού ασθενούς και θα υλοποιεί τη χρήση και ενημέρωσή του.

2.2. Ειδικές απαιτήσεις

2.2.1. Πρόγραμμα λειτουργίας κλινικής

Στην εν λόγω κλινική λειτουργούν τρεις χώροι – ιατρεία, εξοπλισμένοι με οδοντιατρική έδρα, κατάλληλα για την παροχή φροντίδας σε οδοντιατρικούς ασθενείς. Σε κάθε ιατρείο αντιστοιχεί ξεχωριστό πρόγραμμα ημέρας. Στην κλινική μπορούν να εργάζονται πολλοί οδοντίατροι, συνεπώς είναι σημαντικό να γνωρίζουμε σε κάθε ραντεβού ασθενή, ποιος οδοντίατρος θα απασχοληθεί και σε ποιο ιατρείο. Για κάθε ραντεβού, είναι απαραίτητο να καταγράφεται το ονοματεπώνυμο του ασθενούς (σε σύνδεση με τον πλήρη φάκελό του), η ημερομηνία και η ώρα ραντεβού, η διάρκεια, ο επιβλέπων ιατρός, καθώς και ειδικά σχόλια σχετικά με το ραντεβού (σχεδιαζόμενη επέμβαση κ.α).

2.2.2. Ηλεκτρονικός φάκελος ασθενή

Στον ηλεκτρονικό φάκελο του ασθενή θα τηρηθεί ένα πλήθος στοιχείων. Τα γενικά στοιχεία του ασθενούς περιλαμβάνουν ονοματεπώνυμο, ημερομηνία γέννησης,

πατρώνυμο ή όνομα κηδεμόνα, αριθμό μητρώου κοινωνικής ασφάλισης (ΑΜΚΑ), διεύθυνση κατοικίας, τηλέφωνο επικοινωνίας. Θα τηρηθεί επίσης ιστορικό γενικής υγείας, στο οποίο καταγράφουμε τυχόν χρόνιες παθήσεις, αλλεργίες σε φάρμακα και υφιστάμενη φαρμακευτική αγωγή. Το οδοντιατρικό ιστορικό περιλαμβάνει και το μεγαλύτερο όγκο στοιχείων. Σε αυτό καταγράφονται λεπτομέρειες για την κατάσταση κάθε δοντιού, σύμφωνα με το παγκόσμιο (universal) σύστημα κατάταξης του οδοντικού φραγμού. Καταγράφονται επίσης γενικές πληροφορίες για την κατάσταση του στόματος, ούλων, χειλέων κλπ. Τέλος, στο οδοντιατρικό ιστορικό βρίσκονται και τα πολυμεσικά ψηφιακά αρχεία του ασθενούς. Κύριο λόγο έχουν οι ψηφιακές ακτινογραφίες και οι εξωστοματικές και ενδοστοματικές φωτογραφίες του ασθενή, με δυνατότητα άμεσης ανάκλησης και προβολής αυτών. Μπορούν επίσης να αποθηκευτούν στην εφαρμογή πολλών ειδών ψηφιακά αρχεία, όπως αξονικές τομογραφίες και οι τρισδιάστατες αναπαραστάσεις αυτών, αρχεία που προκύπτουν από τρισδιάστατη σάρωση εκμαγείων του οδοντικού φραγμού, φωτογραφίες των εκμαγείων, ειδικά αρχεία ηλεκτρονικής καταγραφής της σύγκλεισης των δοντιών και των λειτουργικών κινήσεων της κάτω γνάθου, ειδικά αρχεία και φωτογραφίες χρωματοληψίας των δοντιών, αρχεία ήχου και βίντεο με ομιλία του ασθενούς για τη διερεύνηση οδοντικής αιτιολογίας προβλημάτων άρθρωσης λόγου, και άλλα είδη ψηφιακών αρχείων.

3. Παρουσίαση Της Εφαρμογής

Η εφαρμογή που θα αποτελέσει το πληροφοριακό σύστημα οδοντιατρικής κλινικής είναι μια εφαρμογή για Microsoft Windows, η οποία ενσωματώνει στη λειτουργία της μια βάση δεδομένων. Λειτουργεί σε όλους τους υπολογιστές με λειτουργικό MS Windows έκδοσης 7 ή μεταγενέστερης και προϋποθέτει ενημερωμένη προϋπάρχουσα εγκατάσταση του .NET Framework. Η εφαρμογή έχει σχεδιαστεί με τρόπο που να καλύπτει τις απαιτήσεις μιας οδοντιατρικής κλινικής, όπως αυτές αναφέρθηκαν πιο πάνω.

3.1. Εγχειρίδιο χρήσης

3.1.1. Εισαγωγική Φόρμα

Ξεκινώντας την εφαρμογή, εμφανίζεται η εισαγωγική φόρμα στην οποία πρέπει να εισάγουμε τους ειδικούς κωδικούς χρήστη. Τα κουμπιά «ΑΣΘΕΝΕΙΣ» και «PANTEBOY» είναι εκ των προτέρων απενεργοποιημένα, και δεν είναι δυνατόν να πατηθούν. Για την ενεργοποίηση οποιουδήποτε από τα παραπάνω κουμπιά, είναι απαραίτητη η εισαγωγή του σωστού κωδικού στο κατάλληλο πεδίο κειμένου. Μετά την εισαγωγή του κωδικού χρήστη, μπορούμε να επιλέξουμε το κουμπί «ΑΣΘΕΝΕΙΣ» για πρόσβαση στο αρχείο ηλεκτρονικών φακέλων οδοντιατρικής υγείας, ή το κουμπί «PANTEBOY», για πρόσβαση στο σύστημα προγραμματισμού εργασιών της κλινικής. Στη φόρμα υπάρχει και το κουμπί «ΕΞΟΔΟΣ», το οποίο τερματίζει τη χρήση της εφαρμογής. Ο έλεγχος του κωδικού γίνεται αυτόματα κατά την πληκτρολόγηση αυτού, χωρίς να χρειαστεί να πατηθεί κάποιο άλλο κουμπί.



Εικόνα 5. Εισαγωγική Φόρμα

3.1.2. Φόρμα «ΑΣΘΕΝΕΙΣ»

Μετά την επιλογή του κουμπιού «ΑΣΘΕΝΕΙΣ» στην κεντρική οθόνη, εμφανίζεται η φόρμα των γενικών στοιχείων των ασθενών. Σε αυτή τη φόρμα μπορούμε να κάνουμε αναζήτηση για υπάρχοντες ασθενείς, να προσθέσουμε νέους, να δούμε και να επεξεργαστούμε τα βασικά στοιχεία των ασθενών και να μεταβούμε σε άλλες υποενότητες – φόρμες.

Πιο συγκεκριμένα, η αναζήτηση υπάρχοντων φακέλων ασθενών γίνεται βάσει επιθέτου ή ΑΜΚΑ. Στο αντίστοιχο κουτί κειμένου (textbox) πληκτρολογούμε το επίθετο του ασθενούς ή το ΑΜΚΑ αυτού. Τα αποτελέσματα της έρευνας εμφανίζονται στον σχετικό πίνακα και ανανεώνονται με το πάτημα κάθε πλήκτρου. Για αυτό το λόγο δεν υπάρχει ειδικό κουμπί «αναζήτησης». Ο πίνακας αποτελεσμάτων εμφανίζει το επίθετο, το όνομα και το ΑΜΚΑ του ασθενούς.

ΑΙΣΘΗΣΗ

ΑΝΑΖΗΤΗΣΗ

ΕΠΙΘΕΤΟ: ΑΜΚΑ:

ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΑΜΚΑ
Θεοδώρα	Θεόδωρος	3216549870
Αθανασίου	Γεώργιος	1241251232
Μάρκου	Μαρία	4125215243
Λετίου	Ελένη	2412312541
Ασθενής	Πρώτος	2345678948
Ασθενής	Δεύτερος	9876543210
Ιωάννου	Ιωάννης	1234567890

ΕΠΙΘΕΤΟ: ΘΕΟΔΩΡΟΥ

ΟΝΟΜΑ: ΘΕΟΔΩΡΟΣ

ΑΜΚΑ: 3216549870

ΟΝΟΜΑ ΚΗΔΕΜΟΝΑ:

ΗΜ/ΝΙΑ ΓΕΝΝΗΣΗΣ: Thursday April 6, 2000

ΤΗΛΕΦΩΝΟ: 7896841383

ΔΙΕΥΘΥΝΣΗ: Θεσσαλονίκη

 ΠΡΟΣΘΗΚΗ
  ΔΙΑΓΡΑΦΗ
  ΑΠΟΘΗΚΕΥΣΗ

ΠΑΝΤΕΒΟΥ
 ΓΕΝΙΚΟ ΙΣΤΟΡΙΚΟ
 ΟΔΟΝΤΙΑΤΡΙΚΟ ΙΣΤΟΡΙΚΟ
 ΕΞΟΔΟΣ

Εικόνα 6. Φόρμα «Ασθενείς»

Επιλέγοντας έναν ασθενή στον πίνακα αναζήτησης, εμφανίζεται το σύνολο των γενικών του στοιχείων σε ξεχωριστά κουτιά κειμένου. Οι πληροφορίες που εμφανίζονται περιλαμβάνουν τα, επίθετο, όνομα, ΑΜΚΑ, όνομα πατρός ή κηδεμόνα, ημερομηνία γέννησης, τηλέφωνο επικοινωνίας και διεύθυνση κατοικίας. Οι πληροφορίες στα κουτιά κειμένου μπορούν να υποστούν επεξεργασία και να αποθηκευτούν με χρήση του αντίστοιχου κουμπιού «ΑΠΟΘΗΚΕΥΣΗ». Το κουμπί «ΠΡΟΣΘΗΚΗ» δημιουργεί ένα νέο αρχείο ηλεκτρονικού οδοντιατρικού φακέλου υγείας, και εν συνεχεία συμπληρώνουμε τα κενά κουτιά κειμένου και επιλέγουμε «ΑΠΟΘΗΚΕΥΣΗ». Σε περίπτωση λανθασμένης πληκτρολόγησης ενός ασθενούς, υπάρχει η δυνατότητα διαγραφής των στοιχείων του. Εάν, όμως, έχουν ήδη γίνει καταχωρήσεις στο μητρώο εξετάσεων και ιστορικού, η διαγραφή αυτών των στοιχείων δεν είναι δυνατή. Αυτό συμβαίνει για να μην είναι δυνατή η απώλεια του αρχείου ασθενών, όπως προβλέπεται από το σχετικό νόμο.

Από αυτή τη φόρμα μπορούμε να εκτελέσουμε και άλλες λειτουργίες. Μια είναι η δημιουργία νέου ραντεβού για τον επιλεγμένο ασθενή, πατώντας το κουμπί «ΠΑΝΤΕΒΟΥ», με το οποίο μεταφερόμαστε στην αντίστοιχη φόρμα διαχείρισης ραντεβού. Μπορούμε επίσης να μεταφερθούμε στο ιστορικό γενικής υγείας του

ασθενούς, στο οδοντιατρικό ιστορικό του ασθενούς, πατώντας τα αντίστοιχα κουμπιά. Τέλος μπορούμε να τερματίσουμε την εφαρμογή, πατώντας το κουμπί «ΕΞΟΔΟΣ».

3.1.3. Φόρμα Οδοντιατρικού Ιστορικού

Η εν λόγω φόρμα τηρεί το ειδικό οδοντιατρικό ιστορικό του κάθε ασθενούς.

The screenshot displays the 'FormDental' application window. At the top, there's a header with the application name and standard window controls. Below this, a date selector shows 'Tuesday, June 6, 2017'. To the right, the patient's name 'Θεοδώρου Θεόδωρος' is entered. The main area contains two columns of 16 input fields each, numbered 11 to 38, corresponding to the teeth in a dental arch. To the right of these fields is a large X-ray image of a full dental arch. Below the X-ray are two smaller images: a panoramic X-ray and a blue-tinted X-ray. At the bottom, there's a large text area labeled 'ΣΗΜΕΙΩΣΕΙΣ' (Notes) and four buttons: 'ΑΣΘΕΝΕΙΣ', 'PANTEBOY', 'ΕΙΣΑΓΩΓΗ ΑΚΤΙΝΟΓΡΑΦΙΩΝ ΕΙΚΟΝΩΝ', and 'ΕΞΟΔΟΣ'.

Εικόνα 7. Φόρμα Οδοντιατρικού Ιστορικού

Το κύριο στοιχείο της είναι τα 32 κουτιά κειμένου, με αντιστοιχία σε κάθε δόντι του μόνιμου φραγμού του ασθενούς. Σε αυτά τα κουτιά καταγράφεται κάθε εύρημα σε σχέση με την υγεία του δοντιού, κάθε εργασία που προϋπάρχει ή οτιδήποτε επιθυμεί ο οδοντίατρος να σημειώσει και εντοπίζεται μόνο σε συγκεκριμένο δόντι.

Κατά την πληκτρολόγηση οδοντιατρικών όρων στα παραπάνω κουμπιά, ενεργοποιείται αυτόματα η λειτουργία αυτόματης συμπλήρωσης συνήθους ορολογίας. Μια προκαθορισμένη λίστα περιέχει ένα σύνολο συχνά χρησιμοποιούμενων όρων και

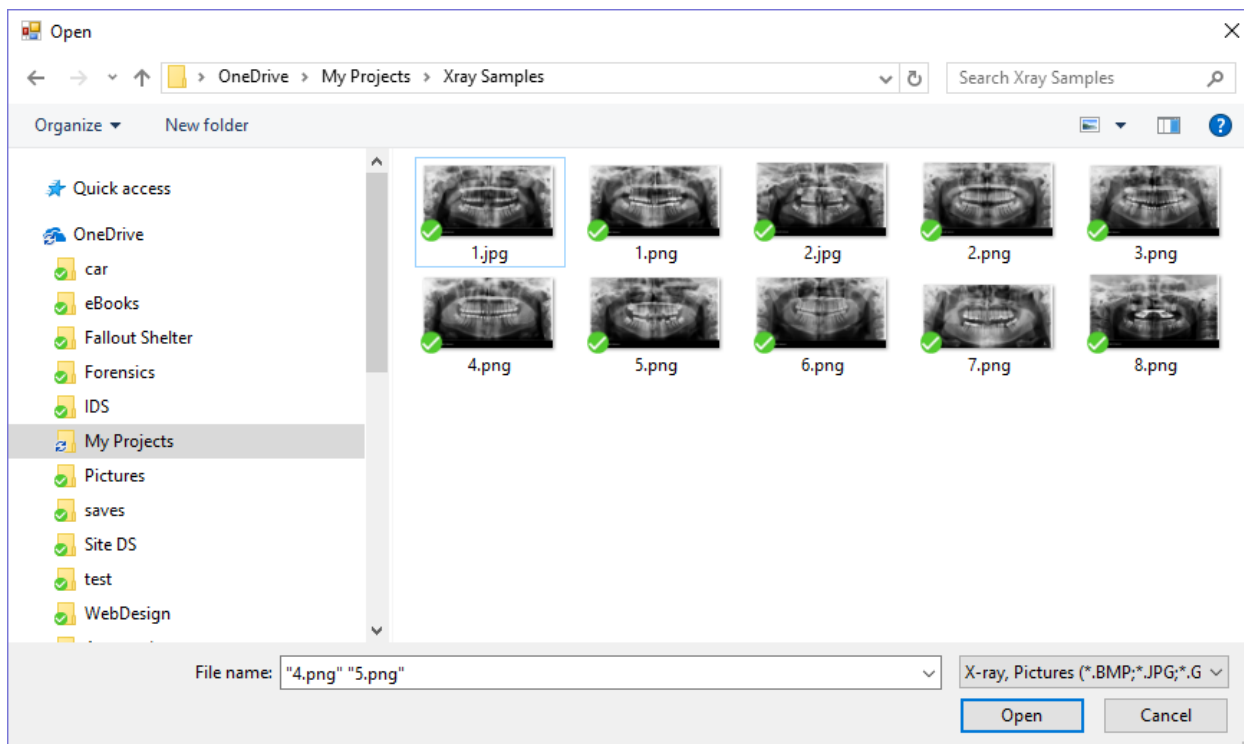
φράσεων της οδοντιατρικής, με τη βοήθεια της οποίας συμπληρώνεται αυτόματα το κείμενο κατά την πληκτρολόγηση. Η λειτουργία αυτή επιταχύνει πολύ τη συμπλήρωση του οδοντιατρικού ιστορικού του ασθενούς, μειώνοντας το χρόνο που χρειάζεται για την εισαγωγή συχνών και επαναλαμβανόμενων όρων στον φάκελο.

Στον χώρο των σημειώσεων, καταγράφεται οποιοδήποτε εύρημα έχει γενικό χαρακτήρα και δεν αφορά σε συγκεκριμένο δόντι ή που αφορά γενικότερα τη στοματική κοιλότητα όπως χείλη, γλώσσα, παρειές, υπερώα και φάρυγγα. Μπορούν ακόμα να καταγραφούν ακτινογραφικά ευρήματα, πληροφορίες για τη σύγκλειση και άλλες σχετικές παρατηρήσεις.

Τα κουμπιά «ΑΣΘΕΝΕΙΣ» και «PANTEBOY» μας μεταφέρουν άμεσα στις αντίστοιχες φόρμες. Αν μεταβούμε στη φόρμα των ραντεβού, μπορούμε να κλείσουμε απευθείας ραντεβού για τον συγκεκριμένο, επιλεγμένο ασθενή μας.

Στο πλαίσιο άνω δεξιά εμφανίζεται η πανοραμική ακτινογραφία του ασθενή, συνεπικουρώντας έτσι στη συμπλήρωση του οδοντιατρικού ιστορικού. Ακριβώς κάτω από την ακτινογραφία του ασθενή, βρίσκεται ένα πλαίσιο που εμφανίζει το σύνολο των ακτινογραφιών και λοιπών εικόνων του ασθενούς. Το πλαίσιο αυτό φέρει ράβδο κύλισης, ενώ όταν επιλέξουμε μια ακτινογραφία από τις εμφανιζόμενες, αυτή εμφανίζεται στο κυρίως πλαίσιο εικόνας.

Η εισαγωγή των ακτινογραφιών που εμφανίζονται στο πλαίσιο συλλογής, γίνεται με πάτημα του κουμπιού «ΕΙΣΑΓΩΓΗ ΑΚΤΙΝΟΓΡΑΦΙΩΝ – ΕΙΚΟΝΩΝ». Το κουμπί αυτό ανοίγει ένα τυποποιημένο παράθυρο διαλόγου – επιλογής αρχείων. Είναι δυνατή η επιλογή μόνο αρχείων εικόνας, βάσει του ενεργού φίλτρου αρχείων (bmp, jpg, png). Η επιλογή των αρχείων έχει ως αποτέλεσμα την αντιγραφή τους από μια εξωτερική τοποθεσία (cd, μνήμη flash, άλλος φάκελος στον υπολογιστή) και τη σύνδεσή τους με το φάκελο κάθε ασθενή. Τα αρχεία αντιγράφονται στη δομή καταλόγου του προγράμματος, ώστε να αποτελούν αναπόσπαστο μέρος του.



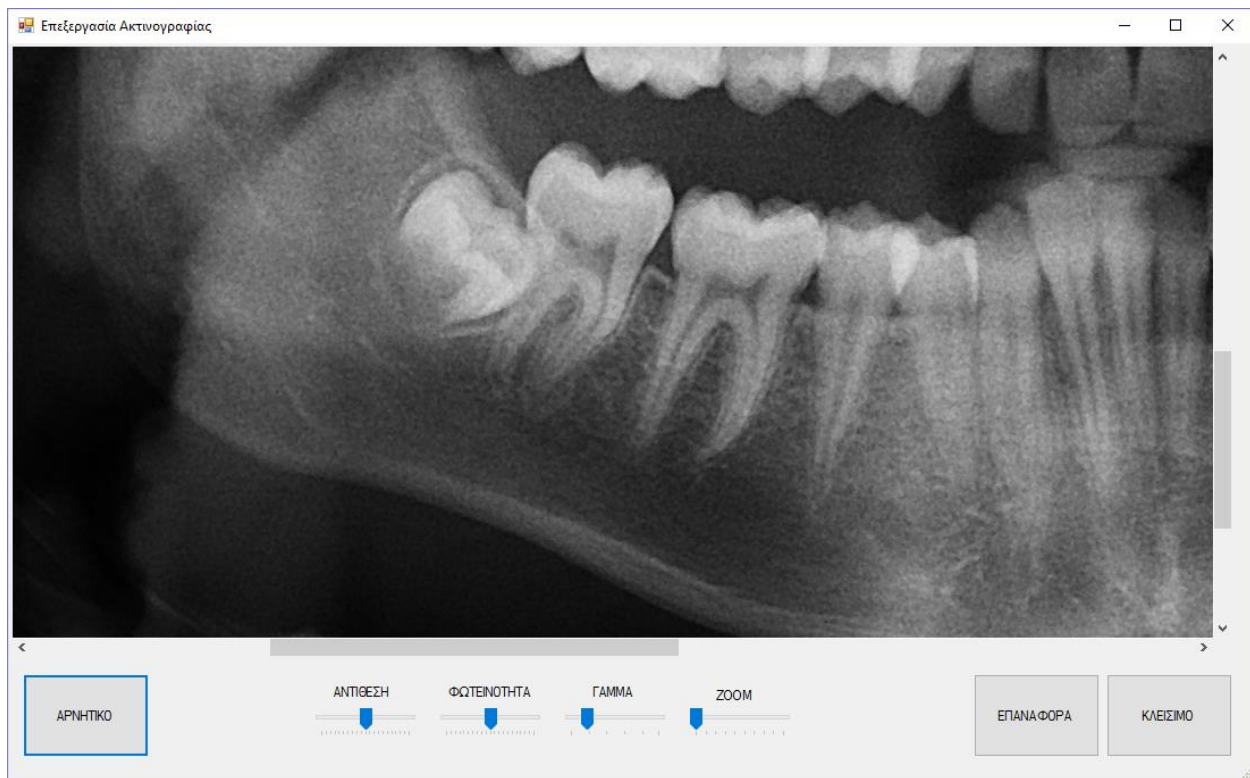
Εικόνα 8. Το παράθυρο εισαγωγής ακτινογραφιών - εικόνων

Τέλος, κάνοντας διπλό κλικ πάνω στην επισκόπηση της ακτινογραφίας, ανοίγει μια νέα φόρμα, μέσω της οποίας είναι δυνατή η μελέτη και επεξεργασία της επιλεγμένης ακτινογραφίας.

3.1.4. Φόρμα Μελέτης – Επεξεργασίας Ακτινογραφίας

Μέσω της φόρμας αυτής είναι δυνατή η μελέτη μιας ακτινογραφίας (αυτής που επιλέχτηκε στη φόρμα οδοντιατρικού ιστορικού) και η ρύθμιση ενός πλήθους παραμέτρων της εικόνας. Η συμβολή αυτής της φόρμας είναι καθοριστική για την ορθή μελέτη της περίπτωσης του ασθενούς και αποτελεί σημαντικό διαγνωστικό εργαλείο για τον οδοντίατρο. Η επεξεργασία της ακτινογραφίας με ηλεκτρονικό υπολογιστή προσφέρει αναντικατάστατες δυνατότητες στον κλινικό οδοντίατρο, που δεν είναι δυνατόν να επιτευχθούν με άλλα μέσα.

Η φόρμα μελέτης έχει ως εξής:



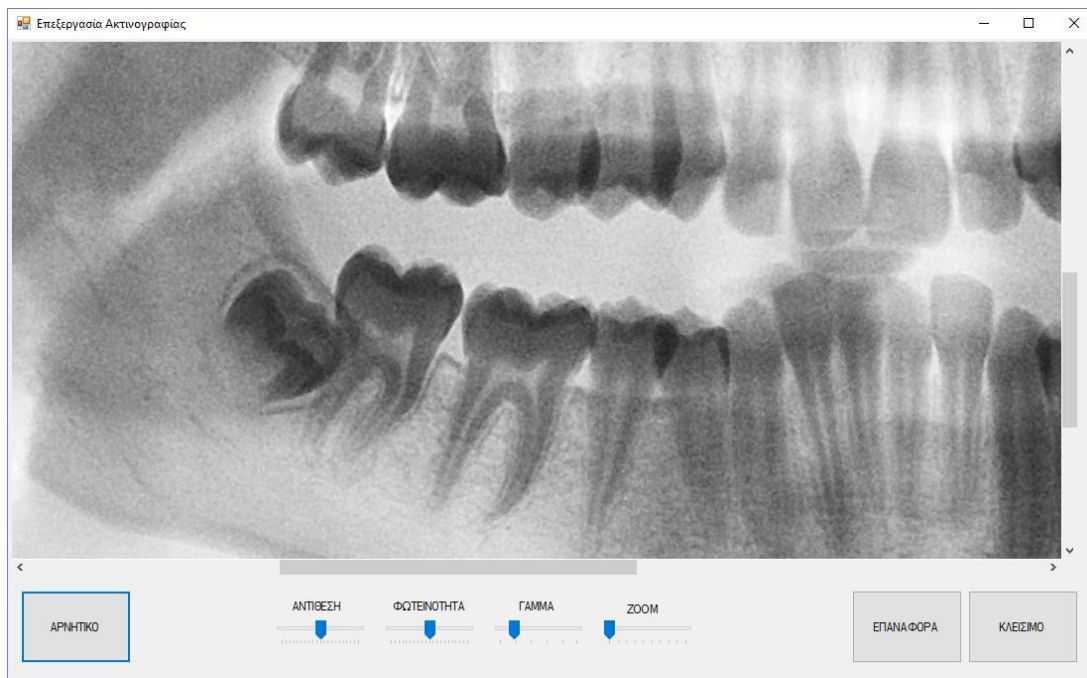
Εικόνα 9. Φόρμα Μελέτης - Επεξεργασίας Ακτινογραφίας

Στη φόρμα αυτή υπάρχει μια σειρά από χειριστήρια και κουμπιά.

Το κουμπί «ΑΡΝΗΤΙΚΟ» δημιουργεί μια πλήρη αντιστροφή των χρωμάτων (φωτεινότητας) της ακτινογραφίας, επιτελώντας έτσι μια ιδιαίτερος χρήσιμη διαγνωστική λειτουργία.

Μια σειρά ρυθμιστικών κύλισης ελέγχουν τις λειτουργίες Αντίθεσης (Contrast), Φωτεινότητας (Brightness), Γάμμα (Gamma) και Μεγέθυνσης (Zoom).

Το κουμπί «ΕΠΑΝΑΦΟΡΑ» ακυρώνει οποιαδήποτε υπάρχουσα επεξεργασία στην ακτινογραφία και την επαναφέρει στην αρχική της κατάσταση. Το κουμπί «ΚΛΕΙΣΙΜΟ» κλείνει το παρόν παράθυρο και μας επιστρέφει στη φόρμα οδοντιατρικού ιστορικού.



Εικόνα 10. Επεξεργασία Ακτινογραφίας - Αρνητικό

Αξίζει να σημειώσουμε ότι η εμφάνιση της φόρμας επεξεργασίας ακτινογραφιών δεν επικαλύπτει πλήρως την προηγούμενη φόρμα οδοντιατρικού ιστορικού, επιτρέποντας έτσι την ταυτόχρονη μελέτη της ακτινογραφίας και την καταχώρηση σχολίων στο ιστορικό του ασθενούς.

3.1.5. Φόρμα Ραντεβού

Σε αυτή τη φόρμα γίνεται η διαχείριση των ραντεβού της κλινικής. Όπως έχει αναφερθεί, η πρότυπη κλινική του υποδείγματος διαθέτει τρεις οδοντιατρικές έδρες (θέσεις εργασίας οδοντιάτρου), στις οποίες εργάζονται πολλοί οδοντίατροι. Το πλήθος των οδοντιάτρων είναι θεωρητικά απεριόριστο, αλλά για πρακτικούς λόγους έχουν προστεθεί πέντε.

The screenshot shows the 'PANTEBOY' software window. At the top, it displays 'ΕΝΕΡΓΟΣ ΦΑΚΕΛΟΣ' (Active File) and the patient's name 'Ιωάννου Ιωάννης'. On the left, there is a calendar for June 2017, with the 6th highlighted. Below the calendar are fields for 'ΩΡΑ:' (Time) set to 12:00, 'ΔΙΑΡΚΕΙΑ:' (Duration) set to 30 minutes, and a dropdown for 'ΙΑΤΡΟΣ:' (Doctor) with 'Ιατρ. Τρίτος' selected. There are also buttons for 'ΠΡΟΣΘΗΚΗ PANTEBOY', 'ΕΠΙΛΟΓΗ ΑΣΘΕΝΟΥΣ', and 'ΔΙΑΓΡΑΦΗ PANTEBOY'. The main area contains three appointment tables labeled 'ΙΑΤΡΕΙΟ 1', 'ΙΑΤΡΕΙΟ 2', and 'ΙΑΤΡΕΙΟ 3'. Each table has columns for 'ΩΡΑ' (Time), 'ΕΠΙΘΕΤΟ' (Surname), and 'ΟΝΟΜΑ' (Name). Below these tables is a large empty box labeled 'ΛΕΙΤΟΜΕΡΕΙΕΣ PANTEBOY'. At the bottom right, there is a button labeled 'ΟΔΟΝΤΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ'.

ΩΡΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
9:00 AM	Θεοδώρου	Θεόδωρος
10:00 AM	Αθανασίου	Γεώργιος
11:00 AM	Λετίου	Ελένη

ΩΡΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
9:00 AM	Μάρκου	Μαρία
12:00 PM	Ιωάννου	Ιωάννης

ΩΡΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
10:00 AM	Ιωάννου	Ιωάννης
1:00 PM	Αθανασίου	Γεώργιος

Εικόνα 11. Φόρμα Ραντεβού

Κύριο χαρακτηριστικό αυτής της φόρμας είναι οι τρεις πίνακες των ραντεβού, ένας για κάθε ιατρείο – εξεταστήριο με οδοντιατρική έδρα. Σε κάθε πίνακα φαίνονται τα ραντεβού της επιλεγμένης ημέρας με μορφή στηλών και γραμμών που περιλαμβάνουν την ώρα του ραντεβού και το ονοματεπώνυμο του ασθενούς. Κάθε ραντεβού μπορεί να επιλεγεί με απλό κλικ και να διαγραφεί με το αντίστοιχο κουμπί κάτω από τους πίνακες.

Το κουμπί «ΕΠΙΛΟΓΗ ΑΣΘΕΝΟΥΣ» κλείνει την παρούσα φόρμα και μας επιστρέφει στην φόρμα «ΑΣΘΕΝΕΙΣ», όπου μπορούμε να επιλέξουμε διαφορετικό ασθενή.

Το κουμπί «ΟΔΟΝΤΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ» μας μεταφέρει στη φόρμα του οδοντιατρικού ιστορικού του ασθενούς, όπου μπορούμε να δούμε το ιστορικό θεραπειών αλλά και τις ακτινογραφίες – φωτογραφίες του ασθενή.

Η διαδικασία προσθήκης ραντεβού εκτελείται από τα διάφορα στοιχεία στην αριστερή πλευρά της φόρμας.

ΕΝΕΡΓΟΣ ΦΑΚΕΛΟΣ

Νοέμβριος 2016

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

Today: 12/12/2016

21:36

ΔΙΑΡΚΕΙΑ

Δεν έχει ανατεθεί

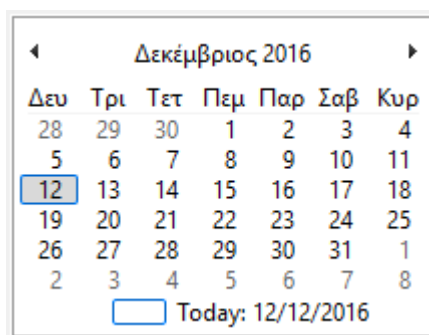
ΙΑΤΡΕΙΟ 1
ΙΑΤΡΕΙΟ 2
ΙΑΤΡΕΙΟ 3

ΠΡΟΣΘΗΚΗ

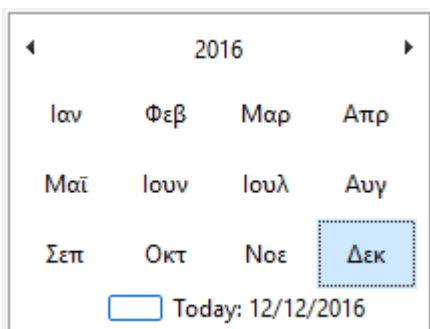
Εικόνα 12. Προσθήκη Ραντεβού

Στο πάνω μέρος της φόρμας υπάρχει το κουτί κειμένου με τίτλο «Ενεργός Φάκελος». Σε αυτό το κουτί εμφανίζεται το ονοματεπώνυμο του ασθενούς, ο οποίος έχει επιλεγεί σε προγενέστερο στάδιο, στη φόρμα «Ασθενείς». Αν δεν έχουμε επιλέξει κάποιον ενεργό ασθενή, λόγω του ότι από την εισαγωγική φόρμα επιλέξαμε άμεση μετάβαση στα ραντεβού, μπορούμε να το κάνουμε πατώντας το κουμπί «Επιλογή Ασθενούς».

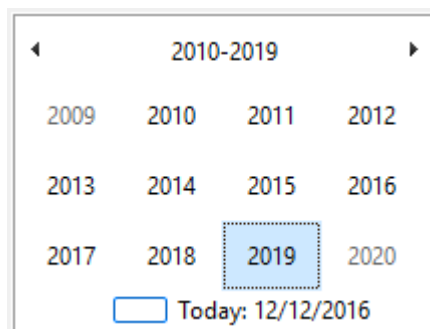
Έχοντας επιλέξει ενεργό ασθενή, στη συνέχεια επιλέγουμε την ημέρα που θέλουμε να δημιουργήσουμε το ραντεβού. Αυτό γίνεται επιλέγοντας ημέρα, μήνα και έτος στο προσφερόμενο ημερολόγιο. Πατώντας «Today» μεταφερόμαστε στην τρέχουσα ημερομηνία, ενώ οι μήνες αλλάζουν εύκολα με τα βελάκια δίπλα από την αναγραφή τους.



Εικόνα 13. Ημερολόγιο σε αρχική μορφή



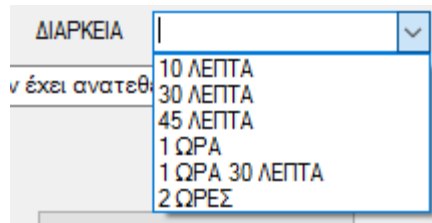
Εικόνα 15. Ημερολόγιο - επιλογή μήνα



Εικόνα 14. Ημερολόγιο - επιλογή έτους

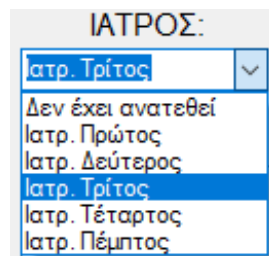
Πατώντας μια φορά πάνω στον τρέχοντα μήνα, το ημερολόγιο αλλάζει μορφή για εύκολη επιλογή του μήνα του τρέχοντος έτους. Πατώντας μια φορά επάνω στο τρέχον έτος, μεταβαίνουμε στην επιλογή έτους εντός της τρέχουσας δεκαετίας. Το αντίστοιχο συμβαίνει αν πατήσουμε επάνω στη δεκαετία.

Στη συνέχεια επιλέγουμε την επιθυμητή ώρα του ραντεβού από το κουτί επιλογής ώρας. Προχωρώντας, επιλέγουμε την υπολογιζόμενη διάρκεια του ραντεβού από την αναδυόμενη λίστα του κουτιού με τίτλο «Διάρκεια».



Εικόνα 16. Επιλογή διάρκειας ραντεβού

Έχοντας επιλέξει την επιθυμητή διάρκεια ραντεβού, μένει να επιλέξουμε τον οδοντίατρο που θα επιληφθεί της θεραπείας και το ιατρείο στο οποίο θα γίνει η θεραπεία. Η επιλογή του ιατρείου γίνεται από την αντίστοιχη λίστα, ενώ η επιλογή του οδοντίατρου γίνεται από την αναδυόμενη λίστα.



Εικόνα 17. Επιλογή οδοντίατρου

Τέλος, στο κάτω κεντρικό κομμάτι της φόρμας, βρίσκεται ένα πλαίσιο κειμένου όπου εμφανίζονται τα στοιχεία κάθε ραντεβού που επιλέγουμε σε κάποιον από τους τρεις πίνακες. Εμφανίζεται το ονοματεπώνυμο του ασθενούς, η ώρα ραντεβού, το ιατρείο που θα χρησιμοποιηθεί, ο θεράπων ιατρός και η πιθανή διάρκεια της συνεδρίας.

3.1.6. Φόρμα Γενικού Ιστορικού Υγείας

Στην παρούσα φόρμα γίνεται καταγραφή του απλού ιατρικού ιστορικού του ασθενούς. Στο πρώτο πεδίο κειμένου σημειώνουμε τυχόν αλλεργίες σε φάρμακα (πχ αντιβιοτικά, αναισθητικά). Στο δεύτερο γράφουμε υποκείμενες νόσους και μόνιμες

συστηματικές καταστάσεις που ενδεχομένως να επηρεάσουν τη θεραπεία. Στο τρίτο πεδίο μπορούμε να σημειώσουμε κάποιο σχόλιο, όπως για παράδειγμα τον επιβλέποντα ιατρό άλλης ειδικότητας, που παρακολουθεί την κατάσταση της υγείας του ασθενή (πχ παθολόγος, καρδιολόγος).

Εικόνα 18. Φόρμα Γενικού Ιστορικού Υγείας

Στη φόρμα αυτή είναι δυνατή η τήρηση μιας και μόνο εγγραφής, η οποία ακολουθεί τον ασθενή παντού και πάντα. Είναι απολύτως επιβεβλημένο, πριν από οποιαδήποτε οδοντιατρική πράξη, ο θεράπων οδοντίατρος να ενημερώνεται για τη γενική κατάσταση της υγείας του ασθενή.

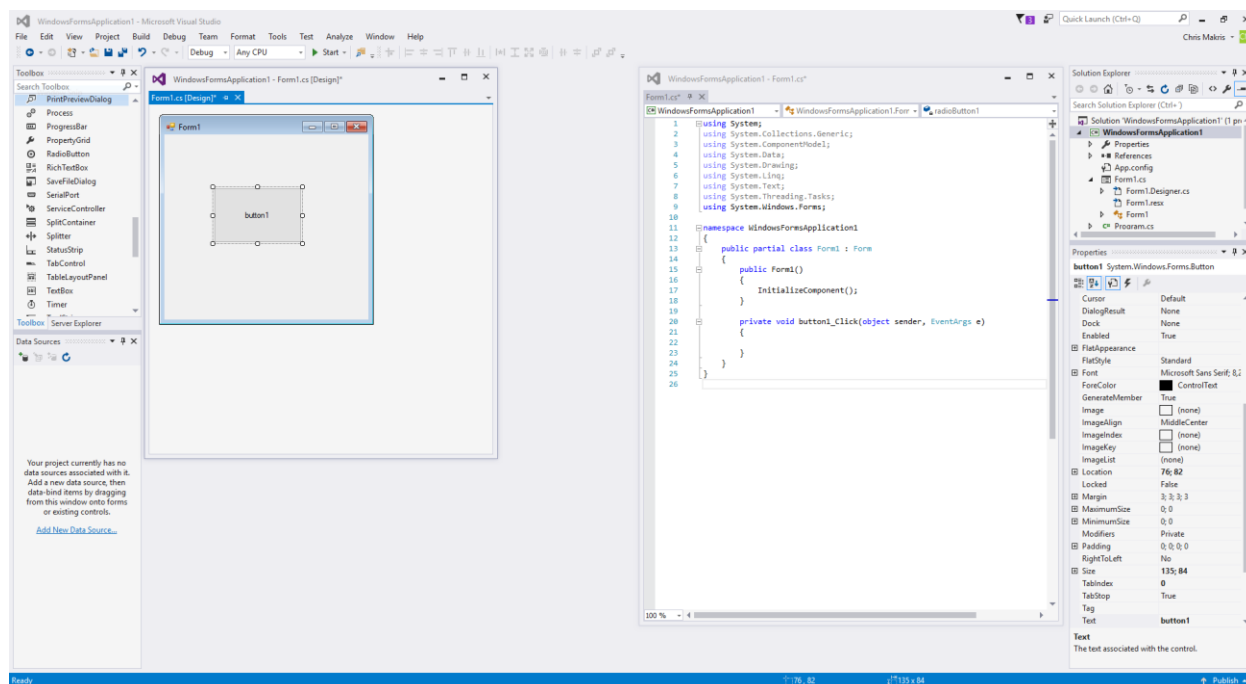
Στη φόρμα βρίσκονται επίσης δύο κουμπιά, τα οποία μας μεταφέρουν είτε πίσω στην επιλογή ασθενούς (ΑΣΘΕΝΕΙΣ), είτε απευθείας στο οδοντιατρικό ιστορικό του ασθενούς (ΟΔΟΝΤΙΑΤΡΙΚΟ ΙΣΤΟΡΙΚΟ). Με αυτόν τον τρόπο εφαρμόζεται η λογική αλληλουχία λήψης γενικού ιστορικού πρώτα, και στη συνέχεια οδοντιατρικού ιστορικού.

4. Εργαλεία ανάπτυξης της εφαρμογής

Για την ανάπτυξη της εφαρμογής έγινε χρήση μια σειράς εργαλείων και τεχνολογιών, με σημαντικότερα το περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio, διάφορες τεχνολογίες .Net και τη βάση δεδομένων Microsoft Access.

4.1. Ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών

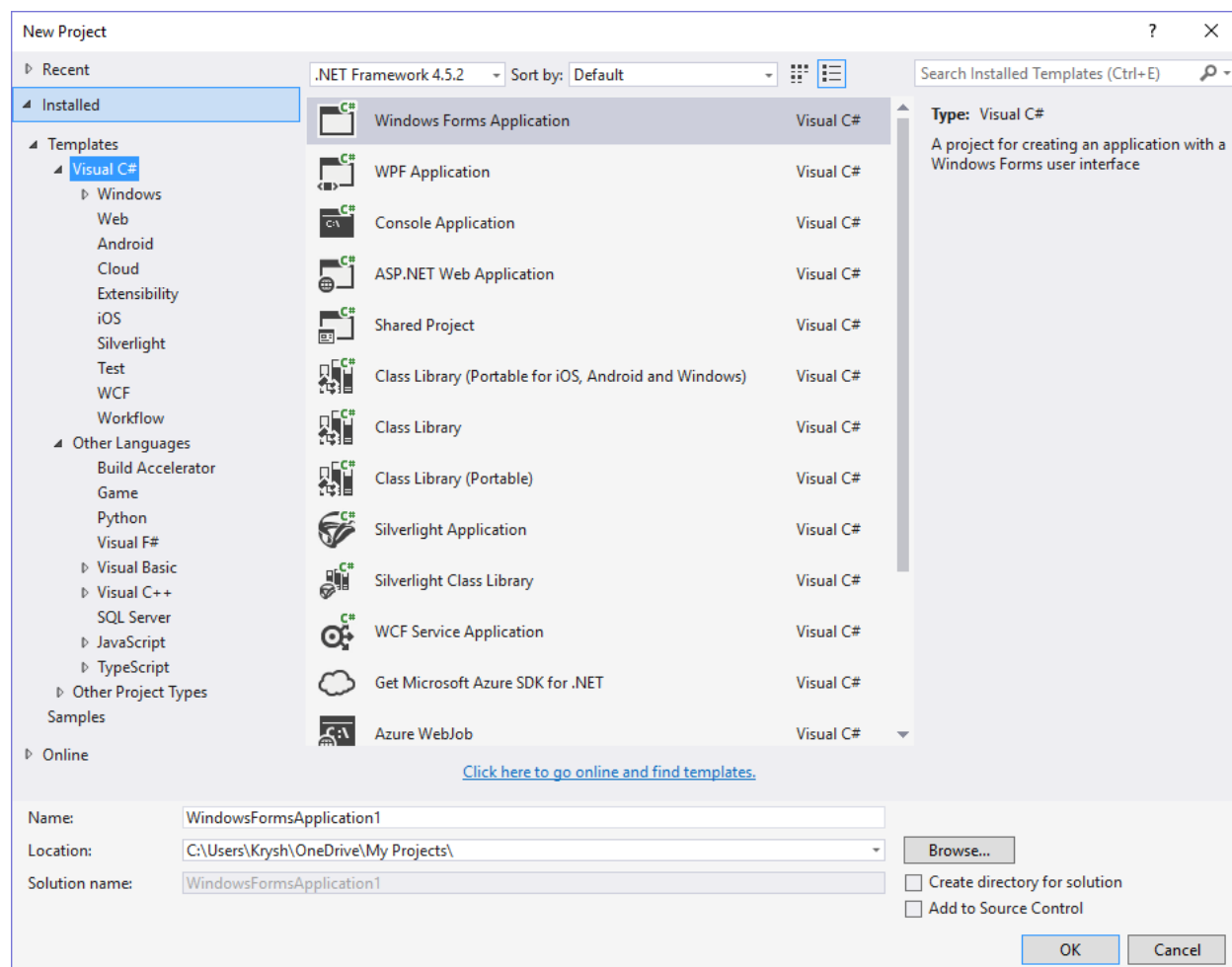
Έγινε χρήση του Microsoft Visual Studio IDE 2015, Community Edition. Το εν λόγω πρόγραμμα είναι ένα IDE – Integrated Development Environment – Ολοκληρωμένο Περιβάλλον Ανάπτυξης Εφαρμογών. Αποτελεί μια ολοκληρωμένη λύση για συγγραφή κώδικα, αποσφαλμάτωση, έλεγχο, συνεργασία στην ανάπτυξη, μεταγλώττιση και παραγωγή προγραμμάτων προς εγκατάσταση και εκτέλεση. (6)



Εικόνα 19. Το περιβάλλον ανάπτυξης του Visual Studio 2015 (6)

Μέσω του IDE μπορεί να γίνει ανάπτυξη εφαρμογών για Windows υπολογιστές, εφαρμογών Web, εφαρμογών για mobile συσκευές (κινητά τηλέφωνα, ταμπλέτες), παιχνιδιών, ειδικών εφαρμογών sql server και άλλων.

Οι υποστηριζόμενες γλώσσες προγραμματισμού είναι πολλές, με ενσωματωμένες τις C, C++ και C++/CLI (ως Visual C++), VB.NET (ως Visual Basic .NET), C# (ως Visual C#) και F#. Μπορεί να προστεθεί υποστήριξη τρίτων γλωσσών, όπως των Python, Ruby, Node.js και M, ενώ επίσης υποστηρίζονται οι XML/XSLT, HTML/XHTML, JavaScript και CSS.



Εικόνα 20. Υποστηριζόμενες γλώσσες και είδη project από το Visual Studio IDE

Ο επεξεργαστής κώδικα (code editor) του Visual Studio υποστηρίζει αυτόματη συμπλήρωση κώδικα IntelliSense και συντακτική σήμανση (Syntax highlighting). Η αυτόματη συμπλήρωση χρησιμοποιείται όχι μόνο για εντολές της γλώσσας, αλλά και για αναφορά σε υπάρχοντα τμήματα του κώδικα, όπως μεταβλητές, αντικείμενα, κλάσεις, στοιχεία διεπαφής κ.α.

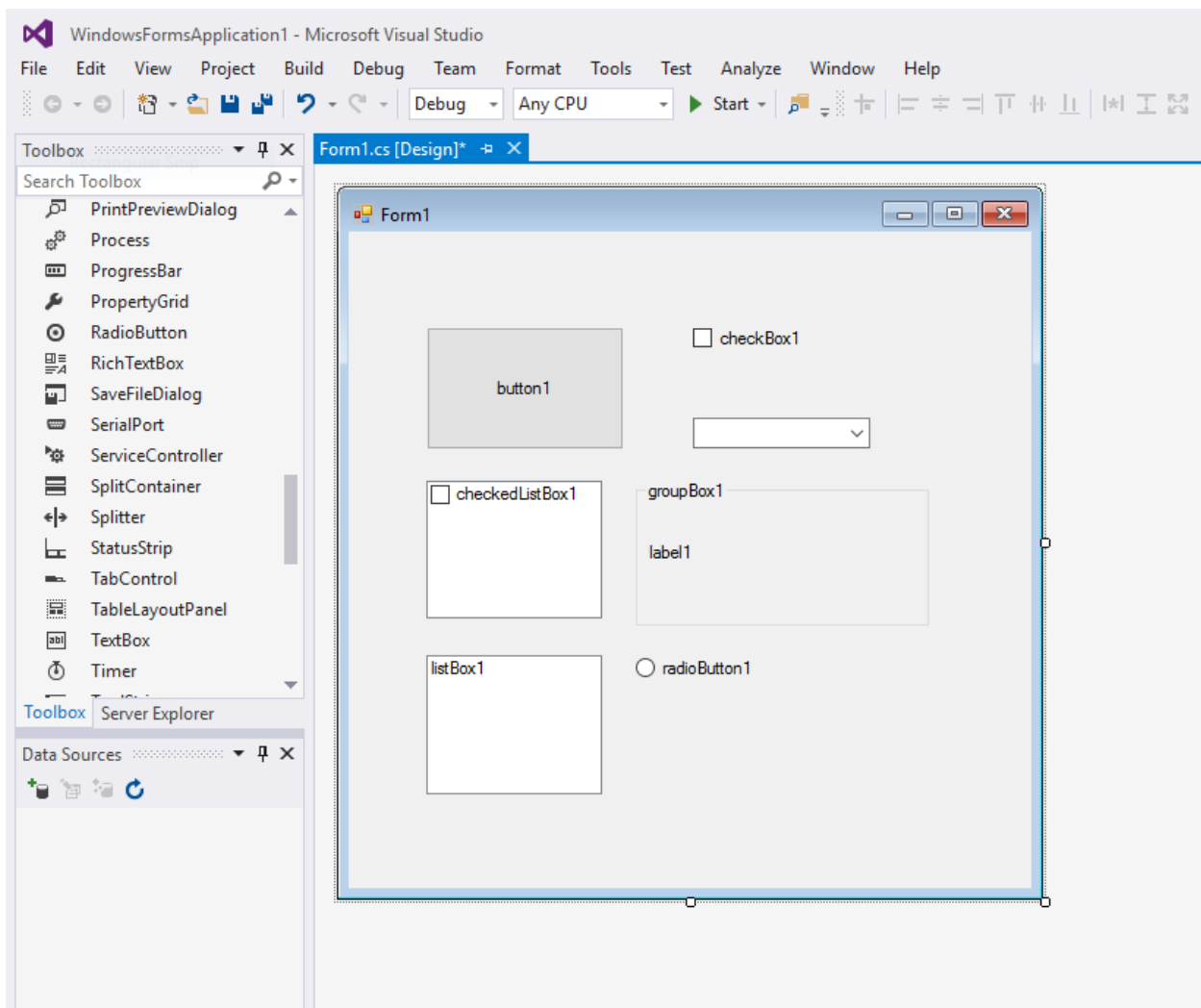
```

10
11 namespace DentalClinic
12 {
13     public partial class FormPatient : Form
14     {
15         public FormPatient()
16         {
17             InitializeComponent();
18         }
19
20         private void patientInfoBindingNavigatorSaveItem_Click(object sender, EventArgs e)
21         {
22             this.Validate();
23             this.patientInfoBindingSource.EndEdit();
24             this.tableAdapterManager.UpdateAll(this.dentalHRDataSet);
25             this.AcceptButton
26         }
27         private void AcceptButton
28             AccessibilityNotifyClients
29             AccessibilityObject
30             AccessibleDefaultActionDescription
31             // TO
32             this.AccessibleDescription
33             this.AccessibleName
34             AccessibleRole
35             Activate
36             Activated
37         {
38             //από εδώ θα πάρουμε το κείμενο του textbox ΕΠΙΘΕΤΟ την ώρα που πληκτρολογείται, έναν-έναν χαρακτήρα
39             try
40             {
41                 //μετατρέπουμε το κείμενο σε μορφή κατάλληλη για query, ένα string με το wildcard %
42                 string search = $"{textBoxNameSearch.Text}%";
43
44                 //γεμίζουμε το table μέσω της μεθόδου fillbyname που είναι query βάσει επιθέτου
45                 this.patientInfoTableAdapter.FillByName(this.dentalHRDataSet.PatientInfo, search);
46             }
47             catch (System.Exception ex)
48             {
49                 System.Windows.Forms.MessageBox.Show(ex.Message);
50             }
51         }
52     }
53 }

```

Εικόνα 21. Επεξεργαστής κώδικα με δυνατότητες αυτόματης συμπλήρωσης

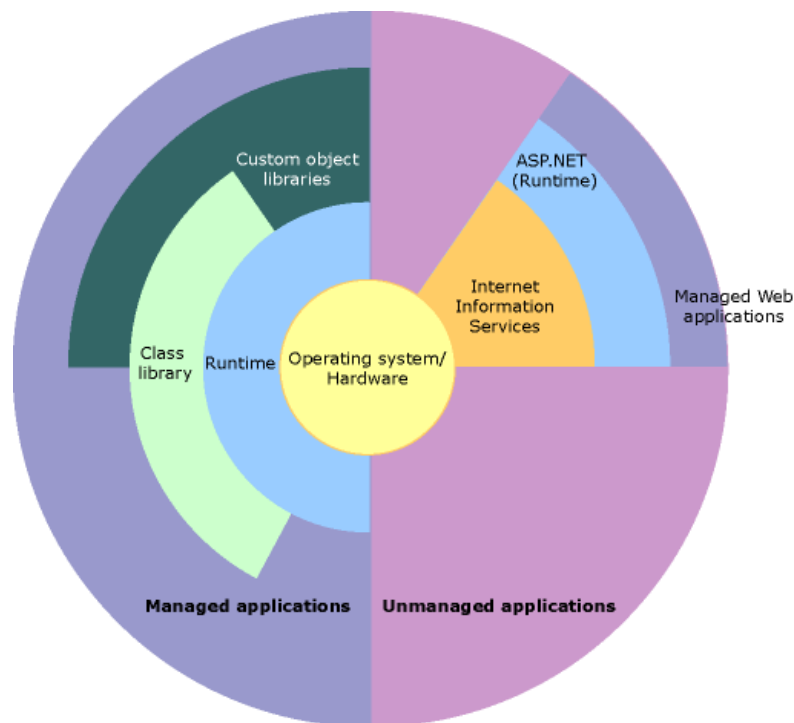
Ιδιαίτερο ενδιαφέρον παρουσιάζει ο Σχεδιαστής (Designer), η χρήση του οποίου διευκολύνει πολύ τη δημιουργία της διεπαφής χρήστη των εφαρμογών (GUI), κάνοντας χρήση «σύρε και άσε» τεχνικών για πλειάδα στοιχείων, όπως κουμπιά, κουτιά κειμένου κλπ. Αντικείμενα της διεπαφής που βασίζονται σε δυναμική σύνδεση δεδομένων με την εφαρμογή, αποκτούν με εύκολο τρόπο την εν λόγω σύνδεση στο Σχεδιαστή. Η λειτουργικότητα των αντικειμένων διεπαφής απλοποιείται με την άμεση παραγωγή του απαραίτητου κώδικα κάνοντας διπλό κλικ στο αντικείμενο και συμπληρώνοντας τον εξειδικευμένο κώδικα.



Εικόνα 22. Σχεδιαστής με δυνατότητες "Σύρε και άσε"

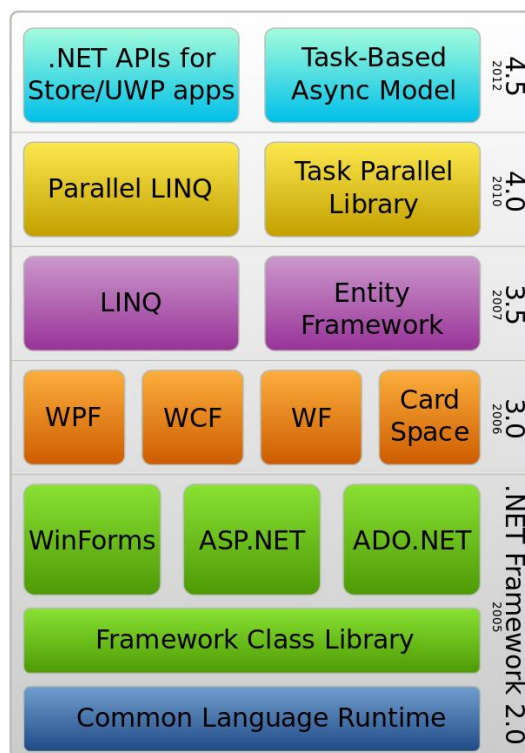
4.2. Τεχνολογίες .NET της Microsoft

Για την ανάπτυξη της εφαρμογής έγινε χρήση διαφόρων τεχνολογιών .net της Microsoft. Το «.NET Framework» είναι ένα πλαίσιο λογισμικού που επιτρέπει την εκτέλεση εφαρμογών γραμμένων σε διαφορετικές γλώσσες προγραμματισμού. Κάνει χρήση της Βιβλιοθήκης Κλάσεων Πλαισίου, μιας μεγάλης βιβλιοθήκης κλάσεων. Με χρήση της Βιβλιοθήκης Κλάσεων Πλαισίου (FCL – Framework Class Library), η ίδια εφαρμογή μπορεί να χρησιμοποιήσει κώδικα γραμμένο σε διαφορετικές γλώσσες προγραμματισμού, εφόσον αυτή είναι φτιαγμένη για εκτέλεση σε περιβάλλον .NET. Η εκτέλεση σε περιβάλλον .NET γίνεται σε επίπεδο λογισμικού (software) και όχι υλικού (hardware). Το περιβάλλον λογισμικού που αναλαμβάνει την εκτέλεση λέγεται Περιβάλλον Εκτέλεσης Κοινής Γλώσσας (Common Language Runtime) και είναι μια Εικονική Μηχανή για εφαρμογές (application virtual machine). Παρέχει στην εφαρμογή υπηρεσίες ασφάλειας, χειρισμού σφαλμάτων (exception handling) και διαχείρισης μνήμης. Ο κώδικας που προκύπτει με στόχο εκτέλεσης το ΠΕΚΓ, είναι ενδιάμεσος κώδικας (managed code) και όχι αυτόνομος (unmanaged) κώδικας. (7)



Εικόνα 23. .NET Framework in context (7)

Πάνω στους δυο βασικούς πυλώνες του .NET Πλαισίου, υλοποιείται πλήθος τεχνολογιών. Στην παρούσα εφαρμογή έγινε χρήση των τεχνολογιών WinForms και ADO.NET. (8)

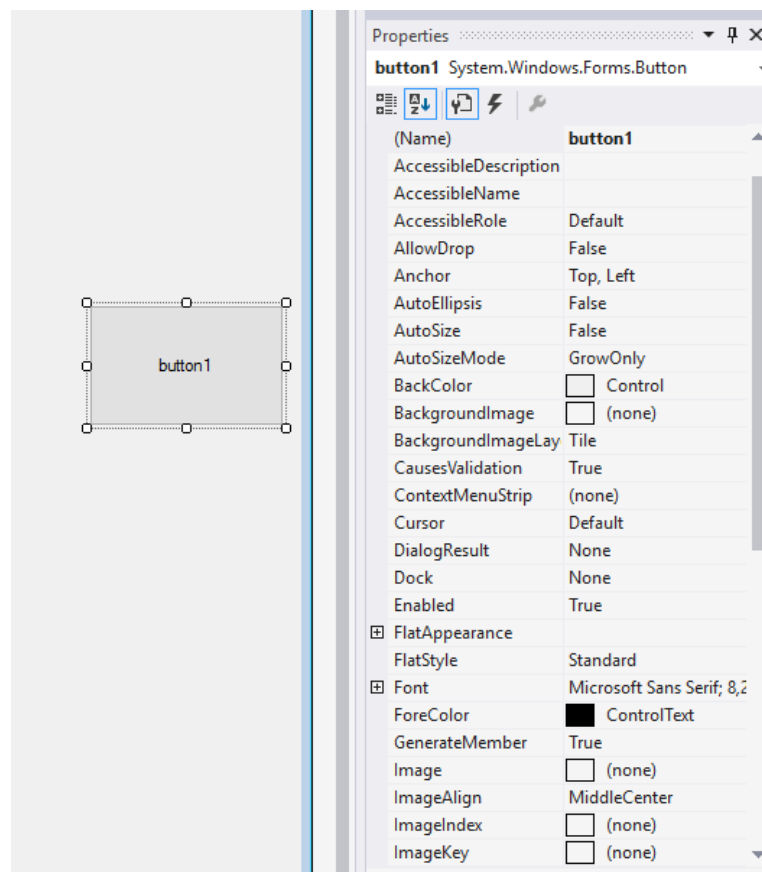


Εικόνα 24. The .NET Framework stack (8)

Η τεχνολογία WinForms της Microsoft έχει βασικό στόχο την ανάπτυξη κλασικών εφαρμογών για Windows. (8) Παρέχει όλα τα βασικά συστατικά μιας διεπαφής χρήστη (κουμπιά, πλαίσια διαλόγου, εμφάνιση γραφικών στοιχείων) και αυτοματοποιεί τη μεταφορά δεδομένων μεταξύ αυτών και της εφαρμογής. Με τη χρήση του Visual Studio IDE, έχουμε μεγάλες δυνατότητες σχεδιασμού της διεπαφής χρήστη της εφαρμογής μας, με έναν εύκολο, σταθερό και συγκροτημένο τρόπο.

Άλλη τεχνολογία που χρησιμοποιήθηκε, είναι η ADO.NET. Μέσω αυτής της τεχνολογίας είναι δυνατή η πρόσβαση σε σχεσιακές βάσεις δεδομένων και σε μη σχεσιακά αρχεία δεδομένων, με έναν ενιαίο και ενσωματωμένο τρόπο. Η πρόσβαση στα δεδομένα και τις λειτουργίες της βάσης δεδομένων είναι διάφανης για τον

προγραμματιστή, χωρίς να είναι αναγκαίο για αυτόν να χρησιμοποιήσει τη «μητρική» γλώσσα κάθε συστήματος. (9)



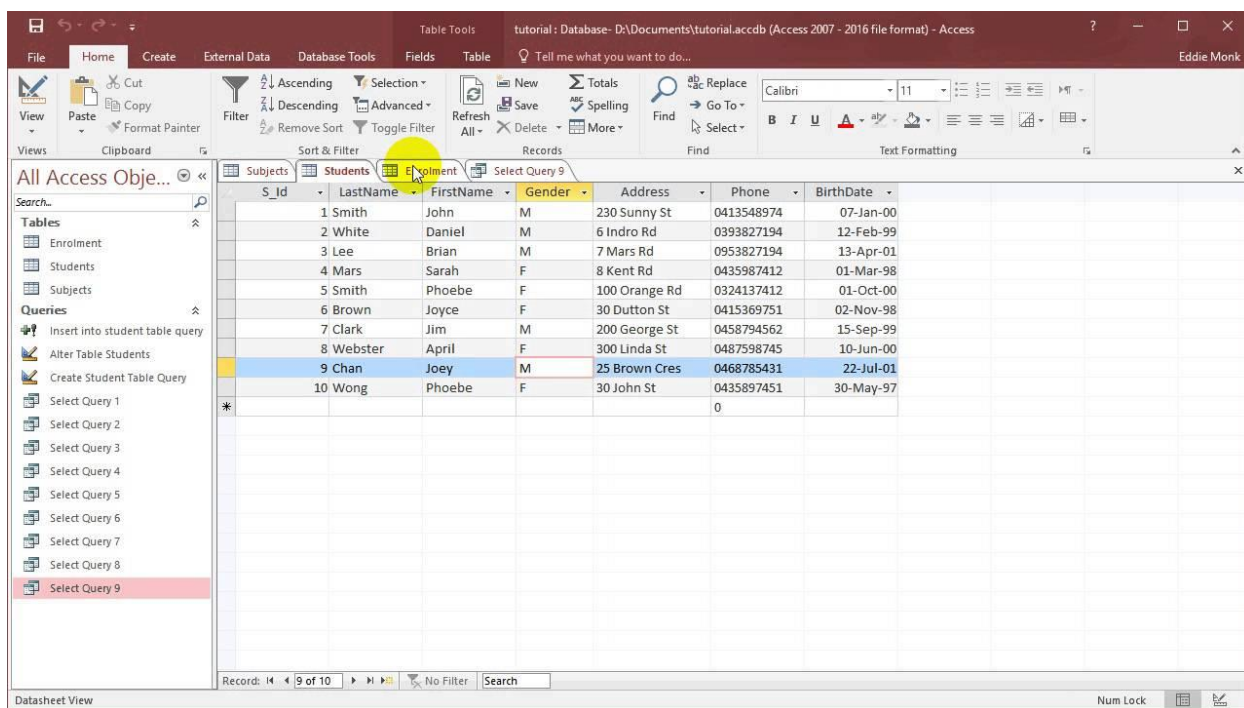
Εικόνα 25. Παράδειγμα WinForms - Κουμπί "button1" και οι ιδιότητές του

4.3. Σύστημα Διαχείρισης Βάσης Δεδομένων Microsoft Access

Η Microsoft Access είναι ένα σύστημα διαχείρισης βάσης δεδομένων που αναπτύχθηκε από τη Microsoft και εισήχθη στην κυκλοφορία το 1992. (10) Κάνει χρήση της Microsoft Jet Database Engine ώστε να παρέχει λειτουργίες σχεσιακής βάσης δεδομένων και συνδυάζει το γραφικό περιβάλλον διεπαφής με ισχυρά προγραμματιστικά εργαλεία για την ανάπτυξη εφαρμογών βάσεων δεδομένων. Είναι γραμμένη σε γλώσσα C# και αποτελεί μέρος της σουίτας εφαρμογών Microsoft Office. Η

μηχανή Jet ρυθμίζει της λειτουργίες πολυχρηστικότητας και κλειδώματος εγγραφών ώστε η παράλληλη χρήση και επεξεργασία της βάσης δεδομένων από πολλούς χρήστες να είναι απρόσκοπτη και απροβλημάτιστη. Προσφέρει ταυτόχρονη χρήση σε ως 255 χρήστες και σε βάσεις δεδομένων ως 2GB.

Πέρα από τη λειτουργία της βάσης δεδομένων, η Access έχει ενσωματωμένα εργαλεία για τη δημιουργία φορμών, με σκοπό την εύκολη εμφάνιση και επεξεργασία των δεδομένων μέσω της γραφικής διεπαφής χρήστη. Επίσης υποστηρίζει δημιουργία και εκτύπωση αναφορών, αυτοματοποίηση με μακροεντολές και προγραμματιστική επέκταση μέσω της γλώσσας Visual Basic for Applications (VBA).



Εικόνα 26. Η Microsoft Access 2016 (10)

5. Ανάπτυξη της εφαρμογής Πληροφοριακού Συστήματος Οδοντιατρικής Κλινικής

Η υλοποίηση της εφαρμογής έγινε σε δύο στάδια. Στο πρώτο δημιουργήθηκε η βάση δεδομένων που αποτελεί το θεμέλιο της εφαρμογής, στη Microsoft Access. Στο δεύτερο στάδιο, πάνω στην υπάρχουσα βάση δεδομένων υλοποιήθηκε η διεπαφή χρήστη με την κατασκευή της εφαρμογής στο ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio. Στη συνέχεια θα γίνει λεπτομερής περιγραφή της ανάπτυξης της εφαρμογής.

5.1. Ανάπτυξη Βάσης Δεδομένων

Η βάση δεδομένων της εφαρμογής δημιουργήθηκε στην Microsoft Access 2016. Το είδος του αρχείου βάσης δεδομένων που επιλέχθηκε είναι το .mdb, για λόγους συμβατότητας και καλύτερης υποστήριξης από το Visual Studio. Η βάση δεδομένων ονομάστηκε “DentalHR.mdb”. Μετά τη συγκέντρωση των απαιτήσεων όσον αφορά την πληροφορία προς αποθήκευση, δημιουργήθηκαν αντίστοιχοι πίνακες στη βάση δεδομένων. Οι εν λόγω πίνακες έχουν ως εξής:

Πίνακας «PatientInfo», όπου αποθηκεύονται τα γενικά στοιχεία επαφής των ασθενών.

Πίνακας «GeneralHealth», όπου αποθηκεύεται το ιστορικό γενικής υγείας κάθε ασθενή.

Πίνακας «Diagnosis», όπου καταγράφεται η παρούσα κατάσταση της οδοντιατρικής υγείας του ασθενή.

Πίνακας «Treatments», όπου καταγράφονται οι οδοντιατρικές πράξεις στις οποίες υποβάλλεται ο ασθενής.

Πίνακας «Multimedia», όπου αποθηκεύονται τα πολυμεσικά αρχεία εξετάσεων των ασθενών.

Πίνακας «Doctors», όπου αποθηκεύονται τα στοιχεία των οδοντιάτρων που προσφέρουν υπηρεσίες στην κλινική.

Πίνακας «Appointments», όπου τηρούνται τα στοιχεία των ραντεβού των ασθενών.

Θα προχωρήσουμε στην επιμέρους ανάλυση των ανωτέρω πινάκων, σε επίπεδο πεδίου.

5.1.1. Πίνακας «PatientInfo»

Ο πίνακας PatientInfo περιλαμβάνει τα πεδία δημογραφικών στοιχείων του ασθενή. Πιο συγκεκριμένα αποτελείται από τα εξής πεδία:

Pat_ID: Είναι πεδίο με αυτόματη αρίθμηση και αποτελεί το πρωτεύον κλειδί αυτού του πίνακα. Ουσιαστικά λαμβάνει θέση αύξοντα αριθμού ασθενούς.

Pat_Doctor_Key: Σε αυτό το πεδίο αποθηκεύεται δευτερεύον κλειδί από τον πίνακα Doctors, το οποίο έχει την έννοια του επιβλέποντα οδοντιάτρου για κάθε ασθενή. Με αυτόν τον τρόπο είναι δυνατή η άμεση γνώση του οδοντιάτρου στον οποίο έχει ανατεθεί η γενική ευθύνη της οδοντιατρικής υγείας του ασθενή. Δεν είναι απαραίτητο να αποτελεί τον μοναδικό και μόνιμο οδοντίατρο που θα παρέχει υπηρεσίες οδοντιατρικής φροντίδας στον συγκεκριμένο ασθενή, γεγονός το οποίο καταγράφεται σε διαφορετικό πίνακα.

Pat_surname: Πεδίο κειμένου όπου καταγράφεται το επίθετο του ασθενούς.

Pat_name: Πεδίο κειμένου όπου καταγράφεται το όνομα του ασθενούς.

Pat_amka: Αριθμητικό πεδίο περιορισμένων θέσεων (11) όπου καταγράφεται ο Αριθμός Μητρώου Κοινωνικής Ασφάλισης – ΑΜΚΑ του ασθενούς.

Pat_guardian: Πεδίο κειμένου όπου καταγράφεται το όνομα του κηδεμόνα του ασθενούς, στην περίπτωση που ο ασθενής είναι ανήλικος. Στην περίπτωση ενηλίκου μπορεί να καταγραφεί το όνομα πατρός, μητρός ή συζύγου.

Pat_birth: Πεδίο ημερομηνίας/ώρα με μορφή σύντομης ημερομηνίας, όπου καταγράφεται η ημερομηνία γέννησης του ασθενούς.

Pat_tele: Αριθμητικό πεδίο όπου καταγράφεται ο αριθμός τηλεφώνου του ασθενούς.

Pat_address: Πεδίο σύντομου κειμένου όπου καταγράφεται η διεύθυνση κατοικίας του ασθενή.

The screenshot shows the 'PatientInfo' form in the DentalHR application. The form is divided into two main sections: a table of patient information fields and a section for patient details.

Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή (προαιρετικά)
Pat_ID	Αυτόματη Αρίθμηση	
Pat_Doctor_Key	Αριθμός	
Pat_surname	Σύντομο κείμενο	
Pat_name	Σύντομο κείμενο	
Pat_amka	Σύντομο κείμενο	
Pat_guardian	Σύντομο κείμενο	
Pat_birth	Ημερομηνία/ώρα	
Pat_tele	Σύντομο κείμενο	
Pat_address	Σύντομο κείμενο	

Ιδιότητες πεδίου

Γενικές	Εμφάνιση
Μέγεθος πεδίου	Μεγάλος ακέραιος
Νέες τιμές	Θηματοκ
Μορφή	
Λεζάντα	
Με ευρετήριο	Ναι (δεν επιτρέπονται διπλότυπα)
Στοιχείο κειμένου	Γενική

Τα ονόματα πεδίων μπορούν να έχουν έως 64 χαρακτήρες (με τα κενά). Για βοήθεια σχετικά με τα ονόματα πεδίων, πατήστε F1.

Εικόνα 27. Ο πίνακας PatientInfo

5.1.2. Πίνακας «GeneralHealth»

Στον πίνακα GeneralHealth αποθηκεύεται το ιστορικό γενικής υγείας του ασθενή. Πριν τη διενέργεια οποιασδήποτε ιατρικής ή οδοντιατρικής πράξης, είναι απαραίτητο να ληφθεί από τον επαγγελματία υγείας το ιστορικό υγείας του ασθενούς. Σε αυτό θα καταγραφούν τυχόν προϋπάρχοντα χρόνια προβλήματα υγείας, χρόνιες παθήσεις, πρόσφατες αλλά σημαντικές παθήσεις, ιστορικό χειρουργικών επεμβάσεων, επιβλαβείς έξεις όπως κάπνισμα ή κατανάλωση αλκοόλ και γνωστές αλλεργίες σε φαρμακευτικά σκευάσματα.

Τα πεδία που περιέχονται σε αυτόν τον πίνακα έχουν ως εξής:

Gen_Pat_Key: Αποτελεί το πρωτεύον κλειδί του πίνακα, το οποίο ισοδυναμεί με το αντίστοιχο κλειδί του πίνακα PatientInfo. Στην ουσία ο πίνακας GeneralHealth αποτελεί συνέχεια του PatientInfo, αφού συνδέονται με σχέση «Ένα προς Ένα». Ο διαχωρισμός της συνέχειας της πληροφορίας σε δύο πίνακες έγινε για λόγους ιατρικού απορρήτου. Τα στοιχεία του γενικού ιατρικού ιστορικού που τηρούνται στον αντίστοιχο πίνακα, αποτελούν μέρος ιατρικής πληροφορίας που προστατεύεται από το ιατρικό απόρρητο. Με το χωρισμό σε δύο πίνακες είναι δυνατή η τμηματική και βάσει εξουσιοδότησης, πρόσβαση στην ιατρική πληροφορία.

Gen_Doctor_Key: Αυτό το πεδίο αποτελεί δευτερεύον κλειδί του πίνακα Doctors. Με αυτόν τον τρόπο αποθηκεύεται ο ιατρός που έλαβε το ιατρικό ιστορικό του ασθενούς.

Gen_Allergies: Πεδίο κειμένου στο οποίο καταγράφονται γνωστές αλλεργίες σε φαρμακευτικά σκευάσματα.

Gen_Conditions: Πεδίο κειμένου στο οποίο καταγράφονται χρόνιες παθήσεις, συγγενείς ελλείψεις ενζύμων και λοιπά ιατρικά στοιχεία που αφορούν τη γενική υγεία του ασθενούς.

Gen_Comments: Πεδίο κειμένου στο οποίο μπορούν να καταγραφούν λοιπά σημαντικά στοιχεία της γενικής κατάστασης του ασθενούς, όπως ιστορικό χειρουργικών επεμβάσεων, έτη και συχνότητα καπνίσματος και κατανάλωσης αλκοόλ, τύπος εργασίας

και διαβίωσης, στοιχεία επικοινωνίας με λοιπούς επιβλέποντες της υγείας του ασθενούς και οτιδήποτε άλλο κρίνεται χρήσιμο από τον λήπτη του ιστορικού.

Ονομα πεδίου	Τύπος δεδομένων	Περιγραφή (προαιρετικά)
Gen_Pat_Key	Αριθμός	
Gen_Doctor_Key	Αριθμός	
Gen_Allergies	Σύντομο κείμενο	
Gen_Conditions	Σύντομο κείμενο	
Gen_Comments	Μεγάλο κείμενο	

Γενικά	Εμφάνιση
Μέγεθος πεδίου	Μεγάλος ακέραιος
Μορφή	Αυτόματα
Δεκαδικές θέσεις	
Μάσκα εισαγωγής	
Αξία	
Προσπελεγμένη τιμή	0
Κανόνας επικύρωσης	
Κείμενο επικύρωσης	
Απαιτείται	Ναι
Με ευρετήριο	Ναι (δεν επιτρέπονται διπλότυπα)
Στοιχείο κειμένου	Γενική

Εικόνα 28. Ο πίνακας GeneralHealth

5.1.3. Πίνακας Diagnosis

Σε αυτόν τον πίνακα μπορεί να καταγραφεί η παρούσα κατάσταση του στόματος του ασθενούς πριν από οποιαδήποτε οδοντιατρική πράξη. Η λεπτομερής καταγραφή της στοματικής κατάστασης σε συνδυασμό με τις λοιπές οδοντιατρικές εξετάσεις (ακτινογραφίες, αξονικές, εκμαγεία) είναι αναγκαία για τη σωστή δημιουργία του σχεδίου θεραπείας του ασθενή, ιδίως όταν αυτό θα συμπεριλάβει εκτεταμένες αποκαταστάσεις. Τα πεδία που περιέχονται στον πίνακα έχουν ως εξής:

Diag_ID: Αποτελεί το πρωτεύον κλειδί του πίνακα. Αντιστοιχεί στον αύξοντα αριθμό των εξετάσεων ασθενών.

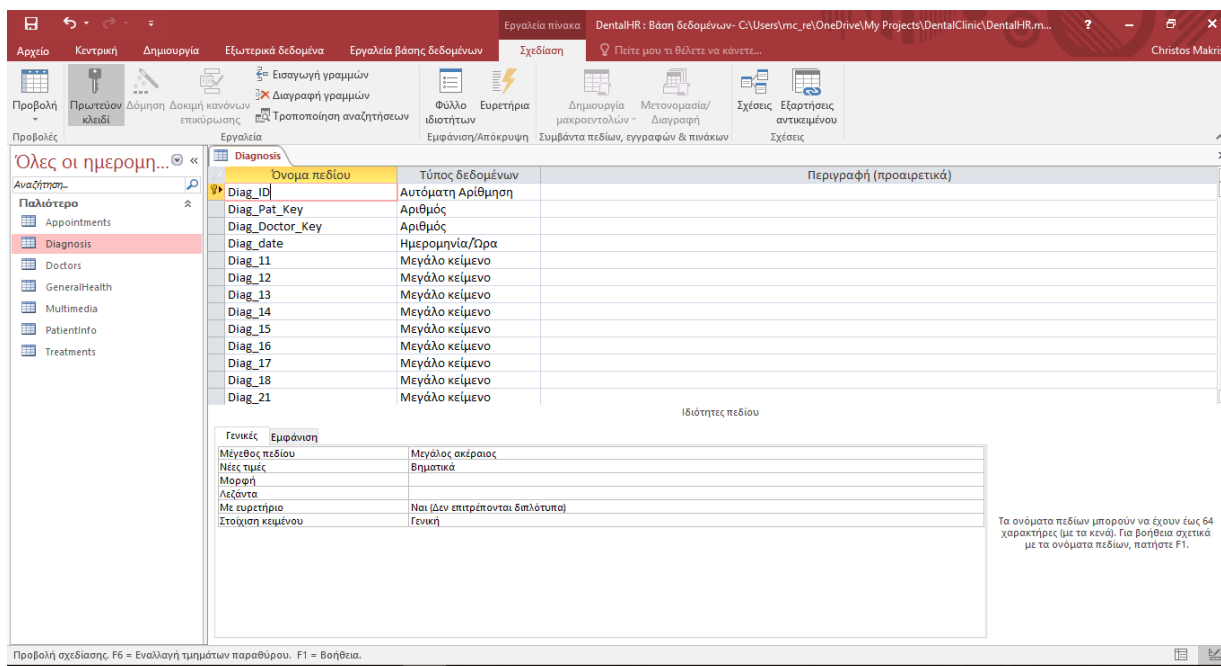
Diag_Pat_Key: Το πεδίο αυτό είναι δευτερεύον κλειδί που προέρχεται από τον πίνακα PatientInfo. Αντιστοιχίζει έτσι έναν ασθενή σε κάθε κλινική εξέταση. Με αυτόν τον τρόπο είναι δυνατό να ληφθούν ανεξάρτητες κλινικές εξετάσεις ανά τακτά χρονικά διαστήματα ώστε να παρακολουθείται η στοματική υγεία του ασθενή.

Diag_Doctor_Key: Το πεδίο αυτό είναι δευτερεύον κλειδί που προέρχεται από τον πίνακα Doctors και με τη χρήση του δηλώνεται ο συγκεκριμένος οδοντίατρος που επιτέλεσε την κλινική εξέταση του ασθενούς.

Diag_date: Πεδίο ημερομηνίας/ώρας στο οποίο καταγράφεται η ημερομηνία λήψης της κλινικής εξέτασης.

Diag_XX: Όπου XX μπαίνει ο διψήφιος αριθμός που αντιστοιχεί σε κάθε δόντι του φραγμού. Συνολικά υπάρχουν 32 πεδία κειμένου, ένα για κάθε μόνιμο δόντι του ασθενούς. Σε αυτά μπορούν να καταγραφούν όλες οι πληροφορίες που ενδιαφέρουν τον οδοντίατρο. Κατά τη λήψη κλινικού ιστορικού, οι πληροφορίες αυτές είναι κατά κανόνα παθολογικής φύσεως.

Diag_notes: Πεδίο κειμένου στο οποίο μπορούν να καταγραφούν όλα τα στοιχεία που δεν εντοπίζονται αυστηρά σε ένα και μόνο δόντι, ή που αφορούν γενικότερα τη στοματική κοιλότητα όπως χείλη, γλώσσα, παρειές, υπερώα και φάρυγγα. Μπορούν ακόμα να καταγραφούν ακτινογραφικά ευρήματα, πληροφορίες για τη σύγκλειση και άλλες σχετικές πληροφορίες.



Εικόνα 29. Ο πίνακας Diagnosis

5.1.4. Πίνακας Treatments

Σε αυτόν τον πίνακα καταγράφονται όλες οι οδοντιατρικές πράξεις που εκτελούνται στο στόμα του ασθενή. Η μορφή του είναι παρόμοια με αυτή του πίνακα Diagnosis. Με το σαφή διαχωρισμό των κλινικών εξετάσεων (Diagnosis) και των κλινικών πράξεων (Treatments) είναι εύκολη η εμφάνιση όλου του ιστορικού πράξεων στον ασθενή, και παράλληλα σύγκριση με την πιθανή παθολογική εικόνα του στόματος πριν ή ανάμεσα στις επεμβάσεις. Τα πεδία του πίνακα έχουν ως εξής:

Treat_ID: Αποτελεί το πρωτεύον κλειδί του πίνακα. Αντιστοιχεί στον αύξοντα αριθμό των εξετάσεων ασθενών.

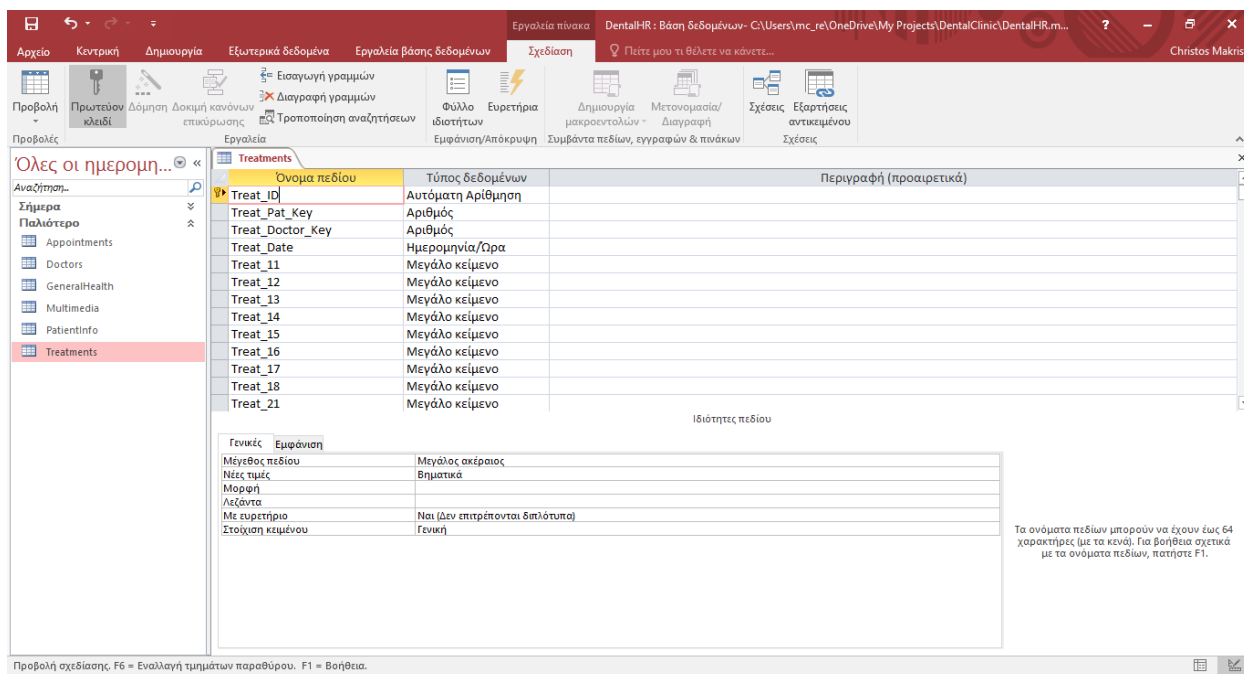
Treat_Pat_Key: Το πεδίο αυτό είναι δευτερεύον κλειδί που προέρχεται από τον πίνακα PatientInfo. Αντιστοιχίζει έτσι έναν ασθενή σε κάθε οδοντιατρική θεραπευτική πράξη ανά συνεδρία.

Treat_Doctor_Key: Το πεδίο αυτό είναι δευτερεύον κλειδί που προέρχεται από τον πίνακα Doctors και με τη χρήση του δηλώνεται ο συγκεκριμένος οδοντίατρος που επιτέλεσε τη συγκεκριμένη οδοντιατρική πράξη στο στόμα του ασθενούς, στη συγκεκριμένη συνεδρία. Παρόλο που ο ασθενής έχει επιβλέποντα οδοντίατρο (όπως ορίστηκε στον πίνακα PatientInfo, δεν υπάρχει δέσμευση στο ποιός οδοντίατρος θα προσφέρει υπηρεσίες υγείας στον ασθενή. Κάθε συνεδρία είναι αυτόνομη και διακριτή, και μπορεί να γίνεται από διαφορετικό οδοντίατρο κάθε φορά.

Diag_date: Πεδίο ημερομηνίας/ώρας στο οποίο καταγράφεται η ημερομηνία της οδοντιατρικής συνεδρίας.

Diag_XX: Όπου XX μπαίνει ο διψήφιος αριθμός που αντιστοιχεί σε κάθε δόντι του φραγμού. Συνολικά υπάρχουν 32 πεδία κειμένου, ένα για κάθε μόνιμο δόντι του ασθενούς. Σε αυτά μπορούν να καταγραφούν όλες οι πληροφορίες που σχετίζονται με την θεραπεία στην οποία υποβάλλεται ο ασθενής και εντοπίζονται συγκεκριμένα και διακριτά σε κάθε δόντι.

Diag_notes: Πεδίο κειμένου στο οποίο μπορούν να καταγραφούν όλα τα στοιχεία που δεν εντοπίζονται αυστηρά σε ένα και μόνο δόντι, ή που αφορούν γενικότερα τη στοματική κοιλότητα όπως χείλη, γλώσσα, παρειές, υπερώα και φάρυγγα. Εδώ θα γίνει καταγραφή εκταταμένων θεραπειών και αποκαταστάσεων και παρακολούθηση της πορείας τους. Μπορεί επίσης να καταγραφούν τυχόν συμβάματα κατά τη διάρκεια της θεραπείας, παρενέργειες αυτής, καταγραφή της πορείας ίασης και καταγραφή των οδηγιών φαρμακευτικής αγωγής και στοματικής υγιεινής που δώθηκαν στον ασθενή.



Εικόνα 30. Ο πίνακας Treatments

5.1.5. Πίνακας Multimedia

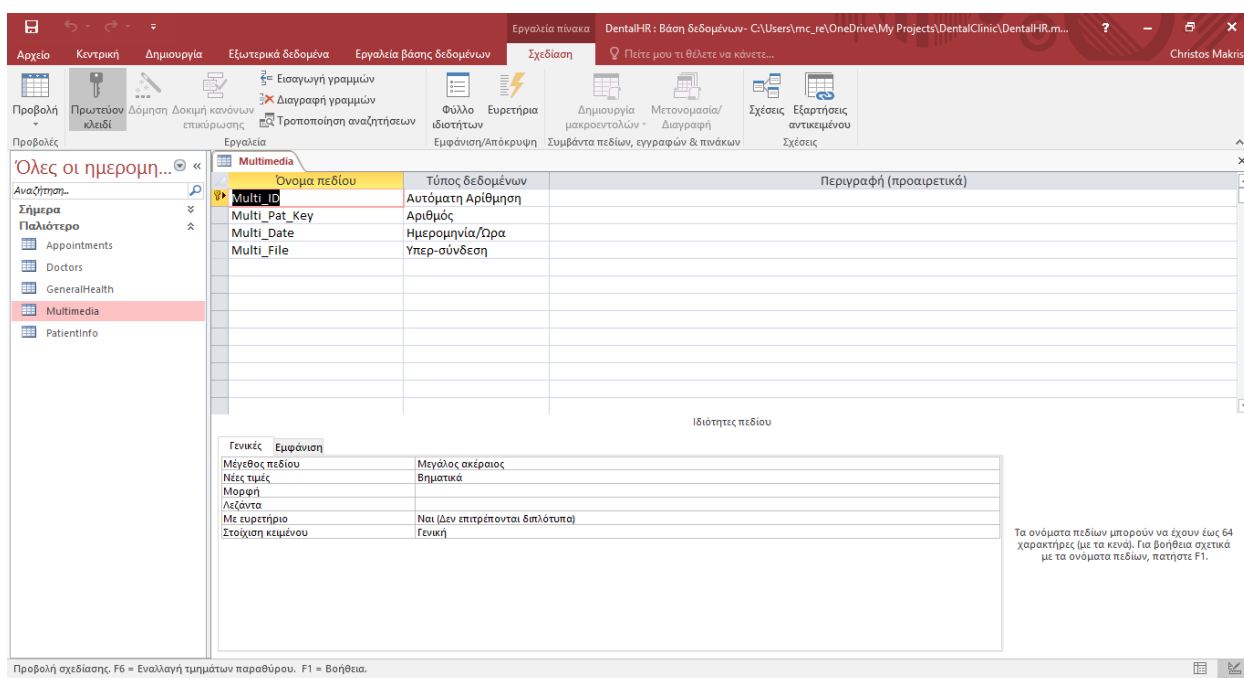
Στον πίνακα Multimedia τηρούνται σύνδεσμοι προς ψηφιακά αρχεία εξετάσεων του ασθενούς. Στο σύγχρονο οδοντιατρείο υφίσταται ένα πλήθος εξετάσεων που παράγουν ψηφιακά αρχεία, σε συνδυασμό με κλασικά έγγραφα από τρίτες πηγές που χρειάζονται σάρωση και αποθήκευση. Τα συνηθέστερα είδη ψηφιακών αρχείων είναι φωτογραφίες του προσώπου και του στόματος του ασθενούς, αρχεία ήχου – ομιλίας του ασθενούς για τη διάγνωση οδοντικών προβλημάτων άρθρωσης, βίντεο με ομιλία και συνηθείς κινήσεις μάσησης και ομιλίας για τη διάγνωση ορθογναθικών προβλημάτων, ψηφιακές ακτινογραφίες ενδοστοματικές και εξωστοματικές, ψηφιακές καταγραφές σύγκλεισης και κινήσεων κάτω γνάθου, αξονικές ακτινογραφίες με τα αντίστοιχα αναπλασμένα 3D μοντέλα, τρισδιάστατες σαρώσεις εκμαγείων του ασθενούς, αρχεία χρωματοληψίας των δοντιών και άλλα. Τα πεδία που χρησιμοποιούνται σε αυτόν τον πίνακα έχουν ως εξής:

Multi_ID: Το πεδίο αυτό αποτελεί τον αύξοντα αριθμό αρχείο πολυμέσων

Multi_Pat_Key: Δευτερεύον κλειδί που προέρχεται από τον πίνακα PatientInfo και συσχετίζει τον κάθε ασθενή με τα αντίστοιχα αρχεία πολυμέσων κυριότητάς του.

Multi_Date: Πεδίο τύπου ημερομηνίας/ώρας με μορφή σύντομης ημερομηνίας που καταγράφει την ώρα προσθήκης του αρχείου πολυμέσων στη βάση δεδομένων.

Multi_File: Πεδίο τύπου υπερσύνδεσης που συνδέει την εγγραφή του πίνακα multimedia με το ψηφιακό αρχείο που βρίσκεται αποθηκευμένο σε ανεξάρτητη τοποθεσία.



Εικόνα 31. Ο πίνακας Multimedia

5.1.6. Πίνακας Doctors

Σε αυτόν τον πίνακα τηρείται κατάλογος με τους οδοντιάτρους που παρέχουν υπηρεσίες υγείας στη συγκεκριμένη κλινική. Σε αυτόν τον πίνακα δε γίνεται ενδεδειγμένη αποθήκευση όλων των στοιχείων των οδοντιάτρων, αλλά μόνο όσων είναι ουσιαστικά για την καθημερινή εύρυθμη λειτουργία της κλινικής, τον προγραμματισμό των

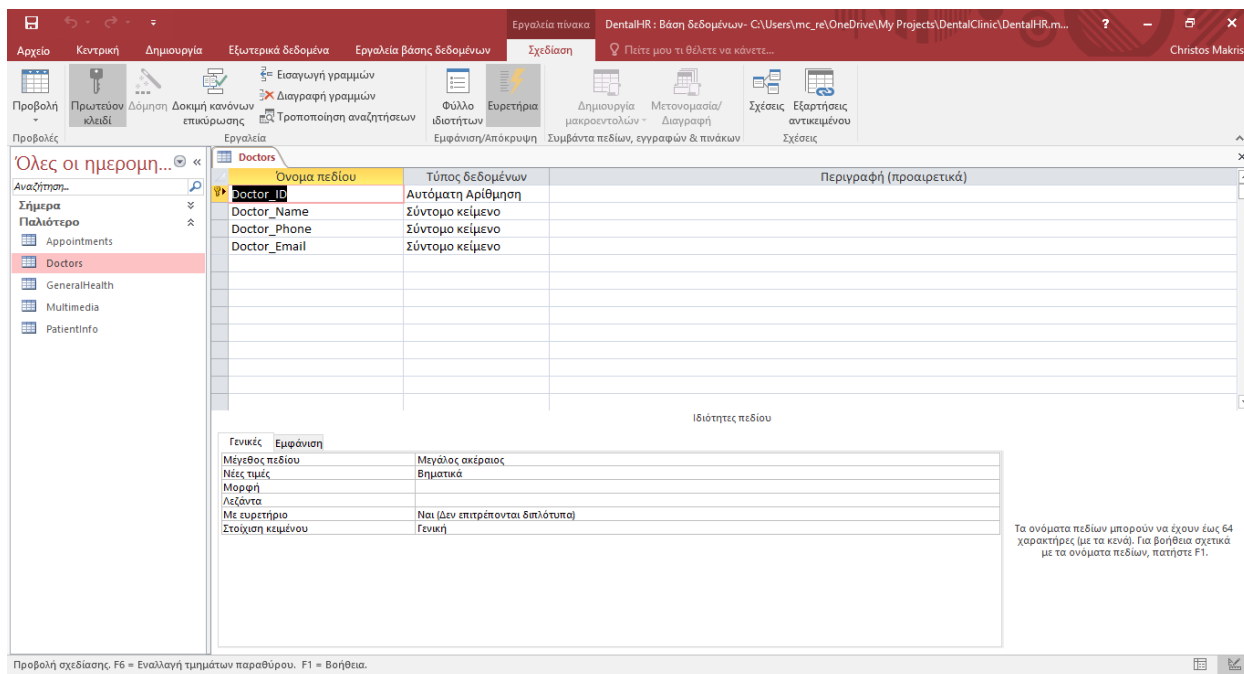
περιστατικών και την καθημερινή κλινική πράξη. Το πρωτεύον κλειδί αυτού του πίνακα χρησιμοποιείται συχνά ως δευτερεύον, όποτε πρέπει να γίνει εξακρίβωση του επιβλέποντα οδοντιάτρου. Τα πεδία του πίνακα έχουν ως εξής:

Doctor_ID: Το πρωτεύον κλειδί του πίνακα, με αυτόματη αρίθμηση. Έχει θέση αύξοντα αριθμού μητρώου οδοντιάτρου.

Doctor_Name: Πεδίο κειμένου όπου αποθηκεύεται το ονοματεπώνυμο του οδοντιάτρου.

Doctor_Phone: Πεδίο κειμένου όπου αποθηκεύεται το τηλέφωνο του οδοντιάτρου.

Doctor_Email: Πεδίο κειμένου όπου αποθηκεύεται το email του οδοντιάτρου. Το email είναι απαραίτητο στην καθημερινή κλινική πράξη, λόγω της πολύ συχνής αποστολής εξετάσεων και ψηφιακών ακτινογραφιών για έλεγχο, εκτός του προγράμματος κλινικών πράξεων.



Εικόνα 32. Ο πίνακας Doctors

5.1.7. Πίνακας Appointments

Ο πίνακας Appointments τηρεί το πρόγραμμα των ραντεβού της κλινικής. Τα πεδία του έχουν ως εξής:

Arpt_ID: Είναι το πεδίο με το πρωτεύον κλειδί του πίνακα. Είναι τύπου αυτόματης αρίθμησης και αποτελεί τον αύξοντα αριθμό ραντεβού της κλινικής.

Arpt_Date: Πεδίο τύπου ημερομηνίας/ώρας, όπου καταγράφεται μόνο η ημερομηνία του ραντεβού, σε μορφή σύντομης ημερομηνίας.

Arpt_Time: Πεδίο τύπου ημερομηνίας/ώρας, όπου καταγράφεται η ώρα του ραντεβού.

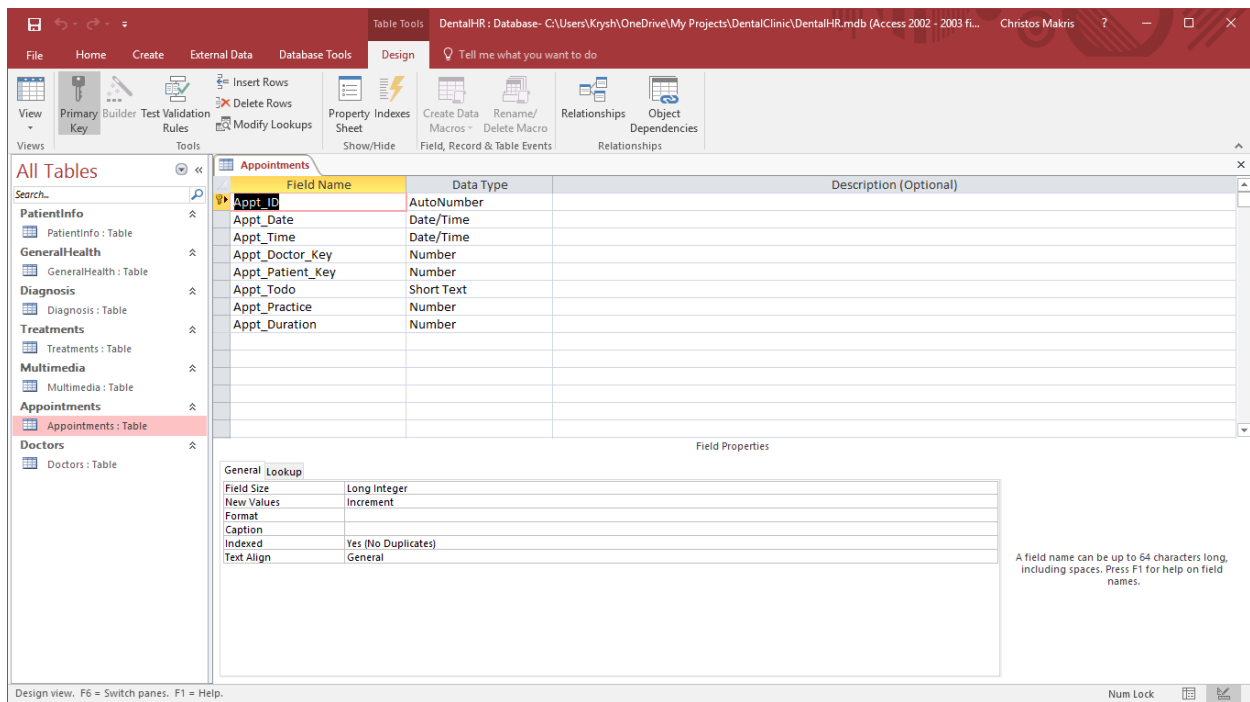
Arpt_Doctor_Key: Δευτερεύον κλειδί του πίνακα Doctors, το οποίο ορίζει τον υπεύθυνο οδοντίατρο για τη συνεδρία.

Arpt_Patient_Key: Δευτερεύον κλειδί του πίνακα PatientInfo, το οποίο ορίζει τον ασθενή για τον οποίο έχει προγραμματιστεί ραντεβού.

Arpt_Todo: Πεδίο κειμένου όπου εκ των προτέρων έχει καταγραφεί προγραμματισμένη επέμβαση ή θεραπεία, όπου και όταν αυτό κρίνεται αναγκαίο. Η χρήση του εν λόγω πεδίου είναι σημαντική για το σωστό προγραμματισμό της διάρκειας των ραντεβού, καθώς και για διάφορες ιδιαιτερότητες και ειδικές ανάγκες της θεραπείας, για τις οποίες πρέπει να γίνει προγραμματισμένη προεργασία.

Arpt_Practice: Πεδίο αριθμού ο οποίος αντιστοιχεί στο συγκεκριμένο θάλαμο εξέτασης – οδοντιατρική έδρα. Οι οδοντιατρικές κλινικές μπορεί να έχουν από μικρό αριθμό εδρών ως και διψήφιο.

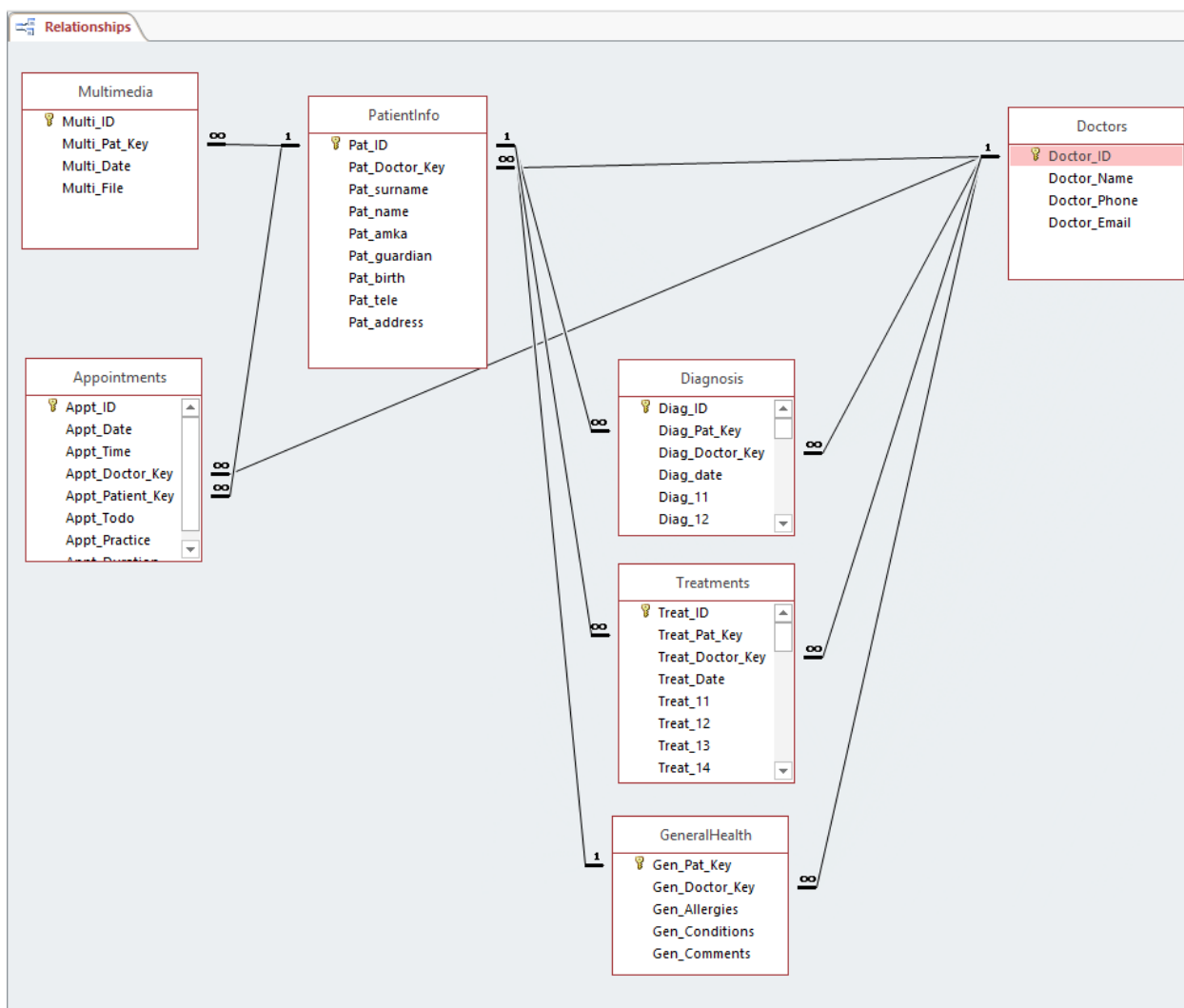
Arpt_Duration: Πεδίο αριθμού στο οποίο συμπληρώνεται ο επιθυμητός από τον οδοντίατρο ή προσυμφωνημένος από τη φύση του περιστατικού, χρόνος διάρκειας ραντεβού. Η καταγραφή γίνεται σε λεπτά της ώρας.



Εικόνα 33. Ο πίνακας Appointments

5.1.8. Σχέσεις πινάκων

Μετά την ανάπτυξη και επεξήγηση των πινάκων της βάσης δεδομένων της εφαρμογής πληροφοριακού συστήματος οδοντιατρικής κλινικής, θα ακολουθήσει η περιγραφή των σχέσεων μεταξύ αυτών των πινάκων.



Εικόνα 34. Οι σχέσεις μεταξύ πινάκων στη βάση δεδομένων

Θεωρώντας τον πίνακα PatientInfo ως τον πλέον σημαντικό, παρατηρούμε ότι το πρωτεύον του κλειδί χρησιμοποιείται ως δευτερεύον σχεδόν σε όλους τους άλλους πίνακες.

ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ	ΣΧΕΣΗ
PatientInfo:Pat_ID	GeneralHealth:Gen_Pat_ID	ΕΝΑ ΠΡΟΣ ΕΝΑ
PatientInfo:Pat_ID	Treatments:Treat_Pat_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
PatientInfo:Pat_ID	Diagnosis:Diag_Pat_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
PatientInfo:Pat_ID	Appointments:Appt_Pat_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
PatientInfo:Pat_ID	Multimedia:Multi_Pat_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ

Άξια σημείωσης αποτελεί η σχέση «ένα προς ένα» μεταξύ των πινάκων PatientInfo και GeneralHealth, ουσιαστικά συνδέοντας και επεκτείνοντας τους δύο πίνακες εγγραφή προς εγγραφή. Ο λόγος αυτού του διαχωρισμού οφείλεται στο διαφορετικό είδος απορρήτου που χαρακτηρίζει τα πεδία των δύο πινάκων. Ο πίνακας με τα δημογραφικά στοιχεία PatientInfo εμπίπτει στο ιδιωτικό απόρρητο, ενώ ο πίνακας GeneralHealth εμπίπτει στο ιατρικό απόρρητο. Με αυτό το διαχωρισμό είναι πιο εύκολος ο έλεγχος χρήσης και πρόσβασης από την εφαρμογή.

Άλλος εξαιρετικά σημαντικός πίνακας είναι ο πίνακας Doctors, το πρωτεύον κλειδί του οποίου εισέρχεται ως δευτερεύον σχεδόν σε όλους τους άλλους πίνακες, επίσης.

ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ	ΣΧΕΣΗ
Doctors:Doctor_ID	GeneralHealth:Gen_Doctor_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
Doctors:Doctor_ID	Treatments:Treat_Doctor_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
Doctors:Doctor_ID	Diagnosis:Diag_Doctor_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
Doctors:Doctor_ID	Appointments:Appt_Doctor_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ
Doctors:Doctor_ID	Patients:Pat_Doctor_Key	ΕΝΑ ΠΡΟΣ ΠΟΛΛΑ

5.2. Ανάπτυξη διεπαφής χρήστη της εφαρμογής

Η ανάπτυξη της διεπαφής χρήστη της εφαρμογής έγινε στο ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio 2015. Ο προγραμματισμός έγινε σε γλώσσα Visual C# ενώ χρησιμοποιήθηκαν διάφορες τεχνολογίες .NET της Microsoft, όπως Winforms και ADO.NET.

Η διεπαφή χρήστη αποτελείται από μια σειρά φορμών, οι οποίες επιτελούν αναγκαίες λειτουργίες και συνδέονται με την υφιστάμενη βάση δεδομένων με σκοπό την μεταφορά και διαχείριση των πληροφοριών αυτής.

Το όνομα της εφαρμογής είναι το «DentalClinic».

5.2.1. Αρχικοποίηση εφαρμογής

Με το ξεκίνημα της εφαρμογής φορτώνεται ο αρχικός κώδικας αυτής.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace DentalClinic
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new IntroForm());}}}

```


Οι όροι κλειδιά «using» μας ενημερώνουν για τις υπάρχουσες κλάσεις που φορτώνονται με σκοπό την υποστήριξη της εφαρμογής. Στη φάση αρχικοποίησης χρησιμοποιούνται οι κλάσεις «System» και «System.Windows.Forms», που αποτελεί μέρος της τεχνολογίας Winforms.

Το namespace της εφαρμογής μας είναι το «DentalClinic»

Τελευταία ενέργεια της αρχικοποίησης είναι η δημιουργία μιας νέας εισαγωγικής φόρμας ως αντικείμενο της αντίστοιχης κλάσης.

```
“Application.Run(new IntroForm());”
```

Όπου «IntroForm» είναι το όνομα της εισαγωγικής φόρμας.

5.2.2. Εισαγωγική Φόρμα

Σε αυτό το στάδιο έχει φορτωθεί η εισαγωγική φόρμα, που αναλαμβάνει το ρόλο του ελέγχου πρόσβασης με χρήση κωδικού και της περαιτέρω ανακατεύθυνσης σε φόρμες επόμενου επιπέδου.

Περιλαμβάνει μια ετικέτα (label) στην οποία αναγράφεται το μήνυμα

«ΠΑΡΑΚΑΛΩ ΕΙΣΑΓΕΤΕ ΚΩΔΙΚΟ ΠΡΟΣΒΑΣΗΣ»

Η ετικέτα ακολουθείται από ένα πεδίο κειμένου στο οποίο γίνεται η εισαγωγή του κωδικού και από τρία κουμπιά, τη λειτουργία των οποίων θα περιγράψουμε.

The screenshot shows a Windows application window titled "DentalClinic". Inside the window, there is a light gray background. At the top center, there is a label with the text "ΠΑΡΑΚΑΛΩ ΕΙΣΑΓΕΤΕ ΚΩΔΙΚΟ ΠΡΟΣΒΑΣΗΣ" (Please enter access code). Below this label is a white rectangular text box for input. At the bottom of the window, there are three blue rectangular buttons with white text. From left to right, the buttons are labeled "ΠΑΝΤΕΒΟΥ", "ΑΣΘΕΝΕΙΣ", and "ΕΞΟΔΟΣ".

Εικόνα 35. Η εισαγωγική φόρμα

Ο κώδικας που συνοδεύει τη φόρμα έχει ως εξής:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DentalClinic
{
    public partial class IntroForm : Form
    {
        public IntroForm()
        {
            InitializeComponent();
        }
    }
}
```

```

    }

    private void buttonAppts_Click(object sender, EventArgs e) // το event πατήματος
του κουμπιου PANTEBOY
    {
        FormAppt foz = new FormAppt(); //Δημιουργία νέας φόρμας Παντεβού
        foz.Show(); //Εμφάνιση της φόρμας
        this.Hide(); // Απόκρυψη της παρούσας
    }

    private void buttonPatients_Click(object sender, EventArgs e) // το event
πατήματος του κουμπιου ΑΣΘΕΝΕΙΣ
    {
        FormPatient pat = new FormPatient();
        pat.Show();
        this.Hide();
    }

    private void buttonExit_Click(object sender, EventArgs e) // το event πατήματος
του κουμπιου ΕΞΟΔΟΣ
    {
        Application.Exit(); //Τερματισμός της εφαρμογής
    }

    private void IntroForm_Load(object sender, EventArgs e) // Κώδικας φόρτωσης της
φόρμας
    {
        buttonAppts.Enabled = false; //Απενεργοποίηση των κουμπιών (επιλογή
ασφάλειας)
        buttonPatients.Enabled = false; //Απενεργοποίηση των κουμπιών (επιλογή
ασφάλειας)
    }

    private void textBoxPass_TextChanged(object sender, EventArgs e) //το event
πληκτρολόγησης στο πεδίο κειμένου του κωδικού
    {
        if (textBoxPass.Text == "123") //έλεγχος του πληκτρολογημένου κειμένου σε
σχέση με τον κωδικό
        {
            buttonAppts.Enabled = true; // αν ο έλεγχος επιτύχει, τα κουμπιά
ενεργοποιούνται
            buttonPatients.Enabled = true;
        }
    }
}

```

Τα κουμπιά που περιέχονται στη φόρμα είναι τρία και έχουν ως εξής:

Κουμπί με ετικέτα «PANTEBOY», με όνομα «buttonAppts»

Κουμπί με ετικέτα «ΑΣΘΕΝΕΙΣ», με όνομα «buttonPatients»

Κουμπί με ετικέτα «ΕΞΟΔΟΣ», με όνομα «buttonExit»

Κατά την εκκίνηση της φόρμας, τα κουμπιά «PANTEBOY» και «ΑΣΘΕΝΕΙΣ» είναι απενεργοποιημένα και δεν μπορούν να πατηθούν. Αυτό συμβαίνει διότι πρέπει πρώτα να ολοκληρωθεί ο έλεγχος πρόσβασης μέσω κωδικού. Γράφοντας τον κωδικό στο αντίστοιχο πεδίο κειμένου, τρέχει ο κώδικας σύγκρισης.

```
private void textBoxPass_TextChanged(object sender, EventArgs e) //το event
πληκτρολόγησης στο πεδίο κειμένου του κωδικού
{
    if (textBoxPass.Text == "123") //έλεγχος του πληκτρολογημένου κειμένου σε
σχέση με τον κωδικό
    {
        buttonAppts.Enabled = true; // αν ο έλεγχος επιτύχει, τα κουμπιά
ενεργοποιούνται
        buttonPatients.Enabled = true;
    }
}
```

Μόνο εφόσον εισαχθεί ο σωστός κωδικός, ενεργοποιούνται τα κουμπιά. Η διαδικασία ελέγχου του κωδικού γίνεται με έλεγχο ισότητας (equals) σε μια πρόταση if.

Παρατηρούμε ότι κατά το πάτημα οποιουδήποτε από τα κουμπιά της φόρμας, ενεργοποιείται ένα συμβάν (event) το οποίο και παραπέμπει στον αντίστοιχο κώδικα της φόρμας.

Κατά το πάτημα του κουμπιού «PANTEBOY» ενεργοποιείται το συμβάν

buttonPatients_Click

το οποίο δημιουργεί ένα νέο αντικείμενο φόρμας ραντεβού "FormAppts"

```
FormAppt foz = new FormAppt();
```

Εμφανίζει το νέο αντικείμενο

```
foz.Show();
```

Και κρύβει την ενεργή φόρμα IntroForm

```
this.Hide();
```

Αντίστοιχη λειτουργία επιτελεί το πάτημα του κουμπιού «ΑΣΘΕΝΕΙΣ», ενεργοποιώντας το συμβάν

```
buttonPatients_Click
```

δημιουργώντας ένα νέο αντικείμενο της φόρμας ασθενών “FormPatients”, εμφανίζοντάς το και κρύβοντας την ενεργή φόρμα IntroForm.

Πάτημα του κουμπιού «ΕΞΟΔΟΣ» ενεργοποιεί το συμβάν

```
buttonExit_Click
```

το οποίο καλεί τη μέθοδο

```
Application.Exit()
```

Έτσι επιτελείται η ασφαλής έξοδος από την εφαρμογή.

5.2.3. Φόρμα «ΑΣΘΕΝΕΙΣ»

Ο κώδικας της φόρμας «ΑΣΘΕΝΕΙΣ» έχει ως εξής:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;
```

namespace DentalClinic

ΑΣΘΕΝΕΙΣ

ΑΝΑΖΗΤΗΣΗ

ΕΠΙΘΕΤΟ: ΑΜΚΑ:

ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΑΜΚΑ
Θεοδώρα	Θεόδωρος	3216549870
Αθανασίου	Γεώργιος	1241251232
Μάρκου	Μαρία	4125215243
Λετίου	Ελένη	2412312541
Ασθενής	Πρώτος	2345678948
Ασθενής	Δεύτερος	9876543210
Ιωάννου	Ιωάννης	1234567890

ΕΠΙΘΕΤΟ: ΟΝΟΜΑ: ΑΜΚΑ:

ΟΝΟΜΑ ΚΗΔΕΜΟΝΑ:

ΗΜΕΡΙΑ ΓΕΝΝΗΣΗΣ: ΤΗΛΕΦΩΝΟ:

ΔΙΕΥΘΥΝΣΗ:

ΠΡΟΣΘΗΚΗ ΔΙΑΓΡΑΦΗ ΑΠΟΘΗΚΕΥΣΗ

ΠΑΝΤΕΘΥ ΓΕΝΙΚΟ ΙΣΤΟΡΙΚΟ ΟΔΟΝΤΙΑΤΡΙΚΟ ΙΣΤΟΡΙΚΟ ΕΞΟΔΟΣ

Εικόνα 36. Η φόρμα "ΑΣΘΕΝΕΙΣ"

```
{
    public partial class FormPatient : Form
    {
        public FormPatient()
        {
            InitializeComponent();
        }

        public int pat_id;
```

Σημαντική μέθοδος σε αυτή τη φόρμα είναι η παρακάτω, η οποία λαμβάνει το βασικό κλειδί από τον πίνακα της βάσης δεδομένων που αντιστοιχεί στον επιλεγμένο ασθενή. Αυτό είναι αναγκαίο να γίνει ώστε οι επόμενες φόρμες που θα δημιουργηθούν, να αφορούν τον ήδη επιλεγμένο ασθενή.

```
public void activePatient() //θα βάλουμε το id του επιλεγμένου ασθενή στο pat_id
{
    pat_id =
    Convert.ToInt32(((DataRowView)this.patientInfoBindingSource.Current).Row["Pat_ID"]);
    // μεταφέρουμε το current pat_id στην public μεταβλητή
```

```

        // pat_id = this.patientInfoBindingSource.Position; //μεταφέρουμε το pos του
        patientinfo, που αντιστοιχεί στο primary key του πίνακα
    }

```

```

e)
    private void patientInfoBindingNavigatorSaveItem_Click(object sender, EventArgs
    {
        this.Validate();
        this.patientInfoBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.dentalHRDataSet);

    }

    private void FormAppt_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'dentalHRDataSet.PatientInfo'
        table. You can move, or remove it, as needed.
        this.patientInfoTableAdapter.Fill(this.dentalHRDataSet.PatientInfo);

    }

```

Ο παρακάτω κώδικας χρησιμοποιείται για την αναζήτηση ασθενών στη βάση δεδομένων βάσει του επιθέτου τους. Τα αποτελέσματα της αναζήτησης εμφανίζονται με μορφή πίνακα στο αντίστοιχο dataGrid.

```

    private void textBoxNameSearch_TextChanged(object sender, EventArgs e)
    {
        //από εδώ θα πάρουμε το κείμενο του textbox ΕΠΙΘΕΤΟ την ώρα που
        πληκτρολογείται, έναν-έναν χαρακτήρα
        try
        {
            //μετατρέπουμε το κείμενο σε μορφή κατάλληλη για query, ένα string με το
            wildcard %
            string search = $"{textBoxNameSearch.Text}%";

            //γεμίζουμε το table μέσω της μεθόδου fillbyname που είναι query βάσει
            επιθέτου
            this.patientInfoTableAdapter.FillByName(this.dentalHRDataSet.PatientInfo,
            search);
        }
        catch (System.Exception ex)
        {
            System.Windows.Forms.MessageBox.Show(ex.Message);
        }
    }

```

```

    }

    private void textBoxAmkaSearch_TextChanged(object sender, EventArgs e)
    //ισχύουν αντίστοιχα με τα παραπάνω
    {
        try
        {
            string searchA = $"{textBoxAmkaSearch.Text}%";
            this.patientInfoTableAdapter.FillByAMKA(this.dentalHRDataSet.PatientInfo,
searchA);
        }
        catch (System.Exception ex)
        {
            System.Windows.Forms.MessageBox.Show(ex.Message);
        }
    }
}

```

Ο συγκεκριμένος κώδικας διαχειρίζεται το πάτημα του κουμπιού «PANTEBOY», δημιουργώντας μια νέα φόρμα ραντεβού με ενεργοποιημένο τον επιλεγμένο ασθενή από την υφιστάμενη φόρμα.

```

    private void buttonNewAppt_Click(object sender, EventArgs e)
    {
        activePatient();
        Form Appt = new FormAppt(pat_id); //δημιουργώ νέα φόρμα ραντεβου formappt
        μεταφέροντας το pat_id του τρέχοντος ασθενή
        Appt.Show(); //εμφανίζω τη φόρμα
        this.Hide(); //κρύβω αυτή
    }

    private void buttonNewPat_Click(object sender, EventArgs e)
    {
    }

    private void buttonToDental_Click(object sender, EventArgs e)
    {
        activePatient();
        Form Dental = new FormDental(pat_id); //δημιουργώ νέα φόρμα οδοντιατρικής
        εξέτασης formdental μεταφέροντας το pat_id του τρέχοντος ασθενή
        Dental.Show(); //εμφανίζω τη φόρμα
        this.Hide(); //κρύβω αυτή
    }
}

```



```

private void buttonToGeneral_Click(object sender, EventArgs e)
{
    activePatient();
    Form General = new FormGeneral(pat_id);
    General.Show();
    this.Close();
}

private void buttonImageFiles_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void bindingNavigatorDeleteItem_Click(object sender, EventArgs e)
//εξειδικευμένος χειρισμός της διαγραφής ασθενούς
{
    if (MessageBox.Show("Επιθυμείτε την οριστική διαγραφή του ασθενή;",
    "Προσοχή!", MessageBoxButtons.YesNo) ==
    System.Windows.Forms.DialogResult.Yes) //δημιουργία παραθύρου προειδοποίησης
    και ενεργοποίηση μόνο σε περίπτωση θετικής απάντησης
    {
        try
        {
            this.patientInfoBindingSource.RemoveCurrent(); //διαγραφή τρέχουσας
εγγραφής
            this.Validate();
            this.patientInfoBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.dentalHRDataSet); //ενημέρωση
dataset
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

5.2.4. Φόρμα Γενικού Ιστορικού Υγείας

Εικόνα 37. Η Φόρμα Γενικού Ιστορικού Υγείας

Ο κώδικας της φόρμας γενικού ιστορικού έχει ως εξής:

```
namespace DentalClinic
{
    public partial class FormGeneral : Form
    {
        public int pos;

        public FormGeneral(int pos)
        {
            InitializeComponent();
            this.pos = pos;
        }

        private void generalHealthBindingNavigatorSaveItem_Click(object sender,
EventArgs e)
        {

```

```

        this.Validate();
        this.generalHealthBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.dentalHRDataSet);

    }

    private void FormGeneral_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'dentalHRDataSet.PatientInfo'
        table. You can move, or remove it, as needed.
        this.patientInfoTableAdapter.Fill(this.dentalHRDataSet.PatientInfo);
        // TODO: This line of code loads data into the 'dentalHRDataSet.GeneralHealth'
        table. You can move, or remove it, as needed.
        this.generalHealthTableAdapter.Fill(this.dentalHRDataSet.GeneralHealth);
        this.patientInfoBindingSource.Position = pos;
    }

    private void toPatients_Click(object sender, EventArgs e)
    {
        this.Hide();
        Form pat = new FormPatient();
        pat.Show();
    }

    private void toDental_Click(object sender, EventArgs e)
    {
        this.Hide();
        Form dent = new FormDental(pos);
        dent.Show();
    }
}

```

Σε αυτή τη φόρμα ο κώδικας επιτελεί λειτουργίες που έχουμε ήδη δει προηγουμένως, όπως το να ανοίξει τις εγγραφές του προεπιλεγμένου ασθενούς, να δημιουργήσει νέες φόρμες με βάση τα υπάρχοντα κουμπιά και να κλείσει την υπάρχουσα φόρμα.

5.2.5. Φόρμα Οδοντιατρικής Εξέτασης

Εικόνα 38. Η φόρμα Οδοντιατρικής Εξέτασης

Ο κώδικας της φόρμας οδοντιατρικής εξέτασης έχει ως εξής:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.IO; //αναγκαίο για τη διαχείριση αρχείων εικόνων
```

```
namespace DentalClinic  
{  
    public partial class FormDental : Form  
    {
```

Σε αυτή τη φόρμα θα χρησιμοποιήσουμε πληθώρα μεταβλητών, όπως αυτές που δηλώνουν τον επιλεγμένο, από την προηγούμενη φόρμα, ασθενή (pat_id), το ΑΜΚΑ του ασθενή, τη διαδρομή στο σύστημα αποθήκευσης που θα τοποθετηθούν οι εικόνες και ακτινογραφίες που θα προσθέσουμε (imagespath), καθώς και η διαδρομή της

ακτινογραφίας που θα επιλέξουμε για περεταίρω επεξεργασία (xRayPath) στη φόρμα επεξεργασίας εικόνων.

```
public int pat_id; //δηλώνω public την Int που θα κουβαλήσει τον προεπιλεγμένο
ασθενή από τη formpatient
    public string amka; // το αμκα του ενεργού ασθενή
    public string imagespath; //θα μπει το Path που θα σωθούν οι εικόνες
    public string xRayPath; //θα μπει το Path που δείχνει την ακτινογραφία προς
αποστολή στη νέα φόρμα

    public FormDental(int pat_id) //η φόρμα μας καλείται με overload το pat_id που
είχαμε επιλεξει
    {
        InitializeComponent();
        this.pat_id = pat_id; //το overload μεταφέρεται στο public int για χρήση στην
υπόλοιπη φόρμα

    }
```

```
e) private void diagnosisBindingNavigatorSaveItem_Click(object sender, EventArgs
{
    this.Validate();
    this.diagnosisBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dentalHRDataSet);

}

private void FormDental_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'dentalHRDataSet.PatientInfo'
table. You can move, or remove it, as needed.
    this.patientInfoTableAdapter.Fill(this.dentalHRDataSet.PatientInfo);
    // TODO: This line of code loads data into the 'dentalHRDataSet.PatientInfo'
table. You can move, or remove it, as needed.
    this.patientInfoTableAdapter.Fill(this.dentalHRDataSet.PatientInfo);
    // TODO: This line of code loads data into the 'dentalHRDataSet.Diagnosis'
table. You can move, or remove it, as needed.
    this.diagnosisTableAdapter.Fill(this.dentalHRDataSet.Diagnosis);
}
```

Εδώ γίνεται η επιλογή του ασθενή και φορτώνεται η εγγραφή του από τον πίνακα PatientInfo.

```
this.patientInfoBindingSource.Position = patientInfoBindingSource.Find("Pat_ID",  
pat_id); //επιλέγουμε τον ασθενή που μεταφέραμε από την προηγούμενη φόρμα  
κάνοντας αναζήτηση του pat_id του στο bindingsource
```

Από αυτή την εγγραφή θα πάρουμε το ΑΜΚΑ του ασθενούς, μεταφέροντας το αντίστοιχο πεδίο του της εγγραφής στη μεταβλητή που δημιουργήσαμε γι αυτό το σκοπό.

```
amka =  
((DataRowView)this.patientInfoBindingSource.Current).Row["Pat_amka"].ToString();  
//παίρνουμε τον ΑΜΚΑ από το current του bindingsource και το μετατρέπουμε από  
αντικείμενο σε string
```

Και στη συνέχεια θα χρησιμοποιήσουμε το ΑΜΚΑ ως όνομα φακέλου, για να δημιουργήσουμε ένα φάκελο στο σύστημα αποθήκευσης του υπολογιστή, όπου θα αποθηκευτούν οι φωτογραφίες και ακτινογραφίες που θέλουμε να εισάγουμε στο φάκελο του ασθενούς μας.

```
imagespath = "images/" + amka; //φτιάχνω το Path που θα σωθούν οι εικόνες  
  
if (Directory.Exists(imagespath))//ελέγχει αν προϋπάρχει φάκελος εικόνων και  
μόνο τότε επιχειρεί να τις εμφανίσει  
{ showImages(); }  
}
```

Αυτή η μέθοδος εμφανίζει τις μικρογραφίες όλων των εικόνων που έχουμε προσθέσει για τον συγκεκριμένο ασθενή, στο αντίστοιχο imageList.

```
private void showImages() //φορτώνει τις εικόνες στα imagelist, listview  
{  
    imageList1.Images.Clear();  
    listView1.Clear();  
}
```

```

        DirectoryInfo dir = new DirectoryInfo(imagespath); // διαβάζει τα στοιχεία του
ενεργού καταλόγου

        foreach (FileInfo file in dir.GetFiles()) // φορτώνει τα στοιχεία κάθε αρχείου που
βρήκε στον ενεργό κατάλογο
        {
            imageList1.Images.Add(Image.FromFile(file.FullName)); } //προσθέτει κάθε
αρχείο στο imagelist

        for (int i = 0; i < imageList1.Images.Count; i++) //για κάθε εικόνα του imagelist...
        {
            ListViewItem item = new ListViewItem(); //δημιουργεί ένα νέο αντικείμενο
listview..
            item.ImageIndex = i; // του δίνει τον αα της εικόνας από το imagelist
            listView1.Items.Add(item); // και το προσθέτει στο listview
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.Close();
        FormPatient pat = new FormPatient();
        pat.Show();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

```

Αυτός ο κώδικας διαχειρίζεται το πάτημα του κουμπιού εισαγωγής εικόνων, δημιουργώντας τον αντίστοιχο φάκελο, ενεργοποιώντας το φίλτρο επιλογής μόνο ορισμένων τύπων αρχείου και αντιγράφει το σύνολο των επιλεγμένων αρχείων από το παράθυρο διαλόγου που προκύπτει.

```

private void button1_Click(object sender, EventArgs e)
{
    Directory.CreateDirectory(imagespath); //δημιουργία directory εικόνων

```

```

        this.openFileDialog1.Filter = "X-ray, Pictures
(*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG"; //φίλτρο για να φαίνονται μόνο
οι εικόνες στο filedialogue
        openFileDialog1.ShowDialog();
    }

    private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
    {
        foreach (String sourcefile in openFileDialog1.FileNames) //foreach όπου η
        πλήρης διαδρομή κάθε επιλεγμένου αρχείου περνάει από το array στο sourcefile
        {
            string filename = Path.GetFileName(sourcefile); //φτιαχνουμε το filename
            μόνο με το ονομα, απορρίπτοντας τη διαδρομή
            string filepath = imagespath + "/" + filename; //φτιάχνουμε τη διαδρομή στόχο
            (που θα εγγράψουμε το αρχείο)
            File.Copy(sourcefile, filepath); //αντιγράφουμε τα επιλεγμένα αρχεία

            showImages(); //του ζητάμε να εμφανίσει - ξαναεμφανίσει τις εικόνες στο
listview
        }
    }

```

Ο συγκεκριμένος κώδικας διαχειρίζεται την επιλογή μιας μικρογραφίας εικόνας με το ποντίκι και την εμφάνιση αυτής σε μεγαλύτερο μέγεθος στο αντίστοιχο παράθυρο.

```

    private void listView1_ItemActivate(object sender, EventArgs e)
    {
        DirectoryInfo dir = new DirectoryInfo(imagespath); // διαβάζει τα στοιχεία του
ενεργού καταλόγου

        string[] files = Directory.GetFiles(imagespath); //φτιάχνουμε ένα array από
strings και το φορτώνουμε με τα paths των αρχείων του ενεργού καταλόγου

        // label1.Text = this.listView1.FocusedItem.Index.ToString(); //testing

        xRayPath = files[this.listView1.FocusedItem.Index]; // φτιαχνουμε μια μεταβλητή
string με τιμή το path από το προηγούμενο array που έχει ίδιο index με το επιλεγμένο
αρχείο
        pictureBox1.Image = System.Drawing.Image.FromFile(xRayPath); // το
δείχνουμε στο pictureBox1 φορτώνοντας το κατευθείαν από το αρχείο (και όχι από το
listview που έχει χαμηλή ανάλυση)
    }

```



```

private void pictureBox1_DoubleClick(object sender, EventArgs e)
{
    FormXrayProcess fx = new FormXrayProcess(xRayPath);
    fx.ShowDialog();
}
}
}

```

5.2.6. Φόρμα Μελέτης Και Επεξεργασίας Ακτινογραφιών



Εικόνα 39. Φόρμα Μελέτης και Επεξεργασίας Ακτινογραφιών

Ο κώδικας της φόρμας μελέτης και επεξεργασίας ακτινογραφιών έχει ως εξής:

```

namespace DentalClinic
{
    public partial class FormXrayProcess : Form
    {
        public string xRayPath; //το path για το αρχείο εικόνας που έρχεται από το overload
        public Bitmap xray;
        public Bitmap newXray;
    }
}

```

Η υπερφόρτωση που λαμβάνουμε από την προηγούμενη φόρμα, είναι η διαδρομή του αρχείου εικόνας που θα εμφανίσουμε.

```
public FormXrayProcess(string xRayPath)
{
    InitializeComponent();
    this.xRayPath = xRayPath;
    ShowXray();
}

private void ShowXray()
{
    pictureBox1.Image = System.Drawing.Image.FromFile(xRayPath);
    xray = (Bitmap)Bitmap.FromFile(xRayPath); //μετατρέπουμε το image.fromfile σε
bitmap
}

private void buttonClose_Click(object sender, EventArgs e)
{
    this.Dispose();
}
```

Το κουμπί «ΑΡΝΗΤΙΚΟ» καλεί την αντίστοιχη μέθοδο της κλάσης ImageProcessing, και εμφανίζει το επιστρεφόμενο αυτής.

```
private void buttonNegative_Click(object sender, EventArgs e)
{
    xray = ImageProcessing.DrawAsNegative(xray);
    pictureBox1.Image = xray;
}

private void buttonReset_Click(object sender, EventArgs e)
{
    ShowXray();
    //TODO ΝΑ ΕΠΑΝΑΦΕΡΩ ΤΑ SLIDERS
}
```

Το συμβάν της κύλισης του δείκτη Γάμμα λειτουργεί με αυτόν τον κώδικα, ο οποίος λαμβάνει τη μέτρηση του δείκτη, τη μετατρέπει καταλλήλως σε αριθμό που να έχει νόημα ως γάμμα εικόνας και καλεί την αντίστοιχη μέθοδο της κλάσης ImageProcessing με τη χρήση του παραγόμενου γάμμα. Κατόπιν, εμφανίζει την επιστρεφόμενη εικόνα.

```

private void trackBarGamma_Scroll(object sender, EventArgs e)
{
    decimal d = this.trackBarGamma.Value; //μετατρέπουμε το Int του value σε
decimal
    d = d / 10; //Κάνουμε την πράξη εδώ διότι στην πιο πάνω προέχει η μετατροπή
και χάνονται τα δεκαδικά
    float newgamma = (float)d; // μετατρέπουμε το dec σε float, κατάλληλο για τη
ρύθμιση gamma

    newXray = ImageProcessing.ChangeGamma(xray, newgamma); //η εικονα
newXray είναι το επιστρεφόμενο αντικείμενο της μεθόδου changegamma
    pictureBox1.Image = newXray; //εμφανίζουμε τη νέα εικόνα
}

```

Μετακίνηση των δεικτών Αντίθεσης ή/και Φωτεινότητας, και ZOOM, ενεργοποιούν τις παρακάτω μεθόδους.

```

private void trackBarContrast_Scroll(object sender, EventArgs e)
{
    ContrastBrightness();
}

private void trackBarBrightness_Scroll(object sender, EventArgs e)
{
    ContrastBrightness();
}

private void trackBarZoom_Scroll(object sender, EventArgs e) //TODO να τα βάλω
όλα τα δημιουργούμενα αντικείμενα στο imageprocessing για να μη γεμίζει η μνήμη
{
    if (newXray != null) { newXray.Dispose(); } //κατά περίεργο τρόπο ξεφορτώνει
κάποια instances κατά το zoom out, απελευθερώνοντας μνήμη προς αποφυγή leak
    int zoom = this.trackBarZoom.Value;

    Size newSize = new Size((int)(xray.Width * zoom), (int)(xray.Height * zoom));
    newXray = new Bitmap(xray, newSize);
    pictureBox1.Image = newXray;
}

private void ContrastBrightness() //TODO να τα βάλω όλα τα δημιουργούμενα
αντικείμενα στο imageprocessing για να μη γεμίζει η μνήμη
{
    decimal c = this.trackBarContrast.Value; //διαβάζουμε το slider στη δεκαδική c

```

```

        decimal b = this.trackBarBrightness.Value - 10; //αφαιρούμε 10 ώστε η βασική
        τιμή να είναι 0
        c /= 10; //διαιρούμε με 10 για να πάρουμε τιμές μεταξύ 0-1
        b /= 10;
        float contr = (float)c; //μετατρέπουμε σε float
        float bright = (float)b;

        //label5.Text = contr.ToString();
        //label6.Text = bright.ToString();

        newXray = ImageProcessing.ContrastBrightness(xray, contr, bright); //η εικόνα
        newXray είναι το επιστρεφόμενο αντικείμενο της μεθόδου ContrastBrightness
        pictureBox1.Image = newXray; //εμφανίζουμε τη νέα εικόνα
    }
}

```

Όπως βλέπουμε, οι περισσότερες λειτουργίες επεξεργασίας εικόνας γίνονται από τις μεθόδους της κλάσης ImageProcessing, την οποία θα αναλύσουμε στη συνέχεια.

5.2.7. Κλάση ImageProcessing

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing; //Οι κλάσεις χρήσης κ επεξεργασίας γραφικών
using System.Drawing.Drawing2D; //Οι κλάσεις χρήσης κ επεξεργασίας γραφικών
using System.Drawing.Imaging; //Οι κλάσεις χρήσης κ επεξεργασίας γραφικών

namespace DentalClinic
{
    public static class ImageProcessing
    {

```

Για να γίνουν οι διάφοροι μετασχηματισμοί της εικόνας, αυτή πρέπει να μετατραπεί σε μορφή ARGB.

```
private static Bitmap GetArgbCopy(Image sourceImage)
{
    Bitmap bmpNew = new Bitmap(sourceImage.Width, sourceImage.Height,
PixelFormat.Format32bppArgb);

    using (Graphics graphics = Graphics.FromImage(bmpNew))
    {
        graphics.DrawImage(sourceImage, new Rectangle(0, 0, bmpNew.Width,
bmpNew.Height), new Rectangle(0, 0, bmpNew.Width, bmpNew.Height),
GraphicsUnit.Pixel);
        graphics.Flush();
    }

    return bmpNew;
}

private static Bitmap ApplyColorMatrix(Image sourceImage, ColorMatrix
colorMatrix)
{
    Bitmap bmp32BppSource = GetArgbCopy(sourceImage);
    Bitmap bmp32BppDest = new Bitmap(bmp32BppSource.Width,
bmp32BppSource.Height, PixelFormat.Format32bppArgb);

    using (Graphics graphics = Graphics.FromImage(bmp32BppDest))
    {
        ImageAttributes bmpAttributes = new ImageAttributes();
        bmpAttributes.SetColorMatrix(colorMatrix);

        graphics.DrawImage(bmp32BppSource, new Rectangle(0, 0,
bmp32BppSource.Width, bmp32BppSource.Height),
0, 0, bmp32BppSource.Width, bmp32BppSource.Height,
GraphicsUnit.Pixel, bmpAttributes);
    }

    bmp32BppSource.Dispose();

    return bmp32BppDest;
}
```

Η μέθοδος ContrastBrightness ρυθμίζει ταυτόχρονα τη φωτεινότητα και αντίθεση της εικόνας. Αυτό γίνεται με χρήση ενός array τύπου ColorMatrix, που ενσωματώνει τις ρυθμίσεις που επιθυμούμε.

```
public static Bitmap ContrastBrightness (Bitmap sourceImage, float contrast, float
brightness)
{
    ColorMatrix colorMatrix = new ColorMatrix(new float[][]
    {
        new float[] {contrast, 0, 0, 0, 0},
        new float[] {0, contrast, 0, 0, 0},
        new float[] {0, 0, contrast, 0, 0},
        new float[] {0, 0, 0, 1.0f, 0},
        new float[] {brightness, brightness, brightness, 0, 1}
    });

    return ApplyColorMatrix(sourceImage, colorMatrix);
}

public static Bitmap ChangeGamma (Bitmap sourceImage, float newgamma)
{
    Bitmap bmp32BppSource = GetArgbCopy(sourceImage);
    Bitmap bmp32BppDest = new Bitmap(bmp32BppSource.Width,
bmp32BppSource.Height, PixelFormat.Format32bppArgb);

    using (Graphics graphics = Graphics.FromImage(bmp32BppDest))
    {
        ImageAttributes bmpAttributes = new ImageAttributes();
        bmpAttributes.SetGamma(newgamma);

        graphics.DrawImage(bmp32BppSource, new Rectangle(0, 0,
bmp32BppSource.Width, bmp32BppSource.Height),
0, 0, bmp32BppSource.Width, bmp32BppSource.Height,
GraphicsUnit.Pixel, bmpAttributes);
    }

    bmp32BppSource.Dispose();

    return bmp32BppDest;
}

public static Bitmap DrawAsNegative (Bitmap sourceImage)
{
    ColorMatrix colorMatrix = new ColorMatrix(new float[][]
    {
```

```

        new float[] {-1, 0, 0, 0, 0},
        new float[] {0, -1, 0, 0, 0},
        new float[] {0, 0, -1, 0, 0},
        new float[] {0, 0, 0, 1, 0},
        new float[] {1, 1, 1, 1, 1}
    });

    return ApplyColorMatrix(sourceImage, colorMatrix);
}

public static Bitmap DrawAsGrayscale(this Image sourceImage)
{
    ColorMatrix colorMatrix = new ColorMatrix(new float[][]
    {
        new float[] {.3f, .3f, .3f, 0, 0},
        new float[] {.59f, .59f, .59f, 0, 0},
        new float[] {.11f, .11f, .11f, 0, 0},
        new float[] {0, 0, 0, 1, 0},
        new float[] {0, 0, 0, 0, 1}
    });

    return ApplyColorMatrix(sourceImage, colorMatrix);
}
}

```

Όπως είδαμε, όλοι οι μετασχηματισμοί της εικόνας γίνονται με τη χρήση του ColorMatrix. Αυτό είναι ένα array 5x5 με βάση το οποίο μπορεί να γίνει πολλαπλασιασμός πινάκων μεταξύ αυτού και ενός πίξελ μιας εικόνας. Ο πίνακας έχει τη μορφή RGBAW. Εισάγοντας διαφορετικές τιμές παραμέτρων στο array, επιτυγχάνουμε τις αλλαγές της εικόνας κατά τον πολλαπλασιασμό.

5.2.8. Φόρμα Ραντεβού

ENERΓΟΣ ΦΑΚΕΛΟΣ Ιωάννου Ιωάννης

June 2017

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Today: 06-Jun-17

ΩΡΑ: 12:00

ΔΙΑΡΚΕΙΑ: 30 ΛΕΠΤΑ

ΙΑΤΡΟΣ: ΙΑΤΡΕΙΟ 1
ΙΑΤΡΕΙΟ 2
ΙΑΤΡΕΙΟ 3

ιατρ. Τρίτος

ΣΗΜΕΙΩΣΕΙΣ:

ΘΑ ΓΙΝΟΥΝ ΕΜΦΑΝΙΣΗ 13 ΕΞΑΓΩΓΗ 48

QPA	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
9:00 AM	Θεοδώρου	Θεόδωρος
10:00 AM	Αθανασίου	Γεώργιος
11:00 AM	Λετίου	Ελένη

QPA	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
9:00 AM	Μάρκου	Μαρία
12:00 PM	Ιωάννου	Ιωάννης

QPA	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
10:00 AM	Ιωάννου	Ιωάννης
1:00 PM	Αθανασίου	Γεώργιος

ΠΡΟΣΘΗΚΗ PANTEBOI

ΛΕΙΤΤΟΜΕΡΕΙΕΣ PANTEBOI

ΔΙΑΓΡΑΦΗ PANTEBOI

ΕΠΙΛΟΓΗ ΑΣΘΕΝΟΥΣ

ΟΔΟΝΤΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ

Εικόνα 40. Η φόρμα Ραντεβού

Ο κώδικας της φόρμας ραντεβού έχει ως εξής:

```
namespace DentalClinic
{
    public partial class FormAppt : Form
    {
        public int pat_id; //δηλώνω public την Int που θα κουβαλήσει τον προεπιλεγμένο
        ασθενή από τη formpatient

        public FormAppt() // κλήση χωρίς overload - για χρήση από introform
        {
            InitializeComponent();
        }

        public FormAppt(int pat_id) //η φόρμα μας καλείται με overload το pat_id που
        είχαμε επιλεγεί
        {
            InitializeComponent();
            this.pat_id = pat_id; //το overload μεταφέρεται στο public int για χρήση στην
            υπόλοιπη φόρμα
        }
    }
}
```



```

private void FormAppt_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'dentalHRDataSet.PatientInfo'
table. You can move, or remove it, as needed.
    this.patientInfoTableAdapter.Fill(this.dentalHRDataSet.PatientInfo);
    // TODO: This line of code loads data into the 'dentalHRDataSet.Doctors' table.
You can move, or remove it, as needed.
    this.doctorsTableAdapter.Fill(this.dentalHRDataSet.Doctors);
    // TODO: This line of code loads data into the 'dentalHRDataSet.DataTable1'
table. You can move, or remove it, as needed.
    this.apptPractice1TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice1,
DateTime.Today); //Γεμίζουμε το TableAdapter με πηγή τον πίνακα apptpractice1 μέσω
του custom query fillbydate.
    this.apptPractice2TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice2,
DateTime.Today);
    this.apptPractice3TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice3,
DateTime.Today);

    this.patientInfoBindingSource.Position =
patientInfoBindingSource.Find("Pat_ID", pat_id); //επιλέγουμε τον ασθενή που
μεταφέραμε από την προηγούμενη φόρμα κάνοντας αναζήτηση του pat_id του στο
bindingsource

}

// private void monthCalendar1_DateChanged(object sender, DateRangeEventArgs
e)
// {
// Ισως να μπορω να χρησιμοποιήσω και αυτό
// }

private void monthCalendar1_DateSelected(object sender, DateRangeEventArgs
e)
{
    fillAppts();
}

private void fillAppts()
{
    DateTime dt = monthCalendar1.SelectionStart;
    //ToString("dd-mm-yyyy");

```

//μπορώ να βάλω το παραπάνω αν θελω να μετατρέψω το selectionstart σε απλο string, αλλά το type datetime δουλεύει και συγκρίνεται καλά με το υπάρχον query

```
        this.apptPractice1TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice1,
dt); //Το fillbydate είναι δικό μου query στον πίνακα ραντεβού με βάση την επιλεγμένη
ημερομηνία
        this.apptPractice2TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice2,
dt);
        this.apptPractice3TableAdapter.FillByDate(this.dentalHRDataSet.ApptPractice3,
dt);
    }
```

```
private void addAppt_Click(object sender, EventArgs e)
{
    DateTime dt = monthCalendar1.SelectionStart; //παίρνουμε την επιλεγμένη
ημέρα σε μορφή datetime
    DateTime tm = dateTimePicker1.Value; //παίρνουμε την επιλεγμένη ώρα σε
μορφή datetime

    if (this.listBox1.SelectedIndex != -1) // ελέγχουμε αν έχει επιλεγεί ιατρείο. αν δεν
έχει επιλεγεί, το selectedIndex έχει τιμή -1
    {
        DentalHRDataSet.AppointmentsRow newappt =
this.dentalHRDataSet.Appointments.NewAppointmentsRow(); //ΚΑΙΝΟΥΡΙΑ ΕΓΓΡΑΦΗ
(Row) ΤΟΥ ΠΙΝΑΚΑ APPOINTMENTS ΠΟΥ ΘΑ ΓΕΜΙΣΟΥΜΕ
        newappt.Appt_Patient_Key =
Convert.ToInt32(((DataRowView)this.patientInfoBindingSource.Current).Row["Pat_ID"]);
//μετατρέπουμε τη string temp σε int και γράφουμε το pat_id
        newappt.Appt_Date = dt;
        newappt.Appt_Doctor_Key = comboBox2.SelectedIndex + 1; // το +1 είναι
επειδή το index ξεκινάει από 0 (zero based index) ενώ στη βδ ξεκινάει από 1
        newappt.Appt_Practice = listBox1.SelectedIndex + 1; //το ίδιο
        newappt.Appt_Time = tm;
        newappt.Appt_Todo = this.appt_Notes.Text;
        newappt.Appt_Duration = this.comboBox1.SelectedIndex;

        this.dentalHRDataSet.Appointments.Rows.Add(newappt);

this.appointmentsTableAdapter1.Update(this.dentalHRDataSet.Appointments);
fillAppts();
    }
    else { MessageBox.Show("Επιλέξτε Ιατρείο", "ΠΡΟΣΟΧΗ"); }
```

```

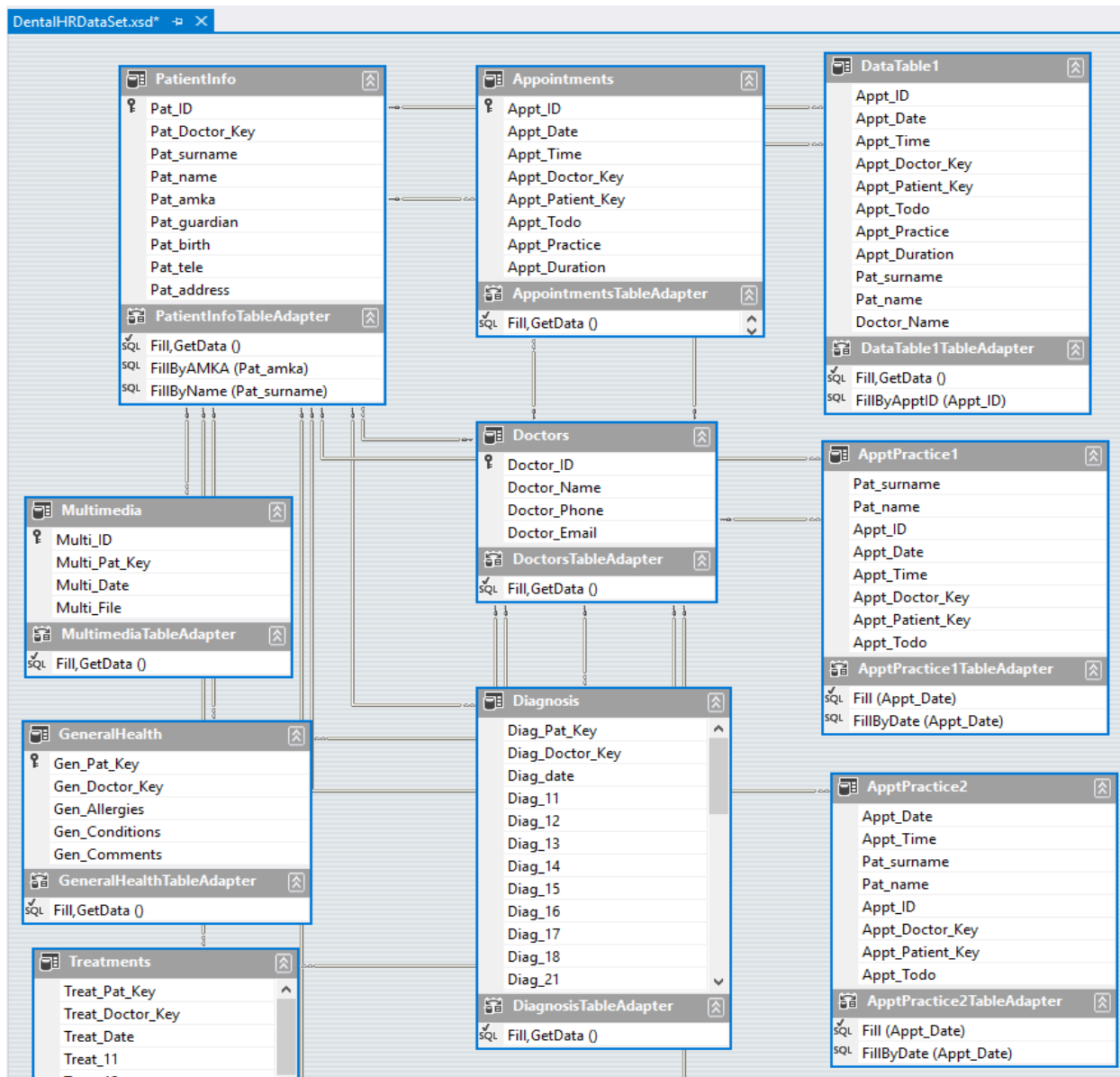
    }
    private void buttonToPatient_Click(object sender, EventArgs e)
    {
        this.Close();
        FormPatient fp = new FormPatient();
        fp.Show();
    }

    private void buttonToDental_Click(object sender, EventArgs e)
    {
        this.Close();
        FormDental fd = new FormDental(pat_id);
        fd.Show(); } } }

```

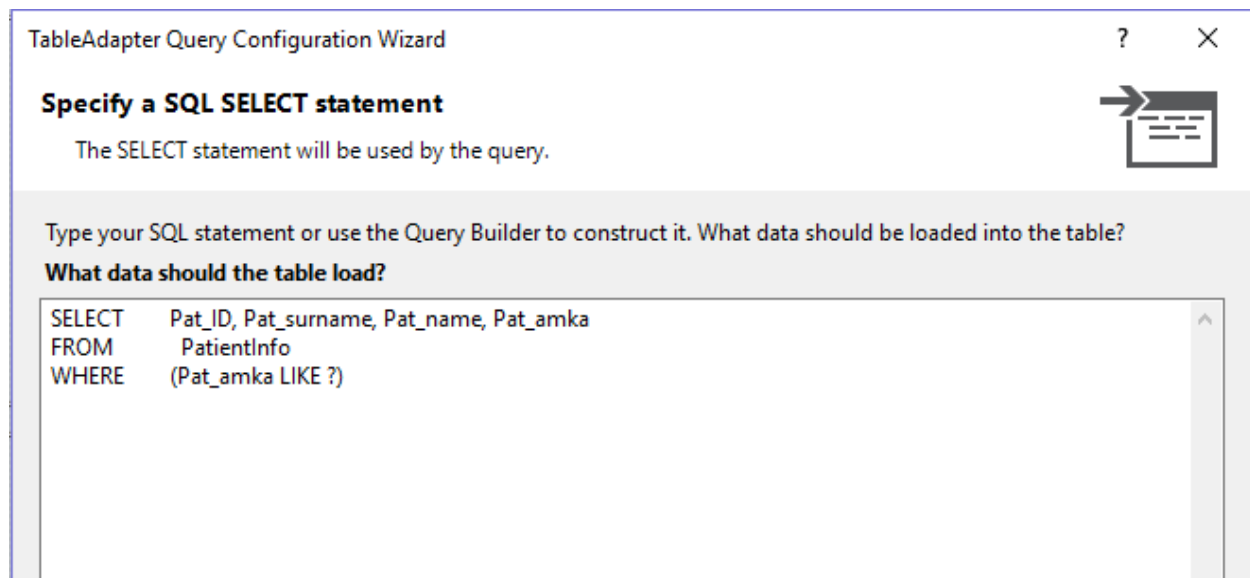
5.2.9. Εξειδικευμένα Query και παράγωγοι πίνακες στο Dataset

Η εφαρμογή που αναπτύχθηκε κάνει εκτεταμένη χρήση της τεχνολογίας Dataset. Το dataset είναι μια εικονική βάση δεδομένων, αντίγραφο της υπάρχουσας DentalHR.mdb, με διατήρηση όλων των πινάκων και των μεταξύ τους σχέσεων. Το πλεονέκτημα της εργασίας με το εικονικό αντίγραφο της βάσης δεδομένων είναι η ταχύτητα και ασφάλεια που προσφέρει καθώς όλες οι αναζητήσεις, αναγνώσεις και εγγραφές γίνονται τοπικά στο εικονικό αντίγραφο πρώτα, και στη συνέχεια μεταφέρονται στην κανονική βάση δεδομένων. Στη σχετική εικόνα μπορούμε να παρατηρήσουμε την παρόμοια δομή του dataset με αυτή της βάσης δεδομένων. Υπάρχουν όμως και διαφορές, καθώς το dataset περιλαμβάνει παράγωγους πίνακες που δεν υφίστανται στην αρχική βάση δεδομένων καθώς και πρόσθετα queries για άμεση χρήση χωρίς ανάγνωση από τη βάση. Τέτοιοι πίνακες είναι για παράδειγμα οι “ApptPractice 1 – 2 – 3”, οι οποίοι χρησιμεύουν στην πλήρωση του προγράμματος ραντεβού κάθε ιατρού. Είναι παράγωγοι πίνακες, διότι περιέχουν πεδία από περισσότερους του ενός αρχικούς πίνακες. Δημιουργούνται με την εκτέλεση ειδικών εντολών Access SQL γλώσσας.



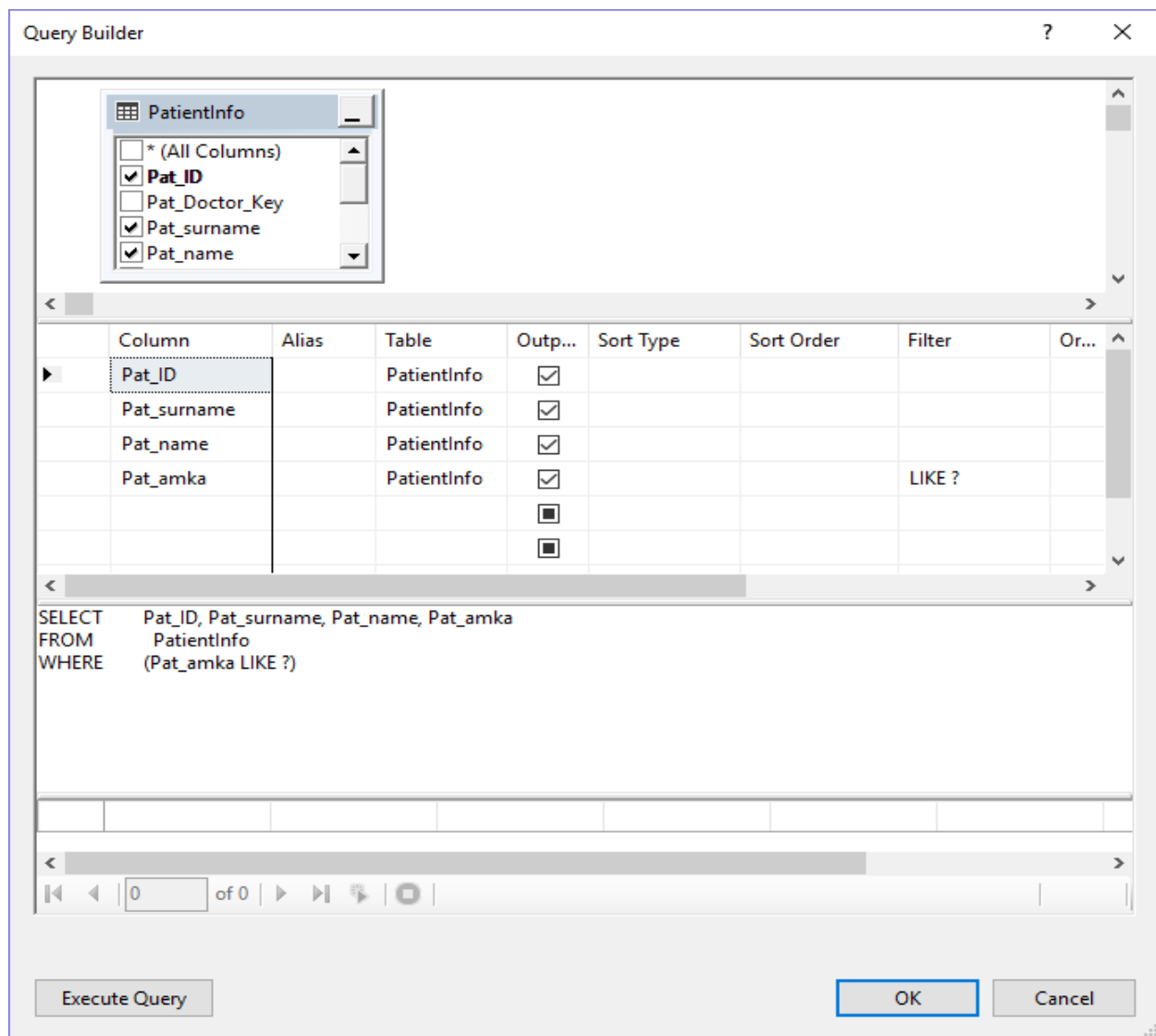
Εικόνα 41. Το Dataset με τους αρχικούς και παράγωγους πίνακες

Ενδιαφέρον παρουσιάζουν τα εξειδικευμένα queries που μπορούν να εφαρμοστούν στους πίνακες του Dataset. Δύο από αυτά εφαρμόζονται στον πίνακα PatientInfo και ονομάζονται FillByAmka και FillByName. Χρησιμοποιούν ώστε να επιστρέφουν τα αποτελέσματα αναζήτησης ασθενών βάσει του ΑΜΚΑ τους ή του επίθετού τους. Ο κώδικας Access SQL που έχει χρησιμοποιηθεί, φαίνεται στις αντίστοιχες εικόνες.



Εικόνα 42. Ο κώδικας του query "FillByAMKA"

Τα αντίστοιχα ερωτήματα είναι ευκολότερο να δημιουργηθούν με χρήση του κατασκευαστή ερωτημάτων, όπου τα διάφορα πεδία επιλέγονται και προστίθενται στο ερώτημα με το ποντίκι, ενώ οποιοσδήποτε επιπλέον κώδικας προστίθεται με πληκτρολόγηση. Είναι επίσης δυνατή η εκτέλεση του ερωτήματος ώστε να γίνει ο έλεγχος των επιστρεφόμενων αποτελεσμάτων.



Εικόνα 43. Το query στον Σχεδιαστή Ερωτημάτων

Specify a SQL SELECT statement

The SELECT statement will be used by the query.



Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the table?

What data should the table load?

```
SELECT Pat_ID, Pat_surname, Pat_name, Pat_amka
FROM PatientInfo
WHERE (Pat_surname LIKE ?)
```

Εικόνα 44. Το query αναζήτησης ασθενή βάσει επιθέτου

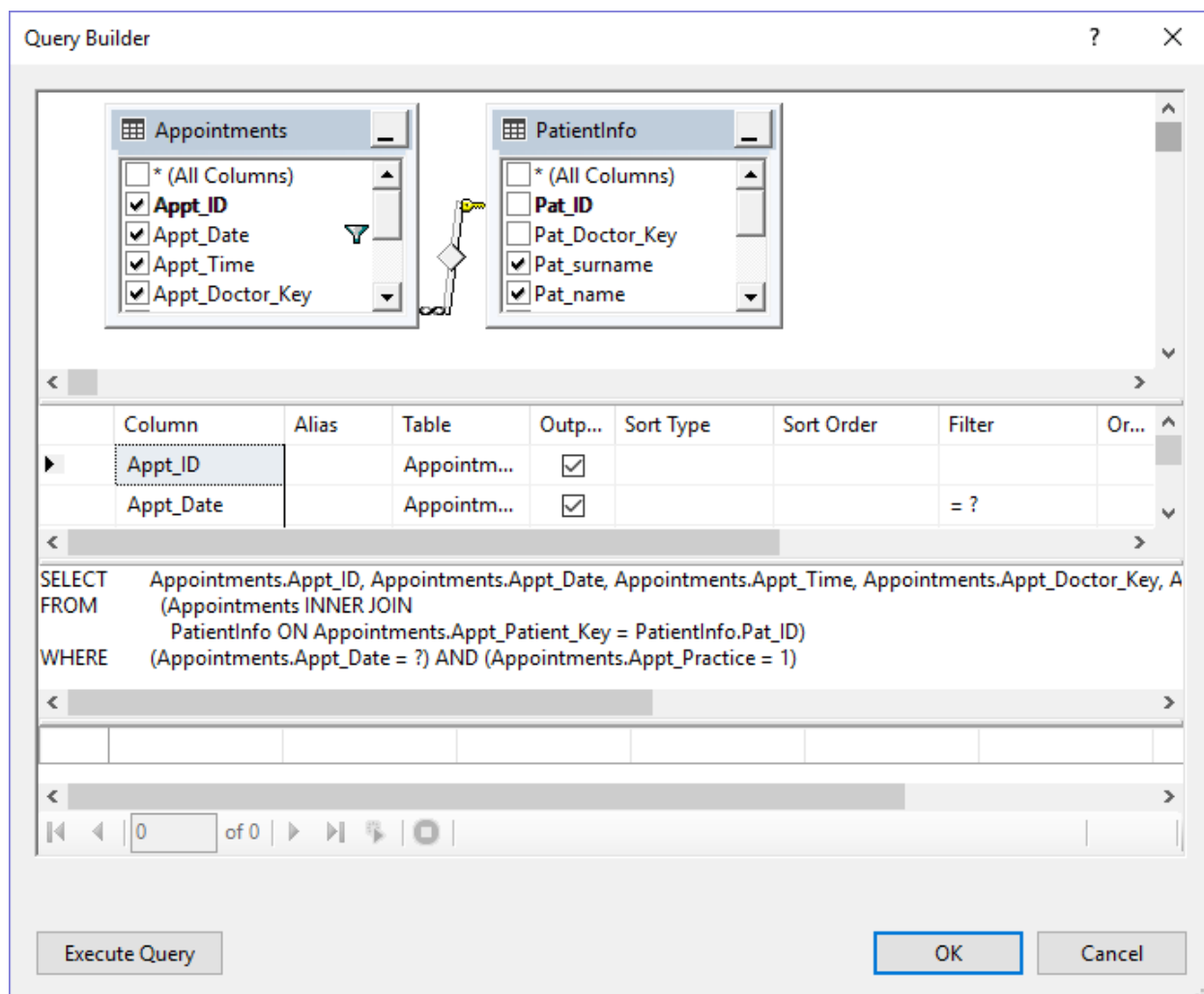
Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι παραγόμενοι πίνακες «ApptPractice», οι οποίοι δεν υπάρχουν στην αρχική βάση δεδομένων. Είναι πίνακες που τηρούν τα ραντεβού κάθε ιατρού, κάθε συγκεκριμένη ημέρα. Περιλαμβάνουν πεδία από τον πίνακα “PatientInfo”, δηλαδή το ονοματεπώνυμο του ασθενούς, αλλά και πεδία από τον πίνακα “Appointments”, όπως ημέρα, ώρα και διάρκεια ραντεβού. Οι αρχικοί πίνακες ενώνονται (JOIN) με τη βοήθεια των βασικών και εξωτερικών κλειδιών των πινάκων. Ο κώδικας που χρησιμοποιήθηκε φαίνεται στις αντίστοιχες εικόνες.

The screenshot shows the 'TableAdapter Query Configuration Wizard' window. The title bar reads 'TableAdapter Query Configuration Wizard'. The main heading is 'Specify a SQL SELECT statement'. Below it, a note states: 'The SELECT statement will be used by the query.' To the right of this note is a small icon of a document with an arrow pointing to it. The main instruction says: 'Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the table?'. Below this is a section titled 'What data should the table load?' which contains a text box with the following SQL query:

```
SELECT    Appointments.Appt_ID, Appointments.Appt_Date, Appointments.Appt_Time,
Appointments.Appt_Doctor_Key, Appointments.Appt_Patient_Key, Appointments.Appt_Todo,
PatientInfo.Pat_surname,
        PatientInfo.Pat_name
FROM      (Appointments INNER JOIN
        PatientInfo ON Appointments.Appt_Patient_Key = PatientInfo.Pat_ID)
WHERE     (Appointments.Appt_Date = ?) AND (Appointments.Appt_Practice = 1)
```

Below the text box is a button labeled 'Query Builder...'. At the bottom of the window are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Εικόνα 45. Ο κώδικας Access SQL που δημιουργεί τον πίνακα ApptPractice1



Εικόνα 46. Ο σχεδιαστής ερωτημάτων του αντίστοιχου πίνακα

Στο σχεδιαστή ερωτημάτων του πίνακα “ArptPractice” βλέπουμε τα επιλεγμένα πεδία των δύο αρχικών πινάκων καθώς και τη σχέση των κλειδιών που τους ενώνει. Με αυτόν τον τρόπο λαμβάνουμε συγχρονισμένα δεδομένα και από τους δύο πίνακες.

6. Συμπεράσματα - Συζήτηση

Η εφαρμογή της πληροφορικής στο χώρο της υγείας είναι από τις πλέον ευεργετικές - για την κοινωνία – πλευρές της τεχνολογικής προόδου. Με τη βοήθειά της, οι υπηρεσίες υγείας αποκτούν μεγαλύτερη ευελιξία, ταχύτερη οργάνωση και διεκπεραίωση διαδικασιών, μείωση ανθρώπινων σφαλμάτων αλλά και μοναδικές, αναντικατάστατες και πρωτοποριακές δυνατότητες.

Μια οδοντιατρική κλινική μπορεί να ωφεληθεί σε μεγάλο βαθμό από ένα πληροφοριακό σύστημα. Η οργάνωση χρόνου εργασιών και ο προγραμματισμός τους γίνεται άκοπα και με ακρίβεια, χωρίς λάθη και με ταχύτατη συνεργασία μεταξύ των εμπλεκόμενων μελών. Οι οδοντιατρικοί φάκελοι υγείας βρίσκονται ανά πάσα στιγμή στη διάθεση των θεραπόντων, εξοικονομώντας χρόνο, αυξάνοντας την ακρίβεια της πληροφόρησης και αποφεύγοντας τα ιατρικά λάθη λόγω κακής ενημέρωσης.

Το πληροφοριακό σύστημα οδοντιατρικής κλινικής που αναπτύχθηκε ως μέρος αυτής της εργασίας, πληροί όλες τις παραπάνω προϋποθέσεις. Πέραν αυτών, προσφέρει μοναδικές δυνατότητες μελέτης και επεξεργασίας ακτινογραφιών, μη επιτεύξιμες με μη πληροφοριακά μέσα. Με αυτόν τον τρόπο οι θεράποντες οδοντίατροι αποκτούν ένα σημαντικό όπλο στα διαγνωστικά τους εργαλεία, το οποίο βελτιώνει τη δυνατότητα μελέτης των ακτινογραφιών και επικουρεί σε καλύτερες και ακριβέστερες διαγνώσεις. Το όφελος για τους ασθενείς από αυτή τη διαδικασία είναι αναντίρρητο.

Για την ανάπτυξη του εν λόγω πληροφοριακού συστήματος επιλέχθηκε η πλατφόρμα των Windows, που είναι το de facto σύστημα πρώτης επιλογής για την οργάνωση ιατρείων. Η βάση δεδομένων της εφαρμογής αναπτύχθηκε σε Microsoft Access, ενώ η γλώσσα προγραμματισμού του συστήματος είναι η C#. Το Microsoft Visual Studio ήταν το περιβάλλον ανάπτυξης της εφαρμογής, ενώ κύριο ρόλο στη λειτουργία αυτής παίζουν οι τεχνολογίες .NET.

Συνολικά, το πληροφοριακό σύστημα που αναπτύχθηκε επιτυγχάνει τους βασικούς στόχους της δημιουργίας του, καταδεικνύοντας τα μεγάλα οφέλη που η πληροφορική προσφέρει στο χώρο της οδοντιατρικής υγείας.

7. Μελλοντική Εργασία

Η παρούσα εργασία μπορεί να αποτελέσει έναυσμα για τη δημιουργία και ολοκλήρωση ενός πληροφοριακού συστήματος κλινικής μεγαλύτερης κλίμακας, με περισσότερες και βελτιωμένες δυνατότητες. Οι πιθανές βελτιώσεις μπορούν πρώτα απ' όλα να εστιάσουν στις λειτουργίες που ήδη καλύπτονται από το παρόν σύστημα, προσθέτοντας δυνατότητες και βελτιώνοντας τις υπάρχουσες.

Κάποιες σχετικά απλές αλλά χρήσιμες δυνατότητες που θα μπορούσαν να προστεθούν, θα ήταν στην κατεύθυνση της ολοκλήρωσης και σύνδεσης (μέρους) του πληροφοριακού συστήματος με την καθημερινότητα όλων των εμπλεκόμενων μερών. Για τους ιατρούς, βοηθούς και προσωπικό της κλινικής θα ήταν χρήσιμο να έχουν πρόσβαση στο πρόγραμμα ημέρας που τους αφορά από το κινητό ή το διαδίκτυο. Για τους ασθενείς θα ήταν επίσης χρήσιμο να λαμβάνουν ειδοποιήσεις σχετικά με τα ραντεβού τους και τους επανελέγχους τους. Η επικοινωνία του θεράποντα με τον ασθενή με σκοπό την παρακολούθηση και την κινητοποίηση του τελευταίου, είναι ένα εργαλείο παροχής κινήτρων που συντελεί στη βελτιωμένη οδοντιατρική υγεία του ασθενούς. Ένα ενοποιημένο πληροφοριακό σύστημα θα μπορούσε να διευκολύνει την επικοινωνία αυτή.

Είναι ευκαίριο, όμως, να αναπτυχθούν δυνατότητες πέραν των συνηθισμένων, ακολουθώντας την αιχμή της τεχνολογικής και επιστημονικής προόδου. Μια τέτοια δυνατότητα θα ήταν η διευκόλυνση της απελευθέρωσης των πληροφοριών του ηλεκτρονικού φακέλου ασθενών, ακολουθώντας κοινά διεθνή πρότυπα κατά την αποθήκευση και την απόδοση αυτής της πληροφορίας στον ασθενή. Ο διαμοιρασμός του φακέλου μεταξύ φορέων υπηρεσιών υγείας θα ήταν επίσης ιδιαίτερα χρήσιμος.

Επίσης, μέσα από τη χρήση του συστήματος από μεγάλο αριθμό ασθενών, προκύπτει ένας μεγάλος όγκος δεδομένων οδοντιατρικής υγείας. Αν αυτά τα δεδομένα συνδυαστούν με ποιοτικά στοιχεία κατά τη λήψη του ιστορικού, μπορούν να χρησιμεύσουν σε μεγάλο βαθμό στην προαγωγή της επιστημονικής έρευνας. Ένα προχωρημένο πληροφοριακό σύστημα θα μπορούσε να φροντίσει όλα τα ζητήματα απορρήτου των πληροφοριών με την ανωνυμοποίησή τους και εξαγωγή για χρήση από

ερευνητικά ιδρύματα, δομές εξόρυξης δεδομένων big data και γενικά προσπάθειες που θα οδηγήσουν σε βελτίωση της πρόληψης και προόδου της επιστήμης.

Δεν είναι υπερβολή να πούμε ότι η πληροφορική έχει προχωρήσει από τον επικουρικό ρόλο στο χώρο της υγείας, αποτελώντας πλέον αναγκαίο και αναντικατάστατο μέρος της παροχής ποιοτικών υπηρεσιών υγείας.

8. Βιβλιογραφία

1. ΦΕΚ: Α 55 20090401. Κώδικας Οδοντιατρικής Δεοντολογίας. 1st ed.; 2009.
2. Codd EF. Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks. Research Report. IBM; 1969.
3. Codd EF. The relational model for database management. 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1990.
4. Armstrong DJ. The quarks of object-oriented development. Communications of the ACM. 2006 February.
5. Microsoft. Introduction to the C# Language and the .NET Framework. [Online].; 2017 [cited 2017 Μάιος 20. Available from: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>.
6. Microsoft. Visual Studio IDE. [Online].; 2017 [cited 2017 Μάιος 20. Available from: <https://docs.microsoft.com/en-us/visualstudio/ide/visual-studio-ide>.
7. Wikipedia..NET Framework. [Online].; 2017 [cited 2017 Μάιος 20. Available from: https://en.wikipedia.org/wiki/.NET_Framework.
8. Wikipedia. https://en.wikipedia.org/wiki/Windows_Forms. [Online].; 2017 [cited 2017 Μάιος 20. Available from: https://en.wikipedia.org/wiki/Windows_Forms.
9. Microsoft. ADO.NET. [Online].; 2017 [cited 2017 Μάιος 20. Available from: <https://msdn.microsoft.com/en-us/library/aa286484.aspx>.
10. Wikipedia. Microsoft Access. [Online].; 2017 [cited 2017 Μάιος 20. Available from: https://en.wikipedia.org/wiki/Microsoft_Access.

11. Microsoft. Overview of the .NET Framework. [Online].; 2017 [cited 2017 Μάιος 20]. Available from: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview>.

9. Ευρετήριο Εικόνων

Εικόνα 1 Σύγχρονη ψηφιακή πανοραμική ακτινογραφία γνάθων - ορθοπαντομογραφία	4
Εικόνα 2. Ψηφιακή αξονική τομογραφία CBCT με 3d ανασύσταση κρανίου	4
Εικόνα 3. Υπέρθεση 3d ανασύστασης προσωπικού κρανίου σε 3d μοντέλο προσώπου ασθενούς	5
Εικόνα 4 Το σχεσιακό μοντέλο δεδομένων κατά Κοντ	6
Εικόνα 5. Εισαγωγική Φόρμα.....	11
Εικόνα 6. Φόρμα «Ασθενείς»	12
Εικόνα 7. Φόρμα Οδοντιατρικού Ιστορικού.....	13
Εικόνα 8. Το παράθυρο εισαγωγής ακτινογραφιών - εικόνων	15
Εικόνα 9. Φόρμα Μελέτης - Επεξεργασίας Ακτινογραφίας	16
Εικόνα 10. Επεξεργασία Ακτινογραφίας - Αρνητικό.....	17
Εικόνα 11. Φόρμα Ραντεβού	18
Εικόνα 12. Προσθήκη Ραντεβού	19
Εικόνα 15. Ημερολόγιο σε αρχική μορφή	20
Εικόνα 13. Ημερολόγιο - επιλογή έτους.....	20
Εικόνα 14. Ημερολόγιο - επιλογή μήνα	20
Εικόνα 16. Επιλογή διάρκειας ραντεβού	21
Εικόνα 17. Επιλογή οδοντιάτρου.....	21
Εικόνα 18. Φόρμα Γενικού Ιστορικού Υγείας	22
Εικόνα 19. Το περιβάλλον ανάπτυξης του Visual Studio 2015	23
Εικόνα 20. Υποστηριζόμενες γλώσσες και είδη project από το Visual Studio IDE	24
Εικόνα 21. Επεξεργαστής κώδικα με δυνατότητες αυτόματης συμπλήρωσης	25
Εικόνα 22. Σχεδιαστής με δυνατότητες "Σύρε και άσε"	26
Εικόνα 23. .NET Framework in context	27
Εικόνα 24. The .NET Framework stack.....	28
Εικόνα 25. Παράδειγμα WinForms - Κουμπί "button1" και οι ιδιότητές του	29
Εικόνα 26. Η Microsoft Access 2016.....	30
Εικόνα 27. Ο πίνακας PatientInfo.....	33
Εικόνα 28. Ο πίνακας GeneralHealth.....	35
Εικόνα 29. Ο πίνακας Diagnosis	37

Εικόνα 30. Ο πίνακας Treatments	39
Εικόνα 31. Ο πίνακας Multimedia.....	40
Εικόνα 32. Ο πίνακας Doctors	41
Εικόνα 33. Ο πίνακας Appointments	43
Εικόνα 34. Οι σχέσεις μεταξύ πινάκων στη βάση δεδομένων.....	44
Εικόνα 35. Η εισαγωγική φόρμα.....	48
Εικόνα 36. Η φόρμα "ΑΣΘΕΝΕΙΣ"	52
Εικόνα 37. Η Φόρμα Γενικού Ιστορικού Υγείας.....	56
Εικόνα 38. Η φόρμα Οδοντιατρικής Εξέτασης.....	58
Εικόνα 39. Φόρμα Μελέτης και Επεξεργασίας Ακτινογραφιών	63
Εικόνα 40. Η φόρμα Ραντεβού	70
Εικόνα 41. Το Dataset με τους αρχικούς και παράγωγους πίνακες.....	74
Εικόνα 42. Ο κώδικας του query "FillByAMKA"	75
Εικόνα 43. Το query στον Σχεδιαστή Ερωτημάτων	76
Εικόνα 44. Το query αναζήτησης ασθενή βάσει επιθέτου.....	76
Εικόνα 45. Ο κώδικας Access SQL που δημιουργεί τον πίνακα ApptPractice1	77
Εικόνα 46. Ο σχεδιαστής ερωτημάτων του αντίστοιχου πίνακα.....	78

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΝΟΣΗΛΕΥΤΙΚΗΣ

ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΙΔΙΚΕΥΣΗ: ΠΛΗΡΟΦΟΡΙΚΗ ΥΓΕΙΑΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΟΔΟΝΤΙΑΤΡΙΚΗΣ ΚΛΙΝΙΚΗΣ ΜΕ
ΔΥΝΑΤΟΤΗΤΕΣ ΠΟΛΥΜΕΣΙΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**

ΜΑΚΡΗ Γ. ΧΡΗΣΤΟΥ

ΠΕΡΙΛΗΨΗ

Η πρόοδος της επιστήμης της πληροφορικής τις τελευταίες δεκαετίες, έχει συνεισφέρει σημαντικά στη βελτίωση των παρεχόμενων υπηρεσιών υγείας σε όλο το φάσμα αυτών. Τα οφέλη του ηλεκτρονικού φακέλου υγείας και της ηλεκτρονικής οργάνωσης μιας δομής υγείας είναι πολλαπλά και έχουν πλέον αποδειχθεί με βεβαιότητα. Πέραν των παραπάνω πεδίων, η πληροφορική υγείας μπορεί να προσφέρει νέες μεθόδους διάγνωσης για το ιατρικό προσωπικό, που θα οδηγήσουν σε βελτιωμένη θεραπευτική προσέγγιση των ασθενών. Με αυτόν τον τρόπο η επιστήμη των υπολογιστών συνδέεται άμεσα και σταθερά με την ιατρική πράξη, αποτελώντας σημαντικό εργαλείο στη διάθεση των ιατρών.

Η παρούσα διατριβή έχει στόχο τη δημιουργία ενός πληροφοριακού συστήματος για χρήση σε οδοντιατρική κλινική. Το εν λόγω πληροφοριακό σύστημα θα καλύψει τις ανάγκες τήρησης ηλεκτρονικού οδοντιατρικού φακέλου υγείας για κάθε ασθενή, με δυνατότητα αποθήκευσης των σχετικών πολυμεσικών αρχείων (ακτινογραφίες, εικόνες). Τα στοιχεία του φακέλου του ασθενούς χωρίζονται σε δημογραφικά, ιστορικό γενικής

υγείας και οδοντιατρικό ιστορικό – σχέδια θεραπείας. Κάθε επίσκεψη του ασθενή στο ιατρείο αποκτά μια καινούρια εγγραφή στο οδοντιατρικό ιστορικό, όπου γίνεται η καταγραφή των διαγνώσεων, των οδοντιατρικών πράξεων και των σχεδίων θεραπείας. Μαζί με το οδοντιατρικό ιστορικό, αποθηκεύονται και τα αρχεία φωτογραφιών και ακτινογραφιών του ασθενούς.

Εκτός από αποθήκευση, τα αρχεία φωτογραφιών και ιδίως οι ακτινογραφίες μπορούν να υποστούν εμφάνιση και επεξεργασία, με σκοπό την ακριβή και ορθή διάγνωση των οδοντιατρικών παθήσεων. Στην επεξεργασία αυτή περιλαμβάνονται λειτουργίες όπως μεγέθυνση και σμίκρυνση, αντιστροφή χρωμάτων (αρνητικό), μεταβολή φωτεινότητας, αντίθεσης και γάμμα (gamma) της ακτινογραφίας. Με αυτά τα χαρακτηριστικά, η εφαρμογή αποτελεί πλέον ισχυρό και αποτελεσματικό διαγνωστικό εργαλείο, με δυνατότητες που δεν μπορούν να υποκατασταθούν από άλλες μεθόδους. Τα εργαλεία διάγνωσης και επεξεργασίας ακτινογραφιών που ενσωματώνονται στην παρούσα εφαρμογή αποτελούν αναπόσπαστο κομμάτι αυτής και είναι στενά συνδεδεμένα με τη λειτουργία του ηλεκτρονικού φακέλου, υλοποιώντας άμεση και εύχρηστη αποθήκευση και επεξεργασία των πληροφοριών οδοντιατρικού φακέλου.

Το πληροφοριακό σύστημα θα καλύψει, τέλος, τις ανάγκες προγραμματισμού των εργασιών της κλινικής, τηρώντας το πρόγραμμα επεμβάσεων για πολλά ιατρεία - το κάθε ένα με οδοντιατρική έδρα - και για πολλούς οδοντιάτρους. Η διαχείριση προγράμματος εργασιών – ραντεβού είναι μια συνηθισμένη λειτουργία σε μια κλινική. Με τη χρήση της πληροφορικής, όμως, το σύνολο του ιατρικού και βοηθητικού προσωπικού έχει άμεση, ταυτόχρονη και από πολλά σημεία, πρόσβαση στο πρόγραμμα εργασιών, με αποτέλεσμα να αποφεύγονται λάθη και να επιτυγχάνεται ομαλή ροή λειτουργίας της κλινικής.

Το πληροφοριακό σύστημα είναι μια εφαρμογή για Windows, που βασίζεται σε βάση δεδομένων Microsoft Access και διεπαφή τεχνολογίας Microsoft .Net - Winforms. Η Access είναι κατεξοχήν εφαρμογή διαχείρισης βάσης δεδομένων της Microsoft, ενώ τα Windows είναι το de facto πρότυπο λειτουργικό σύστημα για χρήση γραφείου – ιατρείου. Ο προγραμματισμός της εφαρμογής είναι αντικειμενοστρεφής, και έγινε σε γλώσσα C#, σε περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio.

NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS
FACULTY OF NURSING

INTERUNIVERSITY POSTGRADUATE PROGRAM IN HEALTH CARE
MANAGEMENT AND HEALTH CARE INFORMATICS

DISSERTATION

DENTAL CLINIC INFORMATION SYSTEM
WITH MULTIMEDIA PROCESSING CAPABILITIES

BY CHRISTOS G. MAKRIS

SUMMARY

The last decades of computer science progress, have contributed to a substantial improvement to the whole spectrum of provided health services. The advantages of use of electronic health records and electronic organization in any health provider, have been well established and proven. Apart from these fields, health informatics may provide new diagnostics applications to the medical personnel, that lead to better therapeutic treatment of patients. Computer science, thus, is connected firmly and steadily to the medical act, and becomes an important tool in the hands of the medical community.

This dissertation aims at the creation of an information system for use in a dental clinic. The aforementioned system will cover the necessity of dental record keeping for each individual patient, with the added capability of storing relevant multimedia files (such as x-rays or pictures). Patient data comprise demographics, general health history and dental history – treatment plans. Every patient's visit to the clinic takes a new record in dental history, where diagnosis, dental treatments and treatment plans are recorded. Patient photos and x-rays are stored with the dental history records.

Apart from storage, picture files and radiographs may be presented and processed, in order to achieve a correct and exact diagnosis of dental ailments. This x-ray processing procedure includes functions like magnification or size reduction, color inversion (negative effect), and adjustment of brightness, contrast and image gamma. These abilities turn the application into a powerful and efficient diagnostic tool, with capabilities impossible to replace with traditional, non-electronic methods. The diagnostics and radiograph tools that are included in this application, are integral and closely connected to the dental health record. In this manner, image processing and dental record keeping is a unified and streamlined process.

Finally, this information system will also take on the role of scheduling the clinic's work, keeping patient appointments in the clinic, which comprises a number of dental chairs and doctors. Operations scheduling and patient appointments management are very common functions within an organized clinic. However, with the assistance of information technology, every person in the medical and assistance personnel has immediate, concurrent and direct access to the clinic's work schedule, resulting in error avoidance and clinic's work streamlining.

This information system is a Microsoft Windows application, based on a Microsoft Access database and a Microsoft .Net – Winforms interface. Microsoft Access is a very common database management system, while the Windows operating system is the de facto standard in office – clinic environments. The development of the application is object oriented and was done in the C# programming language, making use of the Microsoft Visual Studio IDE.