



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Ταυτοποίηση ασθενούς μέσω ραδιοσυχνικής αναγνώρισης
(RFID) με σκοπό τον έλεγχο της χορήγησης αναισθησίας
κατά τη διάρκεια χειρουργικής επέμβασης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αρχοντία Δ. Τσιαμπαλή

Επιβλέπων : Δημήτριος-Διονύσιος Κουτσούρης
Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Ταυτοποίηση ασθενούς μέσω ραδιοσυχνικής αναγνώρισης
(RFID) με σκοπό τον έλεγχο της χορήγησης αναισθησίας
κατά τη διάρκεια χειρουργικής επέμβασης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αρχοντία Δ. Τσιαμπαλή

Επιβλέπων : Δημήτριος-Διονύσιος Κουτσούρης
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12^η Σεπτεμβρίου 2014.

.....
Δ.Δ. Κουτσούρης
Καθηγητής Ε.Μ.Π.

.....
Κ. Νικήτα
Καθηγήτρια Ε.Μ.Π.

.....
Γ. Ματσόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2014

.....
Αρχοντία Δ. Τσιαμπαλή

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αρχοντία Τσιαμπαλή, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας διπλωματικής είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος ελέγχου της χορήγησης μειγμάτων ενδοφλέβιων αναισθητικών φαρμάκων κατά τη διάρκεια χειρουργικών επεμβάσεων για την πρόληψη ανθρώπινων σφαλμάτων.

Σε πρώτο βήμα, παρουσιάζονται οι τεχνικές αναισθησίας και τα αναισθητικά φάρμακα που χρησιμοποιούνται στη χορήγηση γενικής αναισθησίας. Ιδιαίτερη έμφαση δίνεται στην τεχνική της γενικής αναισθησίας καθώς και στην περιγραφή και στις δοσολογίες των ενδοφλέβιων αναισθητικών φαρμάκων. Στη συνέχεια, αφού αναφερθούν οι κυριότερες επιπλοκές που μπορεί να προκύψουν κατά τη χορήγηση γενικής αναισθησίας, παρουσιάζονται τα αποτελέσματα που προέκυψαν από μεγάλες έρευνες που διεξήχθησαν σε διάφορες χώρες σχετικά με τα αίτια των επιπλοκών.

Κατόπιν, γίνεται μια εκτενής περιγραφή της τεχνολογίας ραδιοσυχνικής αναγνώρισης, RFID, και της πλατφόρμας Arduino UNO που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Μελετώνται οι αρχές λειτουργίας και η δομή των συστημάτων RFID, τα χαρακτηριστικά και οι διάφορες κατηγορίες των συστατικών μερών τους, διάφορες εφαρμογές της τεχνολογίας RFID καθώς και τα χαρακτηριστικά της πλατφόρμας Arduino UNO.

Επιπλέον, παρουσιάζονται διεξοδικά και τα υπόλοιπα δομικά στοιχεία του συστήματος που υλοποιήθηκε. Συγκεκριμένα, γίνεται μια περιγραφή του SOAP Server, της βάσης δεδομένων με τα στοιχεία των ασθενών και των μειγμάτων, της διαδραστικής διεπαφής που χειρίζεται ο αναισθησιολόγος καθώς και τον ρόλο τους στην ορθή λειτουργία του συστήματος.

Τέλος, παρουσιάζεται μια υλοποίηση του συστήματος ελέγχου χορήγησης μειγμάτων για 20 ασθενείς που υποβάλλονται σε ωτορινολαρυγγολογικές επεμβάσεις με διαφορετικά χαρακτηριστικά (ηλικία, βάρος, ζωτικά σημεία) και εξάγονται τα σχετικά συμπεράσματα.

Λέξεις-Κλειδιά

Έλεγχος χορήγησης γενικής αναισθησίας, ταυτοποίηση ασθενών, πλατφόρμα Arduino, Ραδιοσυχνική Αναγνώριση, SOAP Server

ABSTRACT

The purpose of this thesis is the design and implementation of a controlled administration system of intravenous anesthetic mixtures during surgery to prevent human error.

In a first step, the techniques of anesthesia and anesthetic drugs used in general anesthesia are presented. Particular emphasis is given to the technique of general anesthesia and in the specification and dosages of intravenous anesthetics. Then, after mentioning the main complications that can arise during the administration of general anesthesia, the results from large surveys conducted in different countries which study the causes of complications are presented.

Subsequently, there is a comprehensive description of Radio Frequency Identification technology, RFID, and platform Arduino UNO which are used in the implementation of the system. The principles and the structure of the RFID systems, the characteristics and the various types of components, various applications of RFID technology and the characteristics of the platform Arduino UNO are studied.

Moreover, the rest components of the system which was implemented are presented in detail. Specifically, the SOAP Server, the database with patient data and mixtures, the interactive interface that the anesthetist handles and their role in the correct function of the system are described.

Finally, an implementation of the controlled administration system of mixtures for 20 patients undergoing surgery with different characteristics (age, weight, vital signs) is presented and the respective conclusions are reached.

Keywords

Controlled administration of general anesthesia, patient identification, Arduino platform, RFID, SOAP Server

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στο Εργαστήριο Βιοϊατρικής Τεχνολογίας της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, υπό την επίβλεψη του Καθηγητή κ. Δημητρίου Κουτσούρη.

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Δημήτριο Κουτσούρη για την ανάθεση ενός τόσο ενδιαφέροντος θέματος και για την καθοδήγηση του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στον κ. Χαράλαμπο Τσίρμπα, υποψήφιο διδάκτορα της σχολής ΗΜΜΥ, για την πολύτιμη βοήθεια του σε κάθε βήμα της εργασίας και τη συμβολή του στην ολοκλήρωση της. Παράλληλα, θα ήθελα να απευθύνω τις ευχαριστίες μου στο προσωπικό του Αναισθησιολογικού Τμήματος του Γενικού Νοσοκομείου Αθηνών και ειδικότερα στην αναισθησιολόγο κ. Άρτεμις Παπαδήμα για τις πολύτιμες επιστημονικές συμβουλές και εύστοχες παρατηρήσεις τους.

Θα ήθελα επίσης να ευχαριστήσω όλους τους φίλους μου για τη συμπαράσταση και την ηθική υποστήριξή τους καθ' όλη τη διάρκεια των σπουδών μου. Ειδικότερα, για την ολοκλήρωση της διπλωματικής εργασίας είμαι ευγνώμων στον Γεώργιο Χατζόπουλο, ο οποίος με στήριξε έμπρακτα με τον δικό του τρόπο.

Τέλος, ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω προς την οικογένειά μου και ιδιαιτέρως στους γονείς μου για την διαχρονική συμπαράστασή τους, την υλική και ηθική στήριξη των επιλογών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	5
Abstract.....	7
Ευχαριστίες.....	9
Ευρετήριο Σχημάτων.....	13
Ευρετήριο Πινάκων.....	14
Εισαγωγή.....	15

1. Αναισθησία

1.1 Γενικά.....	17
1.2 Τεχνικές αναισθησίας.....	18
1.2.1 Γενική αναισθησία.....	18
1.2.2 Περιοχική αναισθησία.....	18
1.2.3 Τοπική αναισθησία.....	18
1.2.4 Καταστολή.....	19
1.2.5 Συνδυασμένη αναισθησία.....	19
1.3 Αναισθητικά φάρμακα στη γενική αναισθησία.....	19
1.3.1 Εισπνεόμενα.....	19
1.3.2 Ενδοφλέβια.....	19
1.4 Επιπλοκές κατά τη γενική αναισθησία.....	23

2. Η Τεχνολογία RFID

2.1 Γενικά.....	29
2.2 Ιστορική αναδρομή.....	29
2.3 Δομή και αρχές λειτουργίες των συστημάτων RFID.....	30
2.4 Συστατικά μέρη ενός RFID συστήματος.....	31
2.4.1 Ετικέτες.....	31
2.4.1.1 Χαρακτηριστικά.....	31
2.4.1.2 Κατηγορίες ετικετών.....	32
2.4.2 Αναγνώστης.....	38
2.4.2.1 Χαρακτηριστικά.....	38
2.4.2.2 Κατηγοριοποίηση.....	41
2.4.3 Ενδιάμεσο Λογισμικό.....	42
2.5 Εφαρμογές της τεχνολογίας RFID.....	43

3. Arduino

3.1 Γενικά για το Arduino UNO.....	47
3.2 Τεχνικά χαρακτηριστικά του Arduino UNO.....	47
3.3 Φυσικά χαρακτηριστικά του Arduino UNO.....	48
3.4 Τροφοδοσία.....	48
3.5 Μνήμη.....	49
3.6 Ακροδέκτες.....	49
3.7 Προστασία.....	50
3.8 Επικοινωνία.....	50
3.9 Μοντέλα Arduino.....	51
3.10 Shields.....	53
3.11 Ethernet Shield.....	54

4. Σχεδιασμός και υλοποίηση του συστήματος

4.1 Γενική περιγραφή του συστήματος.....	57
4.2 Στοιχεία του συστήματος.....	57
4.3 Αρχιτεκτονική του συστήματος.....	63
4.4 Παράδειγμα χρήσης.....	67
4.5 Συμπεράσματα.....	73

Παράρτημα Α.....	75
Παράρτημα Β.....	77
Παράρτημα Γ.....	79
Παράρτημα Δ.....	81
Παράρτημα Ε.....	85
Βιβλιογραφία.....	97

Ευρετήριο σχημάτων

Σχήμα 1.1: Συσκευές παρακολούθησης ζωτικών λειτουργιών του ασθενή (monitoring) και αντλίες.....	17
Σχήμα 2.1: Ιστορική αναδρομή RFID τεχνολογίας.....	30
Σχήμα 2.2: Ιστορική αναδρομή RFID τεχνολογίας.....	30
Σχήμα 2.3: Μοντέλο master-slave.....	31
Σχήμα 2.4: Ετικέτα.....	31
Σχήμα 2.5: Παθητική ετικέτα.....	32
Σχήμα 2.6: Ενεργητικές ετικέτες.....	32
Σχήμα 2.7: Ετικέτες στις διάφορες συχνότητες λειτουργίας.....	35
Σχήμα 2.8: Smart label.....	36
Σχήμα 2.9: Δίσκοι.....	36
Σχήμα 2.10: Γυάλινοι σωλήνες.....	37
Σχήμα 2.11: Ear tag.....	37
Σχήμα 2.12: Collar tag.....	37
Σχήμα 2.13: Ceramic tags.....	37
Σχήμα 2.14: Αναγνώστης (στο αριστερό μέρος διακρίνεται η μονάδα ελέγχου ενώ στο δεξιό η μονάδα υψηλών συχνοτήτων).....	39
Σχήμα 2.15: Μπλοκ διάγραμμα ενός αναγνώστη.....	39
Σχήμα 2.16: Μονάδα ελέγχου.....	40
Σχήμα 2.17: Φορητοί αναγνώστες.....	41
Σχήμα 2.18: Σταθερός αναγνώστης.....	42
Σχήμα 2.19: Ολοκληρωμένοι αναγνώστες.....	42
Σχήμα 2.20: Ρόλος του ενδιάμεσου λογισμικού σε ένα RFID σύστημα.....	43
Σχήμα 2.21: Εφαρμογή της τεχνολογίας RFID σε μια βιβλιοθήκη.....	43
Σχήμα 2.22: Ηλεκτρονικό διαβατήριο.....	44
Σχήμα 2.23: Έλεγχος πρόσβασης με χρήση της τεχνολογίας RFID.....	45
Σχήμα 3.1: Arduino UNO.....	47
Σχήμα 3.2: Μοντέλα Arduino.....	52
Σχήμα 3.3: Arduino WiFi Shield.....	53
Σχήμα 3.4: Arduino Ethernet Shield.....	53
Σχήμα 3.5: Wireless SD Shield.....	53
Σχήμα 3.6: Wireless Proto Shield.....	54
Σχήμα 3.7: Motor Shield.....	54
Σχήμα 3.8: GSM Shield.....	54
Σχήμα 4.1: SOAP Envelope.....	58
Σχήμα 4.2: Σχήμα βάσης δεδομένων.....	60
Σχήμα 4.3: RDM630 Reader.....	61
Σχήμα 4.4: Συνδεσμολογία Arduino-Ethernet Shield-Rfid Reader.....	62
Σχήμα 4.5: Model View ViewModel.....	63
Σχήμα 4.6: Αρχιτεκτονική συστήματος.....	64
Σχήμα 4.7: Αρχικό μήνυμα οθόνης.....	66
Σχήμα 4.8: Εισαγωγή αριθμού μητρώου ασθενούς.....	66
Σχήμα 4.9: Προβολή στοιχείων ασθενούς και μείγματος φαρμάκου που του αντιστοιχεί.....	67
Σχήμα 4.10: Μήνυμα σε περίπτωση επιλογής σωστού μείγματος.....	69
Σχήμα 4.11: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος.....	70
Σχήμα 4.12: Εισαγωγή αριθμού μητρώου ασθενούς.....	70

Σχήμα 4.13: Προβολή στοιχείων ασθενούς και μείγματος που του αντιστοιχεί.....	71
Σχήμα 4.14: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος.....	71
Σχήμα 4.15: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος.....	72
Σχήμα 4.16: Μήνυμα σε περίπτωση επιλογής σωστού μείγματος.....	72
Σχήμα 4.17: Μήνυμα σε περίπτωση εισαγωγής Α.Μ που δεν αντιστοιχεί σε κάποιον ασθενή.....	73

Ευρετήριο Πινάκων

Πίνακας 2.1: Χαρακτηριστικά ετικετών με βάση την πηγή ενέργειας τους.....	33
Πίνακας 2.2: Χαρακτηριστικά ετικετών ανά κλάση.....	38
Πίνακας 3.1: Τεχνικά χαρακτηριστικά Arduino UNO.....	48
Πίνακας 3.2: Χαρακτηριστικά διάφορων μοντέλων Arduino.....	51
Πίνακας 4.1: Χαρακτηριστικά RDM630 Reader.....	61
Πίνακας 4.2: Δοσολογία ενδοφλέβιων αναισθητικών φαρμάκων.....	65

ΕΙΣΑΓΩΓΗ

Τα ιατρικά λάθη, συμπεριλαμβανομένων αυτών που γίνονται κατά τη γενική αναισθησία, αποτελούν μία από τις 10 κυριότερες αιτίες θανάτου παγκοσμίως. Συγκεκριμένα, τη δεκαετία του 1970, η πιθανότητα σοβαρής αναισθησιολογικής επιπλοκής π.χ. θάνατος, μόνιμη εγκεφαλική βλάβη ήταν περίπου 1 στις 10.000 χειρουργικές επεμβάσεις, ενώ στις μέρες μας περίπου 1 στις 250.000.

Μιας και η αναισθησιολογία δεν ανήκει στην ευρύτερη κατηγορία των θεραπευτικών μεθόδων, είναι η μόνη ιατρική ειδικότητα στην οποία απαιτείται 100% επιτυχία. Για να γίνει εφικτό όμως αυτό, θα πρέπει να αναζητηθούν και να μελετηθούν τα αίτια που οδηγούν σε λάθη κατά τη χορήγηση και τη διατήρηση της αναισθησίας.

Σύμφωνα με εκτενείς έρευνες που έχουν διεξαχθεί, το 87% των λαθών οφείλονται στον ανθρώπινο παράγοντα και σε ανεπάρκειες του συστήματος υγείας της κάθε χώρας. Οι περισσότεροι αναισθησιολόγοι παραδέχονται ότι έχουν κάνει τουλάχιστον ένα λάθος στη χορήγηση ενδοφλέβιων αναισθητικών φαρμάκων στην επαγγελματική τους καριέρα (λάθος φάρμακο, λανθασμένη δόση και λανθασμένη οδός χορήγησης). Ενώ οι συνέπειες αυτών των λαθών μπορεί να είναι αμελητέες και ο ασθενής να μην υποστεί καμία σοβαρή επιπλοκή, κάποια λάθη μπορούν να αποβούν μοιραία και να οδηγήσουν ακόμα και στο θάνατο.

Τα τελευταία χρόνια, έχουν αναπτυχθεί διάφορα συστήματα και τεχνικές για την πρόληψη των επιπλοκών κατά την αναισθησία. Η τεχνολογική πρόοδος οδήγησε σε νέα φάρμακα που προκαλούν λιγότερες επιπλοκές και σε σύγχρονες τεχνικές παρακολούθησης των ασθενών που υποκαθιστούν τις αισθήσεις του αναισθησιολόγου. Έτσι παρατηρείται μια μείωση των επιπλοκών αλλά όχι των λαθών που αφορούν τη λάθος χορήγηση φαρμάκων. Τα συστήματα που έχουν προταθεί για την ελάττωση των συγκεκριμένων λαθών είναι ελάχιστα ενώ τα αποτελέσματα της εφαρμογής τους δεν ήταν τα αναμενόμενα καθώς δεν περιόριζαν επαρκώς τον ανθρώπινο παράγοντα στην επιλογή φαρμάκων και παρουσίαζαν προβλήματα χρηστικότητας.

Στην παρούσα διπλωματική παρουσιάζεται ένα σύστημα ελέγχου χορήγησης μειγμάτων ενδοφλέβιων φαρμάκων κατά τη γενική αναισθησία κάνοντας χρήση της τεχνολογίας ραδιοσυχνικής αναγνώρισης (RFID) και ενός υπολογιστικού συστήματος που βασίζεται στην πλατφόρμα Arduino UNO.

Η διπλωματική εργασία αναπτύσσεται σε 4 ενότητες. Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στην αναισθησία και στις τεχνικές της, μια λεπτομερής περιγραφή των αναισθητικών φαρμάκων και των επιπλοκών της αναισθησίας, καθώς και μια παρουσίαση διάφορων ερευνών που διεξήχθησαν σε διάφορες χώρες σχετικά με τις επιπλοκές και τα αίτια τους. Στο δεύτερο κεφάλαιο γίνεται μια διεξοδική ανάλυση της τεχνολογίας RFID, των αρχών λειτουργίας της και των συστατικών μερών ενός RFID συστήματος ενώ στο τρίτο κεφάλαιο παρουσιάζεται εκτενώς η πλατφόρμα Arduino UNO. Τέλος στο τέταρτο κεφάλαιο περιγράφεται διεξοδικά ο σχεδιασμός και η υλοποίηση του συστήματος ελέγχου χορήγησης μειγμάτων αναισθητικών φαρμάκων και εξάγονται τα σχετικά συμπεράσματα.

1. ΑΝΑΙΣΘΗΣΙΑ

1.1 Γενικά

Με τον όρο αναισθησία αναφερόμαστε στην προσωρινή καταστολή της αίσθησης του πόνου και της μνήμης, που επέρχεται με διάφορα φάρμακα, με σκοπό την πραγματοποίηση χειρουργικών επεμβάσεων ή διαγνωστικών διεργασιών. Η χορήγηση των φαρμάκων, που γίνεται ενδοφλέβια ή/και μέσω αναπνευστικής οδού, πραγματοποιείται από εξειδικευμένο αναισθησιολόγο.

Η αναισθησία πρωτοεμφανίστηκε στην αρχαιότητα όταν οι αρχαίοι λαοί χρησιμοποιούσαν το χυμό από φύλλα παπαρούνας (το όπιο) [1] και τις ρίζες του Μανδραγόρα για την περάτωση χειρουργικών επεμβάσεων. Παρόλο που υπάρχουν αναφορές για εφαρμογή κάποιας μορφής αναισθησίας στους επόμενους αιώνες, η ανάπτυξη της ουσιαστικά ξεκίνησε στα μέσα του 19^{ου} αιώνα. Σταθμός στην εξέλιξη της αναισθησίας θεωρείται η 16^η Οκτωβρίου του 1846, όταν ένας οδοντίατρος, ο William Thomas Green Morton, έκανε δημόσια επίδειξη αναισθησίας με αιθέρα κατά τη διάρκεια χειρουργικής επέμβασης στο Γενικό Νοσοκομείο της Μασαχουσέτης [2]. Ύστερα από αυτό το περιστατικό, αρχίζει μια σημαντική πρόοδος της αναισθησίας με τον αιθέρα, το χλωροφόρμιο καθώς και το πρωτοξείδιο του αζώτου να πρωτοστατούν. Κατά τον εικοστό αιώνα εισάγονται νέα φάρμακα και συσκευές παρακολούθησης των ζωτικών λειτουργιών των ασθενών (monitoring), καθιστώντας την αναισθησία πιο ασφαλή και δίνοντας έτσι τη δυνατότητα πραγματοποίησης πολύ δύσκολων και επώδυνων χειρουργικών επεμβάσεων.



Σχήμα 1.1: Συσκευές παρακολούθησης ζωτικών λειτουργιών του ασθενή (monitoring) και αντλίες

1.2 Τεχνικές αναισθησίας

Οι τεχνικές αναισθησίας είναι οι εξής:

1.2.1 Γενική αναισθησία

Η τεχνική αυτή χρησιμοποιείται για επεμβάσεις στις οποίες ο ασθενής πρέπει να αναισθητοποιείται πλήρως. Ο ασθενής δηλαδή χάνει όλες τις αισθήσεις του και όλα τα όργανα του βρίσκονται σε καταστολή.

Οι φάσεις της γενικής αναισθησίας είναι οι εξής:

- Εισαγωγή
- Συντήρηση
- Ανάνηψη

Για την εισαγωγή του ασθενούς στη γενική αναισθησία χορηγούνται συνήθως 3 κατηγορίες φαρμάκων, ένα υπνωτικό, ένα αναλγητικό και ένα μυοχαλαρωτικό. Πραγματοποιείται απώλεια της συνείδησης και καταστολή της αναπνοής. Κατόπιν εξασφαλίζεται ο αεραγωγός του ασθενούς, με την τοποθέτηση ενδοτραχειακού σωλήνα, ή εναλλακτικά λαρυγγικής μάσκας, και συνδέεται με το αναισθησιολογικό μηχάνημα/αναπνευστήρα, οπότε και εφαρμόζεται μηχανικός αερισμός.

Στη φάση της συντήρησης, διατηρείται η αναισθησία συνήθως με τη χορήγηση πτητικού αερίου και γίνεται συμπληρωματική χορήγηση αναλγησίας και μυοχαλαρωτικού φαρμάκου ανάλογα με τις απαιτήσεις της κάθε χειρουργικής επέμβασης.

Στο τέλος της επέμβασης, γίνεται διακοπή της χορήγησης των φαρμάκων και ο ασθενής αφυπνίζεται. Οδηγείται στην αίθουσα ανάνηψης και παραμένει εκεί έως ότου βεβαιωθεί ο αναισθησιολόγος ότι μπορεί να μεταφερθεί στο θάλαμο με ασφάλεια.

1.2.2 Περιοχική αναισθησία

Σε αντίθεση με τη γενική αναισθησία, στην περιοχική ο ασθενής παραμένει ξύπνιος (δηλαδή διατηρεί την αυτόματη αναπνοή) κατά τη διάρκεια της επέμβασης. Τα αναισθητικά φάρμακα εγχέονται με ένεση στην περιοχή από όπου περνάει το δίκτυο νεύρων που νευρώνουν την περιοχή στην οποία γίνεται η χειρουργική επέμβαση. Έτσι αναισθητοποιείται μόνο η συγκεκριμένη περιοχή χωρίς να επηρεάζεται η καρδιά και ο εγκέφαλος. Παραδείγματα περιοχικής αναισθησίας είναι η επισκληρίδιος και η ραχιαία αναισθησία όπου αδρανοποιούνται τα νεύρα από τη μέση και κάτω και εφαρμόζονται κυρίως σε γυναικολογικές, μαιευτικές και ουρολογικές επεμβάσεις.

1.2.3 Τοπική αναισθησία

Η τοπική αναισθησία χρησιμοποιείται για να αναισθητοποιηθεί μόνο ένα μικρό τμήμα του σώματος. Εφαρμόζεται σε μικροεπεμβάσεις όπως βιοψίες δέρματος και οδοντιατρικές επεμβάσεις και έχει μικρή διάρκεια. Τα τοπικά αναισθητικά χορηγούνται με ένεση αλλά κυκλοφορούν και σε άλλες μορφές, όπως σπρέι και αλοιφές.

1.2.4 Καταστολή

Στην καταστολή (αναφέρεται και ως αναισθησία με μέθη), χορηγούνται φάρμακα εισπνεόμενα, ενδοφλέβια ή από το στόμα, που ηρεμούν τον ασθενή και τον κάνουν να νιώσει νυσταγμένο παρόλο που παραμένει ξύπνιος. Χρησιμοποιείται σε πολύ σύντομες επεμβάσεις και περιλαμβάνει παυσίπονο που χορηγείται ενδοφλέβια και ένα ήπιο υπνωτικό.

1.2.5 Συνδυασμένη αναισθησία

Στη συνδυασμένη αναισθησία συνδυάζονται 2 από τις παραπάνω τεχνικές κυρίως η γενική και η περιοχική.

1.3 Αναισθητικά φάρμακα στη γενική αναισθησία

Τα αναισθητικά φάρμακα χωρίζονται αναλόγως τον τρόπο χορήγησής τους σε 2 μεγάλες κατηγορίες: τα εισπνεόμενα και τα ενδοφλέβια.

1.3.1 Εισπνεόμενα

Τα εισπνεόμενα αναισθητικά είναι είτε αέρια είτε πτητικά. Τα αέρια υγροποιούνται υπό πίεση και αποθηκεύονται σε οβίδες από τις οποίες εξέρχονται σε αέρια μορφή, ενώ τα πτητικά είναι υγρά και εξατμίζονται με τη βοήθεια εξατμιστήρων [3]. Επίσης έχουν αναλγητική, υπνωτική και μυοχαλαρωτική δράση.

Το αέριο αναισθητικό που χρησιμοποιείται στις μέρες μας είναι το πρωτοξείδιο (ή υποξείδιο) του αζώτου (N_2O). Έχει ήπιες αναλγητικές ιδιότητες, δεν είναι ισχυρό αναισθητικό και για αυτό το λόγο χρειάζεται να συμπληρώνεται από άλλα ενδοφλέβια ή πτητικά αναισθητικά.

Τα πτητικά αναισθητικά που χρησιμοποιούνται ευρέως στις χειρουργικές επεμβάσεις είναι το ισοφλουράνιο, το δεσφλουράνιο, το αλοθάνιο και το σεβοφλουράνιο.

1.3.2 Ενδοφλέβια

Η κύρια δράση των ενδοφλέβιων αναισθητικών είναι η πρόκληση απώλειας της συνείδησης και αμνησίας. Χορηγούνται σε εφάπαξ δόσεις (bolus), σε επαναληπτικές δόσεις ή σε συνεχή έγχυση. Ακολουθεί μια αναφορά στα ενδοφλέβια αναισθητικά που χρησιμοποιούν σήμερα οι αναισθησιολόγοι καθώς και στη δοσολογία τους σύμφωνα με το Ινστιτούτο Φαρμακευτικής Έρευνας και Τεχνολογίας (ΙΦΕΤ) [4].

- *Προποφόλη*: Η προποφόλη είναι βραχείας δράσης αναισθητικό με κύρια χαρακτηριστικά την ταχεία έναρξη και γρήγορη αφύπνιση. Συνυπολογίζοντας και την αντιεμετική της δράση, είναι το φάρμακο επιλογής σε βραχείας

διάρκειας χειρουργικές επεμβάσεις, για εισαγωγή και συντήρηση της αναισθησίας και καταστολή στη Μονάδα Εντατικής Θεραπείας (ΜΕΘ). Δεν ενδείκνυται σε εγκύους και ασθενείς που έχουν αλλεργία στο φάρμακο.
Δοσολογία:

Εισαγωγή στην αναισθησία	2-3 mg/kg αναλόγως της ηλικίας και των απαιτήσεων της επέμβασης σε ρυθμό έγχυσης 4 ml/10sec ή 2 ml/10 sec αναλόγως της κατάστασης του ασθενούς
Συντήρηση της αναισθησίας	<ul style="list-style-type: none"> • Διαλείπουσα χορήγηση 20-25 mg • Συνεχής έγχυση με ρυθμό 0.1-0.2 mg/kg/min ή 5-10mg/kg/h
Καταστολή στις ΜΕΘ	Συνεχής έγχυση με ρυθμό 0.3-4 mg/kg/h

- *Κεταμίνη*: Η Κεταμίνη προκαλεί διαχωριστική αναισθησία, δηλαδή ενώ ο ασθενής φαίνεται να είναι σε εγρήγορση με ανοιχτά μάτια και νυσταγμικό βλέμμα, δεν ανταποκρίνεται σε αισθητικά ερεθίσματα [5]. Επίσης προκαλεί αμνησία και αναλγησία ενώ στην αφύπνιση ο ασθενής έχει σύγχυση και παραισθήσεις. Αποτελεί το βασικό αναισθητικό για εισαγωγή και διατήρηση αναισθησίας σε ασθενείς με καρδιαγγειακή καταστολή, με άσθμα και με αιμορραγικό shock. Επιπλέον ενδείκνυται για επανειλημμένες χορηγήσεις αναισθησίας ειδικά στα παιδιά και για αναισθησία εκτός χειρουργείου.
Δοσολογία:

Εισαγωγή στην αναισθησία	1-2 mg/kg ενδοφλεβίως ή 4-5 mg/kg ενδομυϊκώς
Συντήρηση της αναισθησίας	1/3 της αρχικής δόσης

- *Ετομιδάτη*: Η Ετομιδάτη είναι αναισθητικό με ταχεία έναρξη δράσης και γρήγορη αφύπνιση με καρδιαγγειακή σταθερότητα και μη έκλυση ισταμίνης [3]. Χρησιμοποιείται για εισαγωγή στην αναισθησία και ειδικότερα σε ασθενείς με βαριά γενική κατάσταση και ασταθές καρδιαγγειακό σύστημα. Δοσολογία:

Εισαγωγή στην αναισθησία	0.2-0.3 mg/kg ενδοφλεβίως ανάλογα με την ηλικία και τη γενική κατάσταση του ασθενούς
--------------------------	--

- *Θειοπεντάλη*: Η Θειοπεντάλη ή Θειοπεντόνη είναι ένα ισχυρό, με ταχεία έναρξη δράσης ενδοφλέβιο αναισθητικό. Χρησιμοποιείται για εισαγωγή την αναισθησία αλλά και για τον έλεγχο σπασμών όπως στην επιληψία. Δοσολογία:

Εισαγωγή στην αναισθησία	3-5 mg/kg ενδοφλεβίως αναλόγως της ηλικίας, του φύλου, του σωματικού βάρους και της γενικής κατάστασης του ασθενούς
Έλεγχος σπασμών	1-3 mg/kg

- *Βενζοδιαζεπίνες*: Οι βενζοδιαζεπίνες χορηγούνται ως αγχολυτικά στην προνάρκωση, για εισαγωγή στην αναισθησία, για καταστολή κατά τη διάρκεια περιοχικής ή τοπικής αναισθησίας ή ως συμπλήρωμα της γενικής, για καταστολή ασθενών στη ΜΕΘ και για έλεγχο των σπασμών. Οι βενζοδιαζεπίνες που χρησιμοποιούνται είναι η *Διαζεπάμη*, η *Μιδαζολάμη* και η *Λοραζεπάμη*. Η *Διαζεπάμη*, που έχει μακρά διάρκεια δράσης, χρησιμοποιείται κυρίως για εισαγωγή στην αναισθησία, ενώ η *Λοραζεπάμη*, που προκαλεί έντονη αμνησία και έχει αργή έναρξη δράσης, κυρίως στην προνάρκωση. Η *Μιδαζολάμη* χορηγείται στην προνάρκωση, στην εισαγωγή στην αναισθησία και στην καταστολή στη διάρκεια περιοχικής αναισθησίας. Χρησιμοποιείται πιο συχνά από τις άλλες δύο βενζοδιαζεπίνες επειδή δεν προκαλεί πόνο στην ενδοφλέβια έγχυση και έχει σύντομη έναρξη δράσεως. Δοσολογίες:

Διαζεπάμη

Προνάρκωση	0.1-0.2 mg/kg από το στόμα μίμηση ώρα πριν την εγχείρηση
Εισαγωγή στην αναισθησία	0.2-0.6 mg/kg ενδοφλεβίως αναλόγως της ηλικίας, του σωματικού βάρους και της γενικής κατάστασης του ασθενούς
Καταστολή	0.04-0.2 mg/kg ενδοφλεβίως

Λοραζεπάμη

Προνάρκωση	<ul style="list-style-type: none"> • 0.05 mg/kg από το στόμα 2 ώρες πριν την εγχείρηση • 0.03-0.05 mg/kg ενδομυϊκώς
Καταστολή	0.03-0.04 mg/kg ενδοφλεβίως

Μιδαζολάμη

Προνάρκωση	<ul style="list-style-type: none"> • 0.15-0.20 mg/kg ενδομυϊκώς για παιδιά 20-30 λεπτά πριν την εγχείρηση • 0.10-0.15 mg/kg ενδομυϊκώς για ενήλικες
Εισαγωγή στην αναισθησία	0.17-0.22 mg/kg ή 10-15 mg εφάπαξ ενδοφλεβίως
Καταστολή	0.01-0.1 mg/kg ενδοφλεβίως

- *Οπιοειδή*: Η κύρια δράση των οπιοειδών είναι η καταστολή του πόνου. Χρησιμοποιούνται σε συνδυασμό με άλλα αναισθητικά φάρμακα στην

προνάρκωση, στην εισαγωγή και στη συντήρηση της αναισθησίας καθώς και στη μετεγχειρητική αναλγησία. Τα οποιοειδή που χρησιμοποιούνται ευρέως είναι η Μορφίνη (κυρίως για την αντιμετώπιση μετεγχειρητικού και χρόνιου πόνου), η Πεθιδίνη, η Ρεμφεντανύλη και η Φεντανύλη. Η Ρεμφεντανύλη, που είναι φάρμακο υπερβραχείας διάρκειας δράσης, χρησιμοποιείται κυρίως για αναλγησία κατά τη διάρκεια της γενικής αναισθησίας. Η Φεντανύλη που χαρακτηρίζεται από ταχεία έναρξη και βραχεία διάρκεια χρήσης, χορηγείται για συντήρηση της αναισθησίας και για μετεγχειρητική αναλγησία. Δοσολογίες:

Μορφίνη

Προνάρκωση	0.05-0.2 mg/kg ενδομυϊκώς
Συντήρηση αναισθησίας	0.1-0.4 mg/kg ενδοφλεβίως σε διηρημένες δόσεις
Μετεγχειρητική αναλγησία	<ul style="list-style-type: none"> • 0.015-0.04 mg/kg/h ενδοφλεβίως (PCA) • 0.05-0.2 mg/kg ενδομυϊκώς

Πεθιδίνη

Προνάρκωση	0.5-1 mg/kg ενδομυϊκώς
Συντήρηση αναισθησίας	2.5-5 mg/kg ενδοφλεβίως σε διηρημένες δόσεις
Μετεγχειρητική αναλγησία	<ul style="list-style-type: none"> • 0.2-0.5 mg/kg ενδοφλεβίως (εφάπαξ δόσεις) • 0.5-1 mg/kg ενδομυϊκώς

Ρεμφεντανύλη

Εισαγωγή στην αναισθησία	0.5-1 μg/kg ενδοφλεβίως σε διάστημα όχι μικρότερο των 30''
Συντήρηση αναισθησίας	Συνεχής στάγδην έγχυση αρχικά 0.25-1 μg/kg/min και έπειτα 0.05-1 μg/kg/min

Φεντανύλη

Συντήρηση αναισθησίας	2-150 μg/kg ενδοφλεβίως σε διηρημένες δόσεις
Μετεγχειρητική αναλγησία	0.15-0.3 μg/kg/h ενδοφλεβίως (PCA)

- *Μυοχαλαρωτικά:* Τα μυοχαλαρωτικά χορηγούνται κυρίως για μυοχάλαση κατά τη διάρκεια της γενικής αναισθησίας. Διακρίνονται σε αποπλωτικούς και σε μη αποπλωτικούς νευρομυϊκούς αποκλειστές. Οι αποπλωτικοί έχουν βραχεία διάρκεια δράσης και κύριος εκπρόσωπός τους είναι η σουκινυλοχολίνη η οποία χρησιμοποιείται για γρήγορη εισαγωγή και επείγουσα διασωλήνωση ασθενών. Τα μη αποπλωτικά μυοχαλαρωτικά διακρίνονται αναλόγως τη διάρκεια δράσης τους σε:

- μακράς διάρκειας (>30 min) π.χ. d-Τουβοκουρορίνη που δεν χρησιμοποιείται πλέον και Πανκουρόνιο
- ενδιάμεσης διάρκειας (20-30 min) π.χ. Ατρακούριο, Σισατρακούριο και Ροκουρόνιο
- Βραχείας δράσης (<20 min) π.χ. το Μιβακούριο [3].

Το μυοχαλαρωτικό που χορηγείται πιο συχνά είναι το Ροκουρόνιο. Χρησιμοποιείται για μυοχάλαση στη διάρκεια της γενικής αναισθησίας και για διευκόλυνση της ενδοτραχειακής διασωλήνωσης [4]. Δοσολογία:

Διευκόλυνση της ενδοτραχειακής διασωλήνωσης	0.6 mg/kg ενδοφλεβίως με έναρξη δράσης σε 60-90 sec.
Διατήρηση του νευρομυϊκού αποκλεισμού	<ul style="list-style-type: none"> • διαλείπουσα χορήγηση 0.15 mg/kg ανάλογα με τις απαιτήσεις • συνεχής στάγδην έγχυση σε ρυθμό 5-10 µg/kg/ min

1.4 Επιπλοκές κατά τη γενική αναισθησία

Από το 1950 και έπειτα, πολλοί επιστήμονες άρχισαν να συνειδητοποιούν πως στη χορήγηση γενικής αναισθησίας ελλοχεύουν πολλοί κίνδυνοι που μπορούν να οδηγήσουν ακόμα και στο θάνατο. Έτσι, άρχισαν να διεξάγονται πολλές έρευνες για την καταγραφή γεγονότων που σχετίζονταν με την αναισθησία (ατυχήματα, θάνατοι, επιπλοκές) καθώς και για την αναζήτηση των αιτιών τους. Παρακάτω ακολουθούν οι κυριότερες επιπλοκές της γενικής αναισθησίας όπως έχουν προκύψει από έρευνες, καθώς και ενδεικτικά αποτελέσματα από μεγάλες έρευνες που πραγματοποιήθηκαν σε επιλεγμένες χώρες.

Οι συχνότερες επιπλοκές της αναισθησίας είναι οι εξής:

- Ναυτία
- Εμετός
- Αλλεργία στα φάρμακα
- Πόνος στο λαιμό
- Τραυματισμός στόματος και δοντιών
- Κεφαλαλγία
- Υποθερμία
- Μυαλγία
- Ζάλη και θολή όραση
- Σύγχυση και απώλεια μνήμης
- Υπόταση
- Υπέρταση
- Βρογχόσπασμος – Λαρυγγόσπασμος
- Διατήρηση συνείδησης

Οι πιο σπάνιες και σοβαρότερες επιπλοκές είναι οι εξής:

- Πνευμονική λοίμωξη
- Καρδιακή βλάβη
- Εγκεφαλική βλάβη
- Θρόμβωση
- Κάκωση νεύρων
- Κακοήθης υπερθερμία
- Θάνατος

Αυστραλία: Σύμφωνα με μία μελέτη που πραγματοποιήθηκε στην Αυστραλία από την AIMS (Australian Incident Monitoring Study) και αφορά τη χρονική περίοδο 1988-2001, από τα 8088 συμβάματα που καταγράφηκαν, τα 896 αφορούσαν λάθη που σχετιζόνταν με φάρμακα (drug error) [6]. Ως σύμβαμα στην ιατρική ορίζεται 'ένα μη αναμενόμενο γεγονός που μείωσε ή θα μπορούσε να μειώσει τα όρια ασφαλείας του ασθενούς' [7]. Από αυτά τα 896, τα 452 (50,4%) οφείλονταν σε λάθη κατά την προετοιμασία της σύριγγας και του φαρμάκου, τα 234 (26,1 %) σε δυσλειτουργία του εξοπλισμού, τα 126 (14,1%) σε εσφαλμένη οδό χορήγησης και τα 35 σε σφάλμα επικοινωνίας του ιατρικού προσωπικού. Από τα 452 λάθη κατά την προετοιμασία της σύριγγας και του φαρμάκου, τα 55 ήταν σφάλματα της διαδικασίας διάλυσης του φαρμάκου, ή σφάλματα στην περιεκτικότητα του φαρμάκου. Τα 169 περιστατικά οφείλονταν στη χορήγηση λάθους φαρμάκου, ενώ τα 187 οφείλονταν σε λάθη στις ετικέτες των φαρμάκων.

Η ίδια έρευνα έδειξε ότι οι κύριοι παράγοντες που συνέβαλαν στα σφάλματα αυτά ήταν οι κάτωθι:

- Έλλειψη προσοχής
- Βιασύνη
- Εσφαλμένες ετικέτες φαρμάκων
- Περισπασμός
- Λάθη επικοινωνίας
- Κόπωση
- Έλλειψη ελέγχων του εξοπλισμού

Τα αποτελέσματα των 896 περιστατικών ήταν 105 περιπτώσεις μικρής νοσηρότητας, 42 περιπτώσεις σημαντικής νοσηρότητας, τα 20 οδήγησαν σε ανάγκη καθημερινής φροντίδας, τα 25 σε παρατεταμένη παραμονή στο νοσοκομείο, τα 40 σε διατήρηση της συνείδησης κατά τη διάρκεια της επέμβασης, τα 658 δεν είχαν κάποια σημαντική επιπλοκή, ενώ τα 3 είχαν μη καθορισμένες παρενέργειες. Τέλος, σε 3 από τις περιπτώσεις, τα σφάλματα οδήγησαν στο θάνατο του ασθενούς.

Επίσης, η έρευνα έδειξε ότι τα φάρμακα που εμπλέκονταν κυρίως στα περιστατικά λάθους χορήγησης ήταν τα οποιοειδή, οι βενζοδιαζεπίνες και τα μυοχαλαρωτικά.

Ιαπωνία: Σε έρευνα που διεξήχθη στα νοσοκομεία της Ιαπωνίας σε διάρκεια 5 ετών (1994-1998), μελετήθηκαν 2,363,038 περιπτώσεις ασθενών και προέκυψε ότι το ποσοστό θνησιμότητας από αναισθησία ήταν περίπου 1:48.000 [8]. Αξιοσημείωτο ήταν ότι το 22,4% των θανάτων οφείλονταν σε αποτρεψίμα ανθρώπινα λάθη.

Πιο πρόσφατα, η Εταιρία Ιαπόνων Αναισθησιολόγων, JSA (Japanese Society of Anesthesiologists) μελέτησε 27.454 περιπτώσεις αναισθησίας σε ένα πανεπιστημιακό νοσοκομείο σε διάρκεια 8 ετών (1999-2007). Καταγράφηκαν 233 συμβάματα από τα οποία τα 61 οφείλονταν σε υπερδοσολογία, παράλειψη ή χορήγηση λάθος φαρμάκου [9].

Δανία: Σε έρευνα που βασιζόταν σε δεδομένα της Ασφαλιστικής Ένωσης ασθενών Δανίας (Danish Patient Insurance Association), προέκυψε ότι κατά τη χρονική διάρκεια μεταξύ του 1996 και 2004, 24 θάνατοι στα νοσοκομεία της Δανίας προήλθαν από λάθη κατά τη χορήγηση της αναισθησίας. Μάλιστα, οι 8 εξ' αυτών προήλθαν από λάθη στη χορήγηση φαρμάκων όπως υπερδοσολογία [10].

Γαλλία: Σε έρευνα που διεξήχθη το 1999 σε ολόκληρη τη Γαλλία [11], προέκυψε ότι τα ποσοστά θνησιμότητας από την αναισθησία μειώθηκαν 10 φορές σε σχέση με αυτά που είχαν προκύψει από μία άλλη έρευνα του Υπουργείου Υγείας της Γαλλίας στο διάστημα 1978 έως 1982 [12]. Συγκεκριμένα, στο διάστημα αυτών των 4 ετών, καταγράφηκαν 268 σοβαρές επιπλοκές που οφείλονταν στην αναισθησία, συμπεριλαμβάνοντας 67 θανάτους και 16 περιπτώσεις όπου ο ασθενής έπεσε σε κώμα. Το ποσοστό θνησιμότητας υπολογίστηκε 1:13.207 ενώ το 1999 0,69:100.000.

Ηνωμένο Βασίλειο: Το Εθνικό Σύστημα Αναφοράς και Εκμάθησης του Ηνωμένου Βασιλείου (National Reporting and Learning System) μελέτησε στο χρονικό διάστημα από τον Ιανουάριο του 2004 έως το Φεβρουάριο του 2006, 12.606 περιστατικά σχετιζόμενα με την αναισθησία που μπορούσαν να βλάψουν ή έβλαψαν οποιοδήποτε ασθενή που έλαβε περίθαλψη χρηματοδοτούμενη από την Εθνική Υπηρεσία Υγείας (NHS). Τα 2842 (22,5%) κατέληξαν σε μικρού ή μεσαίου βαθμού νοσηρότητα του ασθενή ενώ 269 (2,1%) σε σοβαρή επιπλοκή ή θάνατο [13].

Νέα Ζηλανδία: Σε μία έρευνα που πραγματοποιήθηκε στη Νέα Ζηλανδία, το 89% των ιατρών που πήραν μέρος, παραδέχθηκαν πως έχουν κάνει τουλάχιστον ένα λάθος στη χορήγηση αναισθητικού φαρμάκου κάποια στιγμή στην καριέρα τους. Επίσης, προέκυψε ότι το ποσοστό λανθασμένης ενδοφλέβιας χορήγησης είναι 1:200 όπου το 20% οφείλεται σε λάθος δόση και ένα άλλο 20% σε χορήγηση λάθος φαρμάκου [14].

Καναδάς: Το 85% των 687 αναισθησιολόγων που πήραν μέρος σε έρευνα που διεξήχθη μεταξύ της Ένωσης Αναισθησιολόγων του Καναδά, παραδέχθηκαν ότι έχει κάνει τουλάχιστον ένα λάθος στη χορήγηση φαρμάκων. Από την ίδια έρευνα προέκυψε ότι το 70,4% των λαθών αφορούσαν χορήγηση λάθος φαρμάκου, το 46,8% λάθη στις ετικέτες των φαρμάκων και το 62,9% λάθη στην ανάγνωση της ετικέτας. Τα αποτελέσματα αυτών των λαθών ήταν τα εξής:

- 368 (35,4%) περιπτώσεις μικρής νοσηρότητας
- 597 (57,5%) περιπτώσεις με καμία σημαντική επιπλοκή
- 15 (1,4%) περιπτώσεις σημαντικής νοσηρότητας (καρδιακή προσβολή, έμφραγμα, τραυματισμός)
- 4 (0,4%) θάνατοι.

Αξιοσημείωτο είναι επίσης πως η πλειοψηφία των αναισθησιολόγων παραδέχτηκε ότι δεν διαβάζει σχεδόν ποτέ το όνομα του φαρμάκου στην ετικέτα [15].

Ηνωμένες Πολιτείες της Αμερικής: Σε έρευνα που διεξήχθη σε 4 μεγάλα νοσοκομεία της Βοστώνης διαπιστώθηκε ότι τα ανθρώπινα λάθη ήταν η κύρια αιτία των άσχημων περιστατικών που σχετίζονταν με την αναισθησία. Καταγράφηκαν 1089 συμβάματα από τα οποία τα 70 ήταν σημαντικής νοσηρότητας [16]. Σε μια άλλη έρευνα που δημοσιεύθηκε από την ίδια επιστημονική ομάδα, προέκυψε ότι οι παράγοντες που συνέβαλαν στα σφάλματα κατά τη διάρκεια της αναισθησίας είναι οι κάτωθι [17]:

- Έλλειψη εμπειρίας (16%)
- Έλλειψη εξοικείωσης με τον εξοπλισμό (9,4%)
- Κακή επικοινωνία με το υπόλοιπο ιατρικό προσωπικό (5,6%)
- Βιασύνη (5,4%)
- Απροσεξία (5,4%)
- Κούραση (5%)
- Υπερβολική εξάρτηση από το υπόλοιπο προσωπικό (5%)
- Αποτυχία να εκτελεστεί φυσιολογικός έλεγχος (4,6%)
- Κατάρτιση ή εμπειρία (4,6%)
- Έλλειψη αρκετής επιτήρησης (3,7%)
- Εργασιακό περιβάλλον ή συνάδελφοι (3,7%)
- Περιορισμένο οπτικό πεδίο (3,5%)
- Ψυχικοί και σωματικοί παράγοντες (3,3%)
- Έλλειψη εξοικείωσης με χειρουργικές επεμβάσεις (2,9%)
- Απόσπαση προσοχής (2,7%)
- Λάθη στις ετικέτες των φαρμάκων (2,5%)
- Παράγοντες σχετιζόμενοι με την επιτήρηση (2,5%)
- Μη τήρηση συνηθισμένων προφυλάξεων (2,1%)
- Έλλειψη εξοικείωσης με την τεχνική αναισθησία (2,1%)
- Διδακτική διαδικασία σε εξέλιξη (1,9%)
- Ανησυχία (1,7%)
- Κατάσταση εκτάκτου ανάγκης (1,2%)
- Απαιτητική ή δύσκολη περίπτωση (1,2%)
- Πλήξη (1%)
- Είδος της αναισθησίας (1%)
- Ανεπαρκής προετοιμασία (0,6%)
- Αργή διαδικασία (0,6%)
- Άλλο (0,6%)

Βραζιλία: Σε έρευνα που διεξήχθη στη Βραζιλία, προέκυψε ότι την περίοδο 1982-1984 τα ποσοστά θνησιμότητας εξαιτίας της αναισθησίας ήταν 2,28:10.000 ενώ την περίοδο 1998-2006 0,12-1,12:10.000 [18].

Νορβηγία: Κατά την περίοδο 1996-1998 πραγματοποιήθηκε μία έρευνα σε νοσοκομείο της Νορβηγίας με σκοπό να μετρηθούν τα ποσοστά λάθους στη χορήγηση φαρμάκων κατά την αναισθησία και κατά πόσο επηρεάζονται από τη χρήση χρωματιστών ετικετών στις

σύριγγες [19]. Καταγράφηκαν 63 περιστατικά σφάλματος σε συνολικά 55.426 χειρουργικές επεμβάσεις από τα οποία τα 37 (58%) αφορούσαν χορήγηση λάθος φαρμάκου, οι 18 (29%) λάθος δόση φαρμάκου και οι υπόλοιπες 8 (13%) άλλες υποθέσεις όπως εκδήλωση αλλεργίας του ασθενή στα φάρμακα. Από αυτά τα περιστατικά, τα 27 ήταν ενδιάμεσης νοσηρότητας και τα 3 χαρακτηρίστηκαν σοβαρά. Επίσης προέκυψε ότι η χρήση χρωματιστών ετικετών δεν μείωσε σχεδόν καθόλου τα ποσοστά λάθους.

Παρατηρούμε ότι στη διάρκεια των τελευταίων ετών, η πιθανότητα σοβαρής αναισθησιολογικής επιπλοκής όπως θάνατος ή καρδιακή ανακοπή έχει μειωθεί αισθητά. Το γεγονός αυτό οφείλεται στις πιο ασφαλείς τεχνικές που ακολουθούνται, στις σύγχρονες ηλεκτρονικές συσκευές παρακολούθησης των ασθενών (monitoring) και στα νέα βελτιωμένα αναισθητικά φάρμακα. Παρ' όλα αυτά, ο αριθμός των επιπλοκών είναι ακόμα μεγάλος και ένας από τους κύριους παράγοντες, όπως προκύπτει από τις έρευνες, είναι το ανθρώπινο λάθος κατά τη χορήγηση φαρμάκου (π.χ. υπερδοσολογία, ακούσια χορήγηση λάθος φαρμάκου). Δυστυχώς, ένα ποσοστό ανθρώπινου λάθους είναι αναπόφευκτο και δεν είναι αναγκαστικά συνδεδεμένο με ανικανότητα. Η χορήγηση ενδοφλέβιας αναισθησίας είναι μια ιδιαίτερη πολύπλοκη διαδικασία που μπορεί να απαιτήσει μέχρι και 40 βήματα. Έτσι, υπό συνθήκες πίεσης, κούρασης, έλλειψης προσωπικού και επικοινωνίας συνυπολογίζοντας και το γεγονός ότι πολλά φάρμακα έχουν παρόμοια συσκευασία, δεν είναι δύσκολο να γίνει ένα λάθος. Πολλές στρατηγικές έχουν προταθεί για την ελάττωση των σφαλμάτων κατά τη χορήγηση φαρμάκων όπως βελτίωση της εκπαίδευσης των αναισθησιολόγων και του προσωπικού που παίρνει μέρος στη χορήγηση αναισθησίας, χρησιμοποίηση χρωματικών ετικετών για τις σύριγγες, χρησιμοποίηση σταθερών και μη μεταβαλλόμενων διαλυμάτων φαρμάκων [7]. Βέβαια όπως είδαμε στη Νορβηγία, τα ποσοστά επιπλοκών δεν άλλαξαν σημαντικά με τη χρησιμοποίηση χρωματικών ετικετών στις σύριγγες. Γίνεται κατανοητό λοιπόν πως πρέπει να ληφθούν διαφορετικά μέτρα. Αυτός είναι ο σκοπός της παρούσας διπλωματικής. Χρησιμοποιούμε την τεχνολογία RFID, ένα υπολογιστικό σύστημα βασισμένο στην πλατφόρμα Arduino και μια βάση δεδομένων με τα χαρακτηριστικά των φαρμάκων αναισθησίας για να φτιάξουμε ένα σύστημα ελέγχου της χορήγησης ενδοφλέβιας αναισθησίας με σκοπό τη μείωση της συμμετοχής του ανθρώπινου παράγοντα στη διαδικασία, που θα οδηγήσει στη μείωση της πιθανότητας επιπλοκών λόγω ανθρώπινου λάθους.

2. Η ΤΕΧΝΟΛΟΓΙΑ RFID

2.1 Γενικά

Μια από τις πιο διαδεδομένες τεχνολογίες στις μέρες μας θεωρείται η τεχνολογία RFID (Radio Frequency Identification). Η “ταυτοποίηση μέσω συχνοτήτων”/ “Ραδιοσυχνική Αναγνώριση” όπως αναφέρεται στην ελληνική βιβλιογραφία, ανήκει στην γενικότερη κατηγορία των Συστημάτων Αυτόματης Αναγνώρισης και Συλλογής Δεδομένων AIDC (Automatic Identification and Data Capture Technology) και αποτελεί εξέλιξη των ραβδωτών κωδικών (barcodes). Η τεχνολογία RFID χρησιμοποιεί τα ραδιοκύματα για να αναγνωρίζει, να εντοπίζει και να συλλέγει πληροφορίες τόσο για έμψυχα όσο και για άψυχα αντικείμενα.

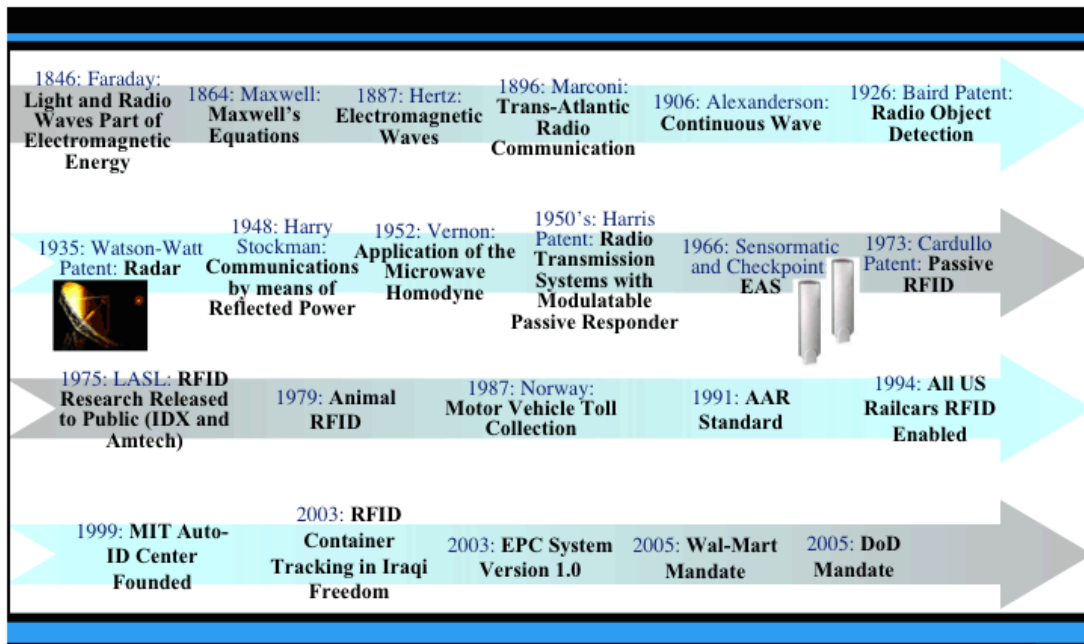
2.2 Ιστορική αναδρομή

Η τεχνολογία RFID πρωτοεμφανίστηκε κατά τη διάρκεια του Β' Παγκοσμίου πολέμου για στρατιωτικούς λόγους. Συγκεκριμένα αναπτύχθηκε από τους Βρετανούς πιλότους το σύστημα IFF (Friend or Foe Identification) για να αποτρέπεται η κατάρριψη των “φιλικών” αεροπλάνων. Το στρατιωτικό σύστημα των συμμαχικών δυνάμεων εξέπεμπε ένα ραδιοσήμα σε κάθε αεροπλάνο που πλησίαζε και εφόσον ήταν φιλικό, ανάγκαζε τον αναμεταδότη του αεροπλάνου να ανταποκριθεί [20].

Με την απαραίτητη τεχνολογική ανάπτυξη, στο διάστημα από το 1940 έως και το 1970 παρουσιάστηκε μεγάλη πρόοδος στην έρευνα αυτής της νέας τεχνολογίας κυρίως για εμπορικούς λόγους. Αναπτύχθηκαν συστήματα ηλεκτρονικής παρακολούθησης αντικειμένων (EAS-electronic article surveillance) [20], που χρησιμοποιούσαν ετικέτες με πληροφορία ενός bit, για την ανίχνευση προϊόντων που δεν είχαν πληρωθεί στα καταστήματα και κατ' επέκταση την αποτροπή κλοπής.

Το 1971 ο Mario Cardullo έκανε επίδειξη της τεχνολογίας, σε μορφή παρόμοια με αυτή που χρησιμοποιείται σήμερα, σε εταιρία διαχείρισης λιμένων με σκοπό την παρακολούθηση εμπορευμάτων. Λίγα χρόνια αργότερα, το 1973 γίνεται η παρουσίαση της πρώτης RFID ετικέτας από τους Steven Depp, Alfred Koelle, and Robert Frayman στο ερευνητικό εργαστήριο του Los Alamos των ΗΠΑ [21]. Παρόλ' αυτά η πρώτη πατέντα με τον όρο RFID κατοχυρώθηκε το 1983 από τον Charles Waton [22].

Στις επόμενες δεκαετίες ο αριθμός των εταιριών και των ιδρυμάτων που ασχολήθηκαν με τη μελέτη ανάλογων εφαρμογών αυξήθηκε κατακόρυφα. Η τεχνολογία RFID αρχίζει να χρησιμοποιείται σε εργοστάσια, αυτοκινητοβιομηχανίες, σε δρόμους, στον εντοπισμό ζώων, φορτίων και σε πολλούς άλλους τομείς. Σήμερα χρησιμοποιείται σε πιο ευρεία γκάμα εφαρμογών (παρακάτω θα αναφερθούν ενδεικτικά ορισμένες) και είναι συνυφασμένη με τη ζωή μας.

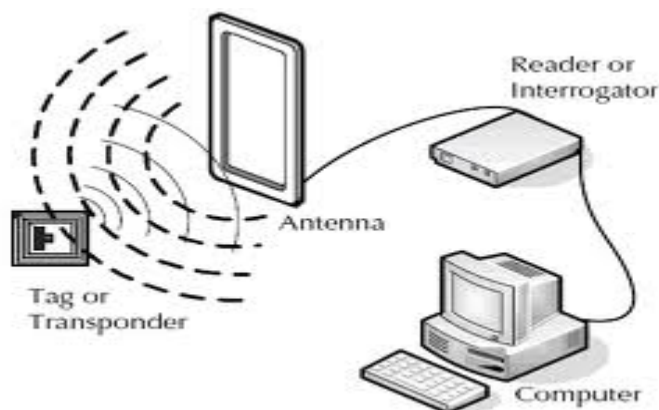


Σχήμα 2.1: Ιστορική αναδρομή RFID τεχνολογίας

2.3 Δομή και αρχές λειτουργίας των συστημάτων RFID

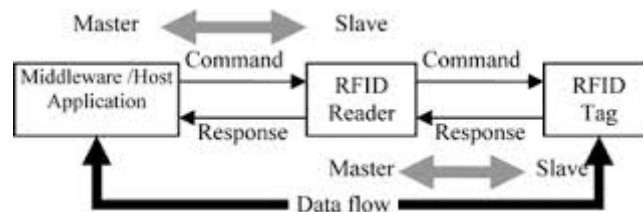
Κάθε RFID σύστημα αποτελείται από τα εξής θεμελιώδη στοιχεία:

1. Έναν ή περισσότερους πομποδέκτες/ετικέτες (transponders/tags) οι οποίοι τοποθετούνται στο προς ταυτοποίηση αντικείμενο.
2. Τον αναγνώστη (reader) ο οποίος αναγνωρίζει τους πομποδέκτες/ετικέτες που βρίσκονται στο πεδίο εμβέλειας του και διαβάζει το περιεχόμενό τους.
3. Το ενδιάμεσο λογισμικό του συστήματος (Middleware), ο ρόλος του οποίου είναι η διαχείριση και η επεξεργασία των δεδομένων που διαβάζει ο αναγνώστης από τους πομποδέκτες/ετικέτες. Μπορεί να εκτελείται σε μικροεπεξεργαστή ή σε ξεχωριστό υπολογιστικό σύστημα.



Σχήμα 2.2: Τυπικό RFID σύστημα

Η λειτουργία ενός RFID συστήματος στηρίζεται στο μοντέλο αφέντη-δούλου (master-slave). Αρχικά το μοντέλο εφαρμόζεται στην επικοινωνία λογισμικού-αναγνώστη όπου το λογισμικό, αναλαμβάνοντας το ρόλο του αφέντη, ενεργοποιεί τον αναγνώστη (δούλο) δίνοντάς του τις απαραίτητες εντολές. Στη συνέχεια, το ίδιο μοντέλο εμφανίζεται στην επικοινωνία μεταξύ του αναγνώστη και της ετικέτας με τον αναγνώστη να αναλαμβάνει το ρόλο του αφέντη. Συγκεκριμένα ο αναγνώστης εκπέμπει μέσω ραδιοσυχνότητας ένα σήμα το οποίο λαμβάνεται από όσες ετικέτες βρίσκονται στο πεδίο εμβέλειάς του. Αυτοί τότε ενεργοποιούνται εκπέμποντας πίσω τα δεδομένα που έχουν αποθηκευμένα. Κατόπιν ο αναγνώστης λαμβάνει αυτά τα δεδομένα, τα οποία και μεταδίδει στο πληροφοριακό σύστημα ώστε να υποστούν επεξεργασία και να αποθηκευτούν. Η επικοινωνία αυτή μπορεί να είναι ενσύρματη ή ασύρματη, σύμφωνα με τις απαιτήσεις και τις δυνατότητες του συστήματος [23].



Σχήμα 2.3: Μοντέλο master-slave

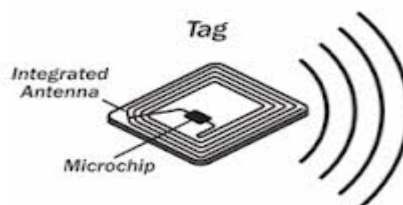
2.4 Συστατικά μέρη ενός RFID συστήματος

2.4.1 Ετικέτες

2.4.1.1 Χαρακτηριστικά

Οι βασικές λειτουργίες μίας ετικέτας είναι η αποθήκευση δεδομένων και η αποστολή τους στον αναγνώστη. Τα βασικά στοιχεία μίας ετικέτας είναι τα εξής [24]:

1. Ένα ολοκληρωμένο κύκλωμα (microchip) που περιέχει τα δεδομένα και αναλαμβάνει τη διαμόρφωση/αποδιαμόρφωση των σημάτων που χρησιμοποιούνται για την επικοινωνία.
2. Την κεραία η οποία χρησιμοποιείται για την επικοινωνία με τον αναγνώστη.
3. Σε κάποιες περιπτώσεις, ενσωματωμένη παροχή ισχύος.



Σχήμα 2.4: Ετικέτα

2.4.1.2 Κατηγορίες ετικετών

- Με βάση την πηγή ενέργειάς τους

Παθητικές (passive): Οι παθητικές ετικέτες δεν χρειάζονται εξωτερική πηγή ισχύος αλλά τροφοδοτούνται από τα ραδιοκύματα που εκπέμπονται από τον αναγνώστη [25]. Έτσι όταν βρίσκονται εκτός του πεδίου εμβέλειας του παραμένουν αδρανείς.



Σχήμα 2.5: Παθητική ετικέτα

Ενεργητικές (active): Οι ενεργητικές ετικέτες διαθέτουν ενσωματωμένη παροχή ισχύος (π.χ. μια μπαταρία) για τη λειτουργία του εσωτερικού τους κυκλώματος. Και αυτή η κατηγορία ετικετών χρησιμοποιεί την ενέργεια των ραδιοκυμάτων για την ενεργοποίησή τους.



Σχήμα 2.6: Ενεργητικές ετικέτες

Σύγκριση παθητικών και ενεργητικών ετικετών: Οι παθητικές ετικέτες έχουν εν γένει χαμηλότερο κόστος κατασκευής και λειτουργίας, καθώς και απεριόριστη διάρκεια λειτουργίας, αφού δεν χρειάζονται εξωτερική πηγή ισχύος. Έχουν συνήθως μικρότερες διαστάσεις, και έχουν αυξημένο χρόνο ζωής το οποίο τους καθιστά πιο αξιόπιστες [25]. Παρ' όλα αυτά παρουσιάζουν κάποια σημαντικά μειονεκτήματα. Καταρχάς οι ενεργητικές ετικέτες λόγω της πηγής ενέργειάς τους, έχουν πολλαπλάσια εμβέλεια (έως 100m) σε αντίθεση με τις παθητικές (έως 3m). Επιπλέον οι παθητικές έχουν μικρότερη ταχύτητα μετάδοσης δεδομένων, αφού στηρίζονται στην ενέργεια που λαμβάνουν από τον αναγνώστη για να τροφοδοτήσουν την αποστολή δεδομένων. Τέλος, μπορεί οι

ενεργητικές ετικέτες να έχουν υψηλό κόστος αλλά έχουν συνήθως κάποιες επιπλέον δυνατότητες, όπως η εκτέλεση λειτουργιών χωρίς την παρουσία του αναγνώστη [26].

Ημιενεργητικές / ημιπαθητικές ετικέτες (semi-active/semi passive):

Η συγκεκριμένη κατηγορία ετικετών είναι ένας συνδυασμός των ενεργητικών και παθητικών ετικετών. Διαθέτουν ενσωματωμένη μπαταρία την οποία χρησιμοποιούν για τη τροφοδοσία του ολοκληρωμένου κυκλώματος του και την αποθήκευση των δεδομένων, αλλά όχι για τη μετάδοση των ραδιοσημάτων στον αναγνώστη [26]. Η εμβέλεια αυτού του τύπου ετικετών είναι μεγαλύτερη από αυτή των παθητικών και μικρότερη από των ενεργητικών, δεν χρειάζονται συντήρηση ενώ το κόστος κατασκευής τους είναι χαμηλότερο από αυτό των ενεργητικών. Ωστόσο, είναι συνήθως μια χρήσης, καθώς αχρηστεύονται όταν αποφορτιστεί η μπαταρία [27].

Πίνακας 2.1: Χαρακτηριστικά ετικετών με βάση την πηγή ενέργειάς τους

Tags and Features	Passive Tag	Active Tag	Semi Passive Tag
Internal Power Source	No	Yes	Yes
Signal by backscattering the carrier wave from the reader	Yes	No	Yes
Response	Weaker	Stronger	Stronger
Size	Small	Big	Medium
Cost	Less expensive	More expensive	Less
Potential Shelf life	Longer	Shorter	Longer
Range	10 centimeters to few meters	Hundreds of meters	Hundreds of meters
Sensors	No	Yes	Yes

➤ Με βάση τη δυνατότητα επανεγγραφής τους

Ετικέτες μόνο ανάγνωσης (Read only): Τα δεδομένα που αποθηκεύονται σε αυτό τον τύπο ετικέτας (συνήθως ένας μοναδικός σειριακός αριθμός ID) εγγράφονται κατά την κατασκευή τους με χρήση laser και δεν μπορούν να τροποποιηθούν στη συνέχεια. Επιτρέπεται δηλαδή μόνο η ανάγνωσή τους όταν βρεθούν στην περιοχή εμβέλειας του αναγνώστη [27].

Ετικέτες μιας εγγραφής – πολλών αναγνώσεων (Write once - Read many): Οι ετικέτες αυτής της κατηγορίας, γνωστές και ως WORM, μπορούν να εγγραφούν μια φορά. Ενσωματώνουν δεδομένα από την κατασκευαστική εταιρία ενώ επιτρέπουν στο χρήστη να εγγράψει τον επιθυμητό σειριακό αριθμό μόνο μια

φορά. Εφόσον η εγγραφή ολοκληρωθεί, οι ετικέτες αυτές μετατρέπονται σε μόνο ανάγνωσης [28].

Ετικέτες ανάγνωσης – εγγραφής (Read – Write): Οι συγκεκριμένες ετικέτες εγγράφονται κατά την κατασκευή τους, αλλά τα δεδομένα τους μπορούν να επανεγγραφούν και να διαβαστούν χωρίς όριο στον αριθμό φορών που μπορεί να συμβεί αυτό. Διαθέτουν τη μεγαλύτερη χωρητικότητα, είναι οι πιο λειτουργικές αλλά και οι πιο ακριβές.

➤ Με βάση τη συχνότητα λειτουργίας τους

Οι συχνότητες των ετικετών διακρίνονται σε 4 κύριες ζώνες σύμφωνα με το εύρος συχνοτήτων στο οποίο λειτουργούν [25]:

Ζώνη χαμηλής συχνότητας (Low Frequency – LF): 125 – 134 KHz

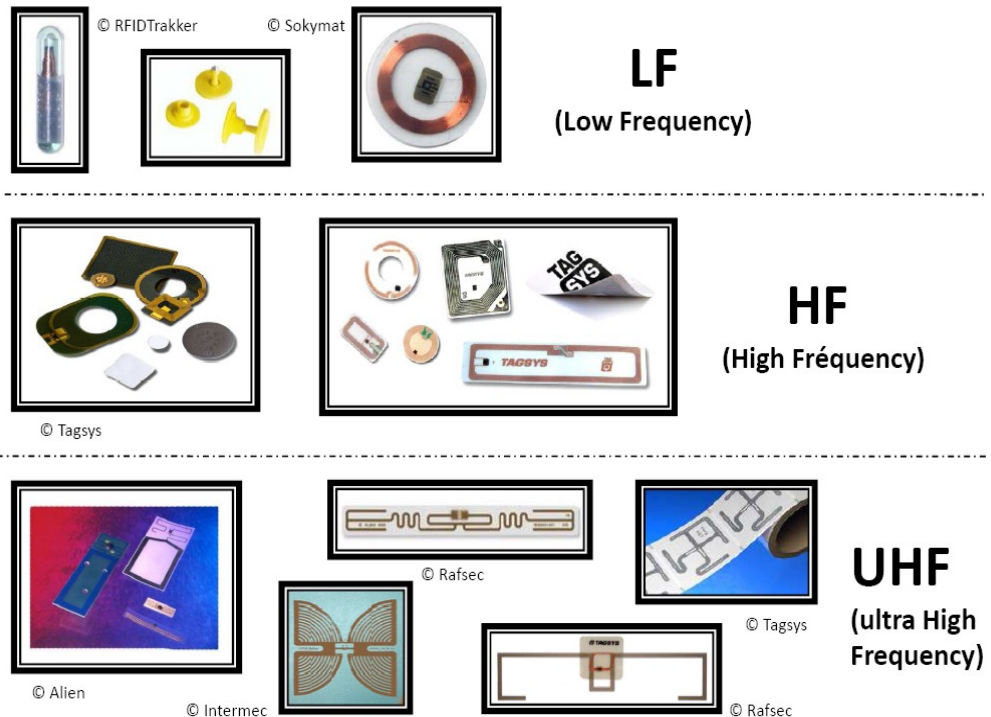
Στις χαμηλές συχνότητες οι ετικέτες έχουν μικρό εύρος εμβέλειας αλλά είναι ιδιαίτερα αποτελεσματικές σε περιβάλλοντα με πολλά εμπόδια, καθώς και για την αναγνώριση αντικειμένων με υψηλή περιεκτικότητα σε νερό. Χρησιμοποιούνται κυρίως για την πρόσβαση σε χώρους με ελεγχόμενη είσοδο, στην αναγνώριση ζώων και προϊόντων και στην παρακολούθηση περιουσιακών στοιχείων.

Ζώνη υψηλής συχνότητας (High Frequency – HF): 13.56 MHz

Οι ετικέτες αυτής της κατηγορίας έχουν μεγαλύτερη εμβέλεια και μεγαλύτερο ρυθμό μετάδοσης δεδομένων από τις ετικέτες LF και δεν επηρεάζονται από το νερό. Χρησιμοποιούνται σε συστήματα πληρωμής, στον έλεγχο αποσκευών, πρόσβασης και στις έξυπνες κάρτες.

Ζώνη πολύ υψηλής συχνότητας (Ultra high frequency-UHF): 433 - 950MHz

Στη ζώνη πολύ υψηλής συχνότητας οι ετικέτες έχουν μεγαλύτερη απόσταση ανάγνωσης (έως και 12m) καθώς και υψηλότερη ταχύτητα μεταφοράς δεδομένων. Ωστόσο, δεν είναι τόσο αποτελεσματικές όταν παρεμβάλλονται μέταλλα και υγρά. Για αυτό το λόγο οι εταιρίες κατασκευής χρησιμοποιούν διάφορα είδη κεραιών, ώστε να αντιμετωπίσουν αυτό το πρόβλημα. Χρησιμοποιούνται κυρίως στην παρακολούθηση αποθεμάτων λιανικής πώλησης, containers και στην καταπολέμηση της παραποίησης των φαρμακευτικών προϊόντων



Σχήμα 2.7: Ετικέτες στις διάφορες συχνότητες λειτουργίας

Μικροκυματική συχνότητα (Microwave frequency) : 2,45 GHz

Στη συχνότητα αυτή, οι ετικέτες παρουσιάζουν παρόμοια χαρακτηριστικά με αυτά των UHF, με τη διαφορά ότι έχουν υψηλότερο ρυθμό μετάδοσης δεδομένων. Είναι πιο ακριβοί και χρησιμοποιούνται σε ειδικές εφαρμογές όπως στον έλεγχο πρόσβασης στα οχήματα.

➤ Με βάση την κατασκευή τους

Αναλόγως την εφαρμογή για την οποία θα χρησιμοποιηθούν, οι RFID ετικέτες διαφέρουν μεταξύ τους ως προς τον τρόπο ενσωμάτωσης της κεραίας και του ενσωματωμένου κυκλώματος. Μερικά από τα πιο διαδεδομένα είδη ετικετών είναι τα εξής:

- Έξυπνες ετικέτες (smart labels): Οι ετικέτες αυτής της κατηγορίας είναι χάρτινες ή πλαστικές και φέρουν εκτυπωμένο το γραμμωτό κώδικα (bar code) και μια ετικέτα τύπου επιφανειακής τοποθέτησης (inlay). Ο συγκεκριμένος τύπος ετικέτας έχει τη μορφή πλαστικού αυτοκόλλητου, στο οποίο τυπώνεται το ολοκληρωμένο σύστημα και η κεραία με μεταξοτυπία ή χάραξη [23]. Χρησιμοποιούνται στον έλεγχο φορτίων παλέτων ενώ η παραλλαγή τους, οι έξυπνες κάρτες μη επαφής (contactless smart cards), χρησιμοποιούνται για τον έλεγχο πρόσβασης.



Σχήμα 2.8: Smart label

- Δίσκος (disk): Είναι το πιο διαδεδομένο είδος ετικέτας. Είναι μια πολύ μικρή στρογγυλή θερμοπλαστική κατασκευή με μια οπή στο κέντρο για την τοποθέτηση βίδας. Λειτουργούν σε μεγάλο εύρος θερμοκρασιών, έχουν μεγάλη αντοχή και λόγω της κατασκευής τους μπορούν να ράβονται σε ρούχα [23].



Σχήμα 2.9: Δίσκοι

- Γυάλινοι σωλήνες (glass tubes): Οι συγκεκριμένες ετικέτες έχουν πολύ μικρό μήκος (περίπου 30mm) και χρησιμοποιούνται για τον εντοπισμό ανθρώπων και ζώων. Τοποθετούνται ενέσιμα κάτω από το προς ταυτοποίηση δέρμα και έχουν μεγάλο αποθηκευτικό χώρο. Άλλες κατηγορίες ετικετών που χρησιμοποιούνται για την ταυτοποίηση ζώων είναι και οι ετικέτες ενωτίου (ear tags) που τοποθετούνται στο αυτί, οι ετικέτες περιλαίμιου (collar tags) καθώς και οι κεραμικές (ceramic tags) οι οποίες καταπίνονται και παραμένουν στο στομάχι τους [27].



Σχήμα 2.10: Γυάλινοι σωλήνες



Σχήμα 2.11: Ear tag



Σχήμα 2.12: Collar tag



Σχήμα 2.13: Ceramic tags

- Coil-on-chip: Με τον όρο coil-on-chip αναφερόμαστε σε ένα σχετικά νέο είδος ετικέτας όπου το πηνίο της κεραίας είναι ενσωματωμένο στο ολοκληρωμένο κύκλωμα. Παρόλο που η απόσταση ανάγνωσης είναι μόνο 3mm, το αποτέλεσμα είναι μία μικροσκοπική ετικέτα που μπορεί να κρυφτεί σε χαρτονομίσματα.

➤ Με βάση την κλάση

Μια άλλη κατηγοριοποίηση των ετικετών γίνεται αναλόγως την κλάση τους. Παρακάτω ακολουθεί ο πίνακας με τα χαρακτηριστικά ανά κλάση

Πίνακας 2.2: Χαρακτηριστικά ετικετών ανά κλάση

Κλάσεις	Είδος ετικέτας	Κόστος	Εμβέλεια	Δυνατότητα αποθήκευσης επιπλέον δεδομένων	Παρουσία Αναγνώστη	Επιπλέον λειτουργίες
0, 1	Παθητικές	Χαμηλό	< 3 m	Όχι	Αναγκαία	Όχι
2	Παθητικές	Χαμηλό	< 3 m	Ναι	Αναγκαία	Ναι π.χ. δυνατότητα ενσωματωμένου αισθητήρα, κρυπτογράφηση, authentication
3	Ημι-παθητικές με ενσωματωμένη μπαταρία	Μεσαίο	< 100 m	Ναι	Αναγκαία	Ναι π.χ. δυνατότητα ενσωματωμένου αισθητήρα με datalogging, κρυπτογράφηση, authentication
4	Ενεργητικές με ενσωματωμένη μπαταρία	Υψηλό	< 100 m	Ναι	Όχι αναγκαία	Ναι π.χ. δυνατότητα ενσωματωμένου αισθητήρα με datalogging, κρυπτογράφηση, authentication

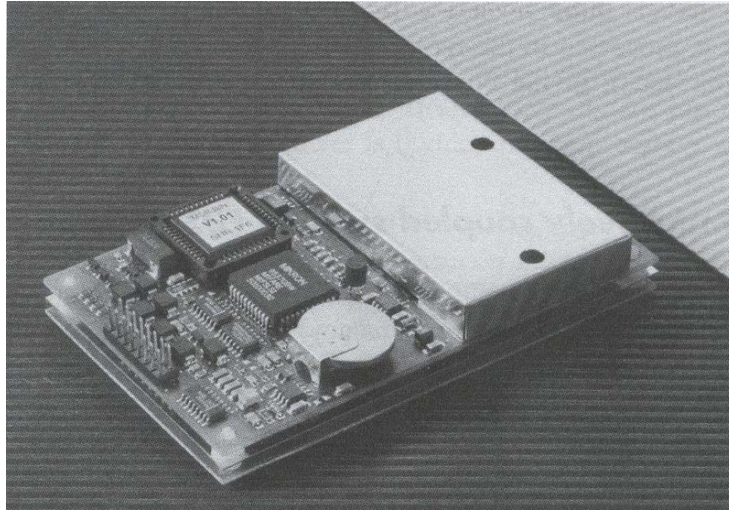
2.4.2 Αναγνώστης

2.4.2.1 Χαρακτηριστικά

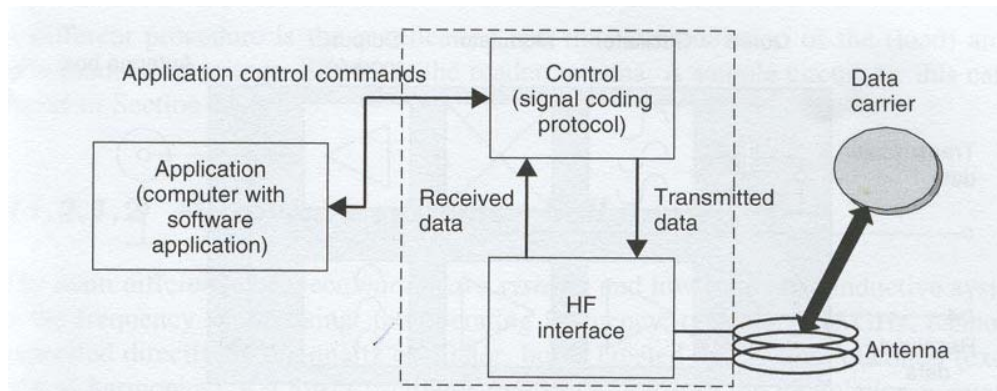
Ο RFID αναγνώστης είναι η συσκευή που επικοινωνεί μέσω των ραδιοκυμάτων με τις ετικέτες που βρίσκονται μέσα στο πεδίο εμβέλειάς τους. Οι βασικές του λειτουργίες είναι η ανάγνωση των δεδομένων των ετικετών, η εγγραφή δεδομένων στις ετικέτες εφόσον είναι επανεγγράψιμοι, η τροφοδότηση των παθητικών και ημιπαθητικών ετικετών και η αμφίδρομη μεταφορά δεδομένων με το λογισμικό. Εκτός από αυτές τις βασικές λειτουργίες κάποιοι RFID αναγνώστες είναι απαραίτητοι για την αποφυγή συγκρούσεων (anti-collision) στην περίπτωση ταυτόχρονης ανάγνωσης πολλών ετικετών, την κρυπτογράφηση δεδομένων και την αυθεντικοποίηση του χρήστη (authentication).

Κάθε RFID αναγνώστης αποτελείται από:

- 1) Μια κεραία
- 2) Μια μονάδα ελέγχου (control system)
- 3) Μια μονάδα υψηλών συχνοτήτων (HF ή RF interface)



Σχήμα 2.14: Αναγνώστης (στο αριστερό μέρος διακρίνεται η μονάδα ελέγχου ενώ στο δεξιό η μονάδα υψηλών συχνοτήτων)



Σχήμα 2.15: Μπλοκ διάγραμμα ενός αναγνώστη

Μονάδα ελέγχου

Η μονάδα ελέγχου είναι υπεύθυνη για τις εξής λειτουργίες [23]:

- επικοινωνία του αναγνώστη με το λογισμικό (για αυτό το λόγο έχει μια θύρα RS232 ή RS485)
- έλεγχο της επικοινωνίας με την ετικέτα μέσω συγκεκριμένων πρωτοκόλλων επικοινωνίας
- κωδικοποίηση και αποκωδικοποίηση σημάτων

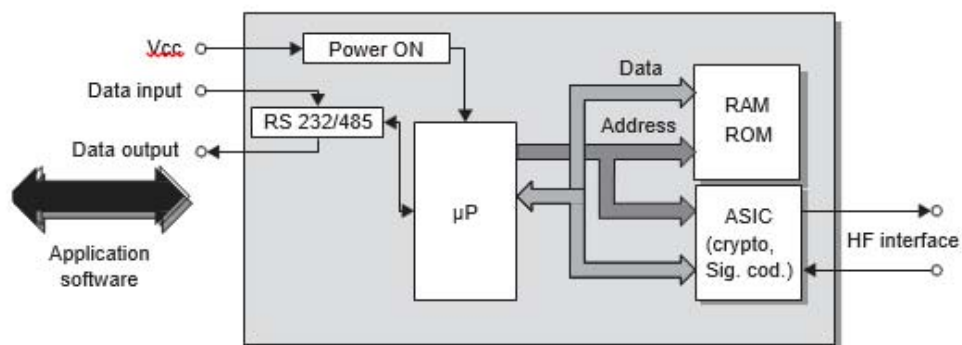
Επίσης είναι υπεύθυνη για τις σύνθετες λειτουργίες του αναγνώστη που προαναφέρθηκαν, δηλαδή:

- την εκτέλεση ενός αλγορίθμου αποφυγής σύγκρουσης (anti-collision algorithm)

- την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που μεταφέρονται μεταξύ ετικέτας και αναγνώστη και τον έλεγχο λαθών (error checking)
- αυθεντικοποίηση μεταξύ ετικέτας και αναγνώστη

Για να πραγματοποιηθούν αυτές οι λειτουργίες η μονάδα ελέγχου αποτελείται από τα εξής [24]:

- Ένα μικροεπεξεργαστή, ο οποίος λαμβάνει δεδομένα και ανάλογα με το πρωτόκολλο που απαιτείται, θα ψάξει τη μνήμη του για τον αντίστοιχο κώδικα του προγράμματος και θα το εκτελέσει.
- Τον ελεγκτή (controller), ο οποίος είναι απαραίτητος για τη μετατροπή των δεδομένων που λαμβάνονται σε δυαδικό κώδικα.
- Τη διεπαφή επικοινωνίας (communication interface), που μπορεί να είναι μέρος του ελεγκτή ή ανεξάρτητη και χρησιμεύει για να μπορεί ο αναγνώστης να επικοινωνεί με το λογισμικό.
- Τη μνήμη, ο ρόλος της οποίας είναι η αποθήκευση των δεδομένων που λαμβάνονται από την ετικέτα.
- Κανάλια εισαγωγής/εξαγωγής για εξωτερικούς αισθητήρες (input/output channels for external sensors). Όταν οι ετικέτες είναι εκτός του πεδίου εμβέλειας του αναγνώστη, οδηγούμαστε στην άσκοπη κατανάλωση ενέργειας για την αναγνώρισή τους. Για να λυθεί αυτό το πρόβλημα, μπορούμε να χρησιμοποιήσουμε εξωτερικούς αισθητήρες οι οποίοι τοποθετούνται σε αυτά τα κανάλια και ενεργοποιούν τον αναγνώστη όταν χρειάζεται.



Σχήμα 2.16: Μονάδα ελέγχου

Μονάδα υψηλών συχνοτήτων HF

Η μονάδα HF είναι υπεύθυνη για τις εξής λειτουργίες [23]:

- Παραγωγή ισχύος υψηλής συχνότητας για την ενεργοποίηση και την τροφοδοσία των ετικετών.
- Διαμόρφωση του σήματος μετάδοσης δεδομένων προς την ετικέτα
- Λήψη και αποδιαμόρφωση των ραδιοσημάτων από την ετικέτα

Για να επιτύχει αυτή την αμφίδρομη επικοινωνία μεταξύ αναγνώστη-ετικέτας, αποτελείται από ένα δέκτη (receiver) και έναν αναμεταδότη (transmitter) [24] καθώς και από μια πηγή ενέργειας. Επιπλέον καλύπτεται από λευκοσίδηρο, για την προστασία από ανεπιθύμητες εκπομπές σημάτων.

2.4.2.2 Κατηγοριοποίηση

Με βάση τις εφαρμογές για τις οποίες προορίζονται οι αναγνώστες και τις ιδιότητες τους έχουμε τις εξής κατηγορίες:

- Φορητοί: Οι φορητοί αναγνώστες έχουν ενσωματωμένη την κεραία, μια LCD οθόνη και έχουν σχετικά μεγάλη χωρητικότητα. Ως επί το πλείστον έχουν μια θύρα για να μπορούν να συνδεθούν σε κάποιο υπολογιστή και χρησιμοποιούνται στην ταυτοποίηση ζώων και σε βιομηχανικές εφαρμογές [23].



Σχήμα 2.17: Φορητοί αναγνώστες

- Σταθεροί: Οι σταθεροί αναγνώστες περιέχουν 2-8 κεραίες, μπορούν να τοποθετηθούν σε διάφορες περιοχές όπως σε τοίχους και πόρτες. Χρησιμοποιούνται σε αποθήκες, εργοστάσια, σε ταινίες μεταφοράς προϊόντων [29].



Σχήμα 2.18: Σταθερός αναγνώστης

- Ολοκληρωμένοι αναγνώστες: Πρόκειται για αναγνώστες σε μορφή ολοκληρωμένου κυκλώματος και χρησιμοποιούνται σε εισόδους καταστημάτων ή άλλων κτιρίων [23].

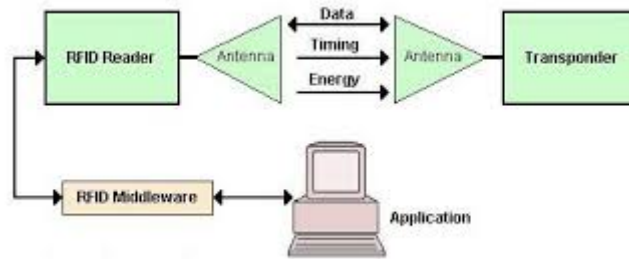


Σχήμα 2.19: Ολοκληρωμένοι αναγνώστες

2.4.3 Ενδιάμεσο Λογισμικό

Ο “εγκέφαλος” ενός συστήματος RFID θεωρείται το ενδιάμεσο λογισμικό του συστήματος (RFID Middleware). Ο ρόλος του είναι η επεξεργασία των δεδομένων που λαμβάνονται και αποστέλλονται από κάθε ετικέτα, η προώθηση προς τον αναγνώστη των δεδομένων και των οδηγιών που δέχεται από το πληροφοριακό σύστημα, καθώς και το αντίστροφο, η αποστολή δεδομένων και οδηγιών από τον αναγνώστη προς το πληροφοριακό σύστημα. Οι οδηγίες αυτές μπορούν να αφορούν τις λειτουργίες που πρέπει να γίνουν στις ετικέτες όπως η αναγνώριση της ύπαρξης μίας ετικέτας, η ανάγνωση του κωδικού μίας ετικέτας, ανάγνωση/εγγραφή των δεδομένων μίας ετικέτας αλλά και λειτουργίες που αφορούν τον ίδιο τον αναγνώστη, όπως η αλλαγή ρυθμίσεων και η ανάγνωση της κατάστασης του.

Έτσι μέσω του ενδιάμεσου λογισμικού βελτιστοποιείται η λειτουργικότητα όλου του συστήματος και επιτυγχάνεται η συνεχής συλλογή και αποθήκευση δεδομένων, ώστε να είναι διαθέσιμο ανά πάσα στιγμή το ιστορικό των διαδικασιών. Τέλος είναι δυνατή η εξαγωγή στατιστικών ή άλλων αναφορών, με ακριβή στοιχεία [30].



Σχήμα 2.20: Ρόλος του ενδιάμεσου λογισμικού σε ένα RFID σύστημα

2.5 Εφαρμογές της RFID τεχνολογίας

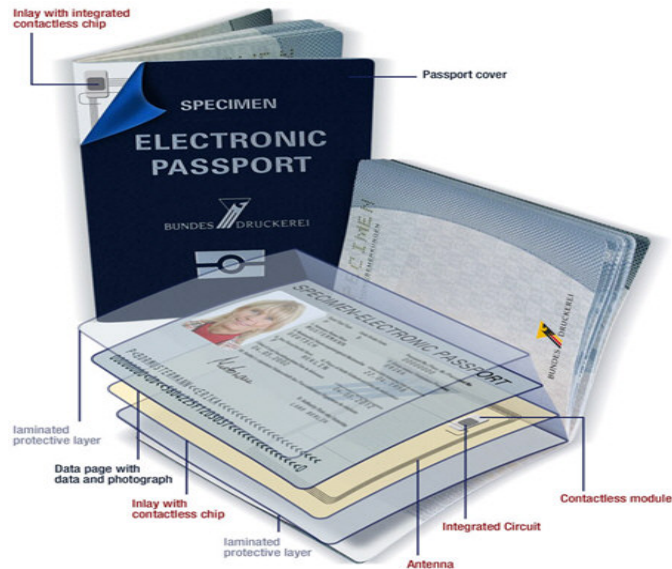
Στη σημερινή εποχή η τεχνολογία RFID έχει πολλές εφαρμογές. Σε αυτό το κεφάλαιο παρουσιάζονται κάποιες από τις σημαντικότερες:

- **Βιβλιοθήκες:** Σε κάθε βιβλίο επικολλάται μια ετικέτα RFID που περιέχει πληροφορίες όπως τον τίτλο του βιβλίου, το συγγραφέα, τις ημερομηνίες δανεισμού και επιστροφής ενώ στις πόρτες της βιβλιοθήκης οι κατάλληλοι RFID αναγνώστες. Έτσι αν κάποιος προσπαθήσει να εξέλθει από τη βιβλιοθήκη με βιβλίο που δεν έχει δανειστεί, θα γίνει αμέσως αντιληπτός.



Σχήμα 2.21: Εφαρμογή της τεχνολογίας RFID σε μια βιβλιοθήκη

- **Πιστωτικές κάρτες:** Στις μέρες μας όλο και περισσότερες πιστωτικές κάρτες χρησιμοποιούν την τεχνολογία RFID.
- **Ηλεκτρονικά διαβατήρια (e-passports):** Στα διαβατήρια αυτά (που ονομάζονται αλλιώς και βιομετρικά) ενσωματώνονται microchip RFID τα οποία περιλαμβάνουν προσωπικά δεδομένα του κατόχου σε ηλεκτρονική μορφή.



Σχήμα 2.22: Ηλεκτρονικό διαβατήριο

- **Πληρωμή διοδίων:** Οι κάτοχοι κάρτας που έχει ενσωματωμένη RFID ετικέτα γλυτώνουν την αναμονή στις ουρές των διοδίων. Στην Ελλάδα εφαρμόζεται το σύστημα e-Pass στην Αττική οδό και το σύστημα TEO PASS στις εθνικές οδούς.
- **Εισιτήρια Μέσων Μαζικής Μεταφοράς:** Σε πολλές χώρες τα χάρτινα εισιτήρια έχουν αντικατασταθεί από RFID tags που περιέχουν πληροφορίες για τον τύπο και την ημερομηνία λήξης του εισιτηρίου π.χ. η Oyster card στον υπόγειο του Λονδίνου.
- **Αυτοκίνητα:** Το 2004 οι εταιρίες Toyota και Lexus δημιούργησαν τα ηλεκτρονικά κλειδιά μηχανής (smart keys) τα οποία ήταν τεχνολογίας RFID. Τα συγκεκριμένα κλειδιά δίνουν τη δυνατότητα στον οδηγό να ανοίξει την πόρτα και τη μηχανή ενώ έχει τα κλειδιά στην τσέπη του καθώς αναγνωρίζουν την παρουσία του σε πολύ μικρή απόσταση από τον αναγνώστη.
- **Χρήση των RFID tags για τον εντοπισμό παλετών σε shipping containers**
- **Αποθήκες:** Με τη χρήση της τεχνολογίας RFID διευκολύνεται ο έλεγχος των αποθεμάτων, της διακίνησης των εμπορευμάτων και η ανίχνευση και ταυτοποίηση προϊόντων κατά τη μεταφορά.
- **Φυλακές:** Ειδικές RFID ετικέτες τοποθετούνται στους κρατούμενους για την καλύτερη παρακολούθησή τους.
- **Αθλητισμός:** Πάνω στα παπούτσια των δρομέων εφαρμόζονται chips που περιέχουν RFID tags και δίνουν τη δυνατότητα καταγραφής των χρόνων τους όταν περνάνε από τα σταθερά χαλάκια κατά μήκος της διαδρομής.
- **Έλεγχος πρόσβασης (Access control):** Σε χώρους και κτίρια που είναι αναγκαία να περιοριστεί η πρόσβαση σε συγκεκριμένους ανθρώπους π.χ. σε ειδικευμένο προσωπικό. Χρησιμοποιούνται σύγχρονα συστήματα ελέγχου πρόσβασης που βασίζονται στην RFID τεχνολογία. Στα σημεία εισόδου τοποθετούνται RFID αναγνώστες οι οποίοι επικοινωνούν με σύστημα που ελέγχει την πόρτα ενώ τα άτομα που μπορούν να έχουν πρόσβαση προμηθεύονται με RFID tags. Έτσι όταν κάποιος πλησιάσει την πόρτα και ο σειριακός αριθμός της ετικέτας που διαθέτει αντιστοιχεί σε εξουσιοδοτημένο άτομο, ανοίγει η πόρτα.



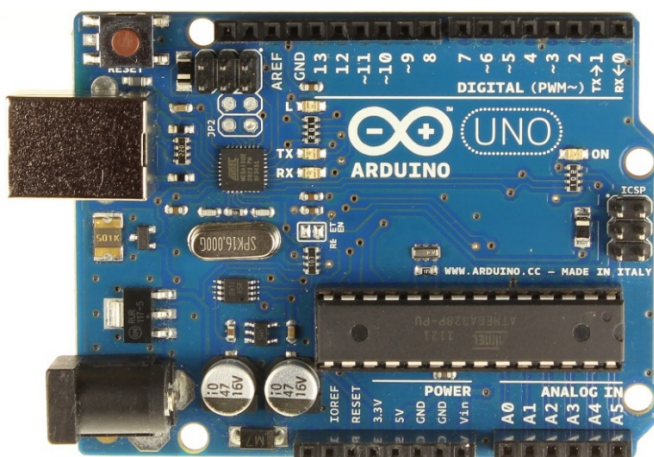
Σχήμα 2.23: Έλεγχος πρόσβασης με χρήση της τεχνολογίας RFID

- **Διαχείριση και καταγραφή ζώων:** Με την τοποθέτηση RFID ετικετών σε ζώα είναι δυνατή η παρακολούθηση και η καταγραφή ζώων, η καταχώρηση πληροφοριών για αυτά όπως εμβολιασμοί, ασθένειες και προέλευση και ο έλεγχος για πιθανές επιδημίες. Στην Ελλάδα ξεκίνησε το 2003 η καταγραφή όλων των ιδιόκτητων σκύλων ενώ η Ταϊβανέζικη κυβέρνηση το 2006 υλοποίησε ένα μεγάλο project για την τοποθέτηση RFID tags στα αποδημητικά πουλιά για να αντιμετωπίσει τη γρίπη των πτηνών.
- **Χιονοδρομικά κέντρα:** Οι επισκέπτες χιονοδρομικών κέντρων σε κάποιες χώρες, όπως στην Ελβετία, προμηθεύονται αντί για εισιτήρια με χιονοδρομικά δελτία που έχουν ενσωματωμένα RFID tags σε μέγεθος πιστωτικής κάρτας. Έτσι οι τηλεχειριζόμενες πύλες στα αναβατόρια που είναι εφοδιασμένες με αναγνώστες, μόλις εντοπίσουν τις ετικέτες που είναι σε ισχύ, ανοίγουν αυτόματα. Τα συγκεκριμένα δελτία μπορούν να χρησιμοποιηθούν και για τον εντοπισμό των χιονοδρόμων σε περίπτωση τραυματισμού.
- **Φαρμακευτικές εταιρίες:** Η χρήση της τεχνολογίας RFID στη φαρμακοβιομηχανία διευκολύνει τη διανομή των φαρμάκων, αποτρέπει την κυκλοφορία παράνομων και επικίνδυνων σκευασμάτων.
- **Παρακολούθηση ασθενών στα νοσοκομεία:** Κάθε ασθενής εφοδιάζεται με βραχιόλι που έχει ενσωματωμένη RFID ετικέτα της οποίας ο μοναδικός σειριακός αριθμός συνδέεται με τον ηλεκτρονικό του φάκελο.
- **Συστήματα αυτοματοποιημένης ιατρικής και φαρμακευτικής περίθαλψης.**

3. ARDUINO

3.1 Γενικά για το Arduino UNO

Για την υλοποίηση της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε η πλατφόρμα Arduino UNO που βασίζεται στο μικροεπεξεργαστή ATmega328. Το Arduino είναι στην ουσία μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή ανοιχτού κώδικα μητρική πλακέτα, έχει ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Ο μικροελεγκτής του προγραμματίζεται χρησιμοποιώντας τη γλώσσα προγραμματισμού Arduino, η οποία βασίζεται στη Wiring. Η γλώσσα Wiring ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες υλοποιημένες στη C++. Οι υλοποιήσεις του Arduino (που πρωτοεμφανίστηκε το 2005 στην Ιταλία για εκπαιδευτικούς σκοπούς [31]) μπορούν να λειτουργήσουν αυτόνομα ή να επικοινωνούν με τον υπολογιστή μέσω προγραμμάτων σε Processing, Flash, Max/MSP. Επιπλέον, μπορεί να αλληλοεπιδρά με το περιβάλλον λαμβάνοντας δεδομένα από διάφορους αισθητήρες και να το επηρεάσει ελέγχοντας LEDs, διακόπτες ή κινητήρες που είναι συνδεδεμένοι σε αυτόν.



Σχήμα 3.1: Arduino UNO

3.2 Τεχνικά χαρακτηριστικά του Arduino UNO

Το Arduino UNO αποτελείται από ένα μικροεπεξεργαστή, 14 ψηφιακές εισόδους/εξόδους (από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι), 6 αναλογικές εξόδους, έναν κεραμικό αντηχητή 16 MHz (ή έναν κρυσταλλικό ταλαντωτή σε κάποιες παραλλαγές), μια σύνδεση USB, βύσμα τροφοδοσίας, ICSP header και ένα διακόπτη επαναφοράς (reset button). Στον πίνακα που ακολουθεί παρουσιάζονται τα τεχνικά χαρακτηριστικά του Arduino UNO.

Πίνακας 3.1: Τεχνικά χαρακτηριστικά Arduino UNO

Μικροεπεξεργαστής	ATmega328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες εισόδου/εξόδου	14 (από τις οποίες οι 6 παρέχουν PWM έξοδο)
Αναλογικοί ακροδέκτες εισόδου	6
DC ρεύμα ανά ακροδέκτη	40 mA
DC ρεύμα για ακροδέκτη τάσης 3.3V	50 mA
Μνήμη flash	32 KB(ATmega328) από τα οποία 0,5 KB χρησιμοποιούνται από το bootloader
Μνήμη SRAM	2 KB(ATmega328)
Μνήμη EEPROM	1 KB(ATmega328)
Ταχύτητα ρολογιού	16 MHz

3.3 Φυσικά χαρακτηριστικά του Arduino UNO

Οι μέγιστες διαστάσεις του Arduino UNO είναι 2,7 ίντσες σε μήκος και 2,1 ίντσες σε πλάτος χωρίς να συμπεριλαμβάνονται σε αυτές η υποδοχή USB και το βύσμα τροφοδοσίας που προεξέχουν.

3.4 Τροφοδοσία

Το Arduino UNO μπορεί να τροφοδοτηθεί είτε μέσω σύνδεσης USB από τον υπολογιστή είτε μέσω εξωτερικής τροφοδοσίας (π.χ. μια μπαταρία ή έναν αντάπτορα AC-DC). Παρόλο που τα ανεκτά όρια της εξωτερικής τροφοδοσίας είναι 6 με 20 V, η προτεινόμενη τροφοδοσία για την αποφυγή προβλημάτων είναι 7 έως 12 V. Συγκεκριμένα για τροφοδοσία μικρότερη των 7 V η πλακέτα παρουσιάζει αστάθεια, ενώ για τροφοδοσία μεγαλύτερη των 12 V προκαλείται υπερθέρμανση του ρυθμιστή τάσεως και κατά συνέπεια καταστροφή της πλακέτας. Οι ακροδέκτες τροφοδοσίας είναι οι ακόλουθοι [32]:

- VIN: Η τάση εισόδου της πλατφόρμας όταν χρησιμοποιείται εξωτερική πηγή. Η τροφοδοσία τάσης γίνεται επίσης μέσω αυτού του ακροδέκτη καθώς και η σύνδεση του βύσματος τροφοδοσίας ή μπαταρίας.
- 5V: Αυτός ο ακροδέκτης παρέχει τη σταθερή τάση των 5 V μέσω του ρυθμιστή στα διάφορα μέρη της πλακέτας και το μικροεπεξεργαστή. Η τάση αυτή παρέχεται

είτε από τη σύνδεση USB ή άλλη ρυθμιζόμενη πηγή τροφοδοσίας, είτε από τον ακροδέκτη VIN (7-12 V). Συνιστάται η πλακέτα να μην τροφοδοτείται μέσω των ακροδεκτών 5 V ή 3V3 γιατί έτσι παρακάμπτεται ο ρυθμιστής και μπορεί να καταστραφεί η πλακέτα.

- 3V3: 3.3 V τάση τροφοδοσίας της πλατφόρμας (μέγιστο ρεύμα κατανάλωσης 50 mA)
- GND: Ακροδέκτες γείωσης
- IOREF: Ο συγκεκριμένος ακροδέκτης παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροεπεξεργαστής. Έτσι, ένα καταλλήλως ρυθμισμένο shield (βλέπε κεφάλαιο 3.10) μπορεί να διαβάσει την τάση του ακροδέκτη IOREF και να επιλέξει την κατάλληλη πηγή τροφοδοσίας.

3.5 Μνήμη

Ο μικροεπεξεργαστής ATmega328 αποτελείται από 3 κατηγορίες μνήμης [32]:

- i. 32 KB μνήμης flash (από τα οποία τα 0,5 χρησιμοποιούνται για τον bootloader). Η μνήμη flash χρησιμοποιείται για την αποθήκευση των προγραμμάτων αφού πρώτα μεταγλωττιστούν στον υπολογιστή και δεν χάνει τα περιεχόμενα σε περίπτωση επανεκκίνησης ή διακοπής τροφοδοσίας.
- ii. 2 KB SRAM (Static Random Access Memory): Στατική μνήμη RAM που χρησιμοποιείται για την αποθήκευση μεταβλητών, πινάκων κλπ. Σε περίπτωση επανεκκίνησης ή απώλειας τροφοδοσίας χάνει τα δεδομένα της.
- iii. 1 KB EEPROM (Electrically Erasable Programmable Read-Only Memory): Χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών καθώς δε χάνει τα δεδομένα σε περίπτωση επανεκκίνησης ή διακοπής τροφοδοσίας.

3.6 Ακροδέκτες

Το Arduino UNO διαθέτει 20 ακροδέκτες από τους οποίους όπως προαναφέρθηκε οι 14 είναι ψηφιακοί και οι 6 αναλογικοί [32].

Ψηφιακοί: Οι 14 ψηφιακοί ακροδέκτες που λειτουργούν στα 5 V μπορεί να χρησιμοποιηθούν και ως είσοδοι και ως έξοδοι. Το μέγιστο ρεύμα που μπορεί να προσφέρει ή να λαμβάνει ο κάθε ακροδέκτης είναι 40 mA. Στη συνέχεια παρουσιάζονται κάποιες εξειδικευμένες λειτουργίες τους.

- Ακροδέκτης 0(RX) και 1(TX): Αυτοί οι ακροδέκτες χρησιμοποιούνται για να λαμβάνουν (RX) και να μεταφέρουν (TX) σειριακά δεδομένα.
- Ακροδέκτες 2,3: Λειτουργούν ως εξωτερικά interrupts. Ρυθμίζονται από το πρόγραμμα για να προκαλέσουν μια διακοπή σε περίπτωση χαμηλής τιμής ή αλλαγή τιμής.

- Ακροδέκτες 3,5,6,9,10,11: Παρέχουν 8-bit έξοδο PWM (Pulse Width Modulation) μέσω της εντολής analogWrite().
- SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK): Οι συγκεκριμένοι ακροδέκτες υποστηρίζουν SPI επικοινωνία χρησιμοποιώντας την SPI βιβλιοθήκη.
- LED 13: Στον ακροδέκτη 13 υπάρχει ενσωματωμένο ένα LED το οποίο είναι αναμμένο όταν η τιμή του ακροδέκτη είναι HIGH ενώ είναι σβηστό όταν η τιμή είναι LOW.

Αναλογικοί: Το Arduino UNO έχει 6 αναλογικούς ακροδέκτες αριθμημένοι από το A0 έως το A5. Ο καθένας παρέχει ανάλυση 10 bits και κατ' επέκταση επιστρέφονται ακέραιοι από 0 έως 1023. Επίσης μετρούν τάσεις από 0 έως 5 V ενώ το άνω όριο τάσης μπορεί να αλλαχθεί χρησιμοποιώντας τον ακροδέκτη AREF και την εντολή analogReference(). Ακολουθούν κάποιες εξειδικευμένες λειτουργίες τους.

- TWI: Ακροδέκτες A4/SDA και A5/SCL. Υποστηρίζουν TWI επικοινωνία μέσω της βιβλιοθήκης WIRE

Κάποιοι άλλοι ακροδέκτες:

- AREF: Η τάση αναφοράς για τις αναλογικές εισόδους που χρησιμοποιείται με την εντολή analogReference()
- Reset: Χρησιμοποιείται για να προσθέσει ένα διακόπτη επαναφοράς όταν υπάρχει shield και μπλοκάρει το αντίστοιχο της πλακέτας.

3.7 Προστασία

Παρόλο που οι υπολογιστές έχουν τη δική τους εσωτερική ασφάλεια, το Arduino UNO για έξτρα προστασία έχει δικιά του (resettable polyfusion). Όταν στη θύρα USB εφαρμοστούν πάνω από 500 mA πέφτει η ασφάλεια μέχρι να λυθεί το πρόβλημα. Με αυτόν τον τρόπο οι θύρες USB του υπολογιστή προστατεύονται από υπερτάσεις και υψηλά ρεύματα.

3.8 Επικοινωνία

Ο ATmega328 παρέχει UART TTL 5V (Transistor-Transistor Logic) σειριακή επικοινωνία η οποία είναι διαθέσιμη στους ακροδέκτες 0 και 1. Το UART (Universal Asynchronous Receiver Transmitter) είναι ένα κύκλωμα των υπολογιστών το οποίο διαμεσολαβεί στην σειριακή επικοινωνία υπολογιστών ή υπολογιστών με συσκευές [33].

Το λογισμικό του Arduino συμπεριλαμβάνει μια σειριακή οθόνη η οποία επιτρέπει να σταλθούν από και προς την πλατφόρμα απλά δεδομένα κειμένου. Όταν τα δεδομένα μεταδίδονται μέσω USB στον υπολογιστή, τα RX και TX LEDs αναβοσβήνουν. Αυτό βεβαίως δε συμβαίνει όταν υπάρχει σειριακή επικοινωνία στους ακροδέκτες 0 και 1. Η σειριακή επικοινωνία σε οποιοδήποτε ψηφιακό ακροδέκτη γίνεται εφικτή με τη βοήθεια της βιβλιοθήκης SoftwareSerial.

Επίσης ο ATmega328 υποστηρίζει I2C/TWI (Two Wire Interface) και SPI (Serial Peripheral Interface) επικοινωνία. Το TWI ή I2C είναι το πρωτόκολλο που δημιούργησε η εταιρία Philips για τη σύνδεση περιφερειακών συσκευών χαμηλής ταχύτητας και

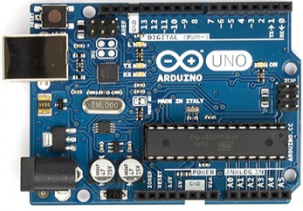

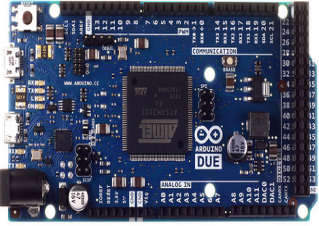








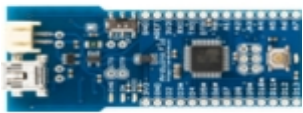
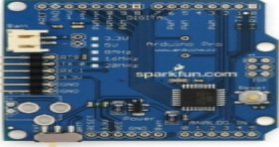
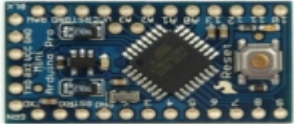
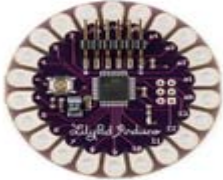



χρησιμοποιεί 2 αμφίδρομες γραμμές επικοινωνίας [34]. Με τον όρο SPI αναφερόμαστε στο πρωτόκολλο που επιτρέπει την αμφίδρομη σειριακή επικοινωνία [32].

3.9 Μοντέλα Arduino

Όπως προαναφέρθηκε, στην παρούσα διπλωματική εργασία επιλέχθηκε να χρησιμοποιηθεί το Arduino UNO. Οι λόγοι ήταν ότι κάλυπτε όλες τις ανάγκες της εργασίας, ότι είναι απλό στη χρήση του και ότι έχει χαμηλό κόστος. Γενικά όμως στην αγορά κυκλοφορούν και άλλα μοντέλα Arduino που παρουσιάζονται στο σχήμα 3.2. Τα μοντέλα αυτά παρουσιάζουν αρκετές διαφορές όσο αναφορά τα χαρακτηριστικά τους π.χ. στους ακροδέκτες η στην τάση εισόδου και εξόδου. Ακολουθεί ένας πίνακας με τα χαρακτηριστικά των διαφορετικών μοντέλων.

Πίνακας 3.2: Χαρακτηριστικά διάφορων μοντέλων Arduino

Όνομα	Επεξεργαστής	Τάση λειτουργίας/ Τάση τροφοδοσίας	CPU ταχύτητα	Αναλογικοί ακροδέκτες εισόδου/εξόδου	Ψηφιακοί ακροδέκτες εισόδου/εξόδου	EEPROM (KB)	SRAM (KB)	Flash (KB)	USB	UART
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	0.512 1	1 2	16 32	Mini-B	1
Ethernet	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16 Mhz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8 Mhz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35 -12 V	8 Mhz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35 -12 V 5 V/5-12 V	8 Mhz 16Mhz	6/0	14/6	0.512	1	16	-	1
LilyPad	ATmega168V ATmega328V	2.7-5.5 V/ 2.7- 5.5 V	8 Mhz	6/0	14/6	0.512	1	16	-	-
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8 Mhz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/ 2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/ 2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-	-

		
UNO	Leonardo	Due
		
Micro	Mega 2560	Mega ADK
		
Mini	Nano	Ethernet
		
Esplora	Arduino BT	Fio
		
Pro	Pro Mini	LilyPad
		
LilyPad USB	LilyPad Simple	LilyPad SimpleSnap

Σχήμα 3.2: Μοντέλα Arduino

3.10 Shields

Με τον όρο shield (ασπίδα), αναφερόμαστε στην πλακέτα που μπορεί να συνδεθεί στην κορυφή των μικροεπεξεργαστών Arduino και να επεκτείνει τις δυνατότητες του κυρίως σε θέματα επικοινωνίας. Κάποια από τα μοντέλα shields παρουσιάζονται παρακάτω [32]:

- **Arduino WiFi Shield:** Συνδέει την πλατφόρμα με το δίκτυο ασύρματα με βάση κάποιες απλές οδηγίες.



Σχήμα 3.3: Arduino WiFi Shield

- **Arduino Ethernet Shield:** Συνδέει την πλατφόρμα στο διαδίκτυο σε λίγα λεπτά με ένα καλώδιο RJ45 με βάση πάλι κάποιες απλές οδηγίες.



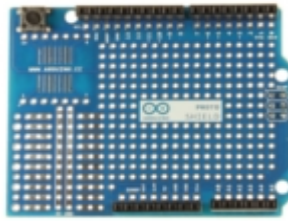
Σχήμα 3.4: Arduino Ethernet Shield

- **Wireless SD Shield:** Επιτρέπει στην πλατφόρμα να επικοινωνεί ασύρματα χρησιμοποιώντας μία ασύρματη μονάδα η οποία έχει εμβέλεια έως 100 πόδια σε εσωτερικούς χώρους και 300 πόδια σε εξωτερικούς.



Σχήμα 3.5: Wireless SD Shield

- **Wireless Proto Shield:** Καθιστά εύκολη τη σχεδίαση εξειδικευμένων κυκλωμάτων.



Σχήμα 3.6: Wireless Proto Shield

- **Arduino Motor Shield:** Επιτρέπει την οδήγηση επαγωγικών φορτίων όπως ρελέ, πηνίων, 2 DC κινητήρων ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά.



Σχήμα 3.7: Motor Shield

- **Arduino GSM Shield:** Συνδέει την πλατφόρμα στο διαδίκτυο χρησιμοποιώντας το ασύρματο δίκτυο GPRS.



Σχήμα 3.8: GSM Shield

3.11 Ethernet Shield

Στην παρούσα διπλωματική χρησιμοποιήθηκε το Arduino Ethernet Shield, το οποίο συνδέει την πλατφόρμα Arduino με το διαδίκτυο σε λίγα μόνο λεπτά και συνδέεται πάνω της χρησιμοποιώντας μακριούς συρμάτινους ακροδέκτες [32]. Βασίζεται στο Wiznet

W5100 Ethernet chip και μπορεί να ανταποκριθεί σε TCP και UDP. Επίσης υποστηρίζει έως και 4 ταυτόχρονες συνδέσεις.

Το Ethernet Shield έχει μια πρότυπη RJ-45 σύνδεση με ενσωματωμένο μετασχηματιστή γραμμής καθώς και μια υποδοχή κάρτας micro-SD, η οποία χρησιμοποιείται για την αποθήκευση αρχείων με σκοπό τη διακίνησή τους μέσω του δικτύου. Πάνω από το Ethernet Shield μπορεί να συνδεθεί άλλο shield.

Για την επικοινωνία με το Arduino, χρησιμοποιεί το SPI bus (μέσω του ακροδέκτη ICSP). Αυτό βρίσκεται στους ψηφιακούς ακροδέκτες 10, 11, 12, και 13 του Arduino Uno οι οποίοι δεν μπορούν να χρησιμοποιηθούν ως ακροδέκτες εισόδου/εξόδου. Ο ακροδέκτης 10 χρησιμοποιείται για να επιλέγεται ο ελεγκτής Ethernet W5100 ενώ ο ακροδέκτης 4 για την SD κάρτα.

Το Ethernet shield διαθέτει ένα διακόπτη επαναφοράς και τα ακόλουθα LEDs:

- PWR: Δείχνει ότι η πλακέτα Arduino και το shield τροφοδοτούνται.
- LINK: Δείχνει ότι υπάρχει σύνδεση δικτύου και αναβοσβήνει όταν γίνεται αποστολή ή λήψη δεδομένων.
- FULLD: Δείχνει ότι η σύνδεση δικτύου είναι full duplex (αμφίδρομη επικοινωνία).
- 100M: Δείχνει ότι υπάρχει σύνδεση 100 Mb/s.
- RX: Αναβοσβήνει όταν γίνεται λήψη δεδομένων.
- TX: Αναβοσβήνει όταν γίνεται αποστολή δεδομένων.
- COLL: Αναβοσβήνει όταν ανιχνεύονται συγκρούσεις δεδομένων (collisions).

4. Σχεδιασμός και υλοποίηση του συστήματος

4.1 Γενική περιγραφή του συστήματος

Το σύστημα που αναπτύχθηκε παρέχει τις εξής λειτουργίες:

- Εισαγωγή του αριθμού μητρώου του ασθενούς στην εφαρμογή.
- Ανάκτηση και προβολή στην οθόνη της διεπαφής των στοιχείων του ασθενούς καθώς και το σωστό μείγμα αναισθησίας. Τα στοιχεία αυτά ανακτώνται από τη βάση δεδομένων του συστήματος. Το σωστό μείγμα αναισθησίας που είναι καταχωρημένο στη βάση δεδομένων, έχει επιλεγθεί από τον υπεύθυνο αναισθησιολόγο τη μέρα πριν την εισαγωγή του ασθενούς στο χειρουργείο.
- Ανάγνωση της RFID ετικέτας που βρίσκεται πάνω σε κάθε μείγμα από τον RFID αναγνώστη. Κάθε RFID ετικέτα περιέχει ένα μοναδικό σειριακό αριθμό που αντιστοιχεί σε κάθε μείγμα. Η βάση δεδομένων περιλαμβάνει την αντιστοίχιση των σειριακών αριθμών RFID με τα μείγματα.
- Ενημέρωση μέσω της οθόνης διεπαφής αν το μείγμα είναι το σωστό ή λάθος και προβολή του σωστού μείγματος στην περίπτωση λάθους.

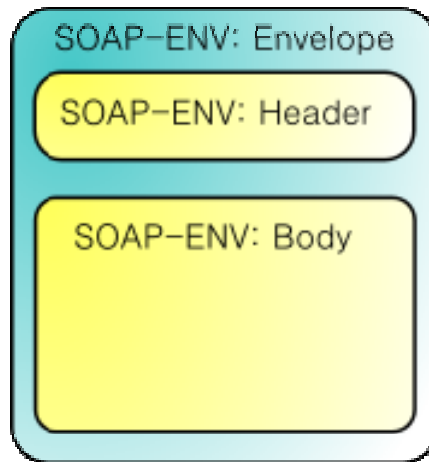
4.2 Στοιχεία του συστήματος

Το σύστημα αποτελείται από τα εξής δομικά στοιχεία:

Server

Για το server που επικοινωνεί με τη βάση δεδομένων και διαχειρίζεται την επικοινωνία με την πλατφόρμα Arduino και τη διαδραστική διεπαφή με το χρήστη, οι απαιτήσεις είναι να έχουμε τη δυνατότητα να αποστέλλουμε δεδομένα και να λαμβάνουμε δομημένες απαντήσεις (structured responses), οι οποίες περιλαμβάνουν προγραμματιστικά αντικείμενα. Για το σκοπό αυτό, επιλέχθηκε η κλήση απομακρυσμένων διαδικασιών (Remote Procedure Calls – RPC) χρησιμοποιώντας το πρωτόκολλο SOAP (Simple Object Access Protocol). Με αυτό τον τρόπο, είναι δυνατό οι πελάτες (clients) της υπηρεσίας να καλούν διαδικασίες του server, αποστέλλοντας τα ορίσματα και να δέχονται τα αποτελέσματα της κλήσης ως αντικείμενα, χρησιμοποιώντας το πρωτόκολλο SOAP.

Το SOAP είναι ένα πρωτόκολλο σχεδιασμένο για την ανταλλαγή δομημένων πληροφοριών σε υπηρεσίες web [35]. Τα μηνύματα SOAP αποτελούνται από ένα SOAP envelope, το οποίο περιλαμβάνει την επικεφαλίδα, με πληροφορίες για το τι περιλαμβάνει το μήνυμα και το σώμα, το οποίο περιλαμβάνει τα δεδομένα που θα αποσταλούν. Για το σώμα, το πρωτόκολλο SOAP χρησιμοποιεί τη γλώσσα XML (Extensible Markup Language), η οποία δομεί τα δεδομένα ως ένα ιεραρχικό δέντρο. Μια απεικόνιση του SOAP envelope μπορεί να βρεθεί στο σχήμα 4.1. Παραδείγματα επικοινωνίας χρησιμοποιώντας το πρωτόκολλο SOAP δίνονται στην περιγραφή των σεναρίων χρήσης του συστήματος.



Σχήμα 4.1: SOAP Envelope

Για τη δημιουργία του SOAP Service που εξυπηρετεί τα αιτήματα τόσο της πλατφόρμας Arduino, όσο και της διαδραστική διεπαφής, χρησιμοποιήθηκε η γλώσσα python. Η γλώσσα python είναι μια προγραμματιστική γλώσσα σεναρίων (scripting programming language), η οποία είναι ευρέως διαδεδομένη στον προγραμματισμό διαδικτυακών εφαρμογών, λόγω της ύπαρξης πληθώρας έτοιμων βιβλιοθηκών για ένα μεγάλο αριθμό λειτουργιών [36]. Βασικά χαρακτηριστικά της γλώσσας είναι η αναγνωσιμότητα των προγραμμάτων και το συντακτικό της, που επιτρέπει τη γραφή σύντομων προγραμμάτων. Η γλώσσα python υποστηρίζει πολλούς διαφορετικούς τρόπους προγραμματισμού, όπως τον προστακτικό προγραμματισμό (imperative programming), τον αντικειμενοστραφή προγραμματισμό (object-oriented programming) και τον συναρτησιακό προγραμματισμό (functional programming). Το σύστημα τύπων της είναι δυναμικό, το οποίο σημαίνει πως τα αντικείμενα δε χρειάζονται ορισμό του τύπου τους, αλλά ο τύπος προκύπτει από τη χρήση των αντικειμένων. Τέλος, η python χρησιμοποιεί αυτόματη διαχείριση μνήμης, το οποίο διευκολύνει τον προγραμματισμό, αφαιρώντας από τον προγραμματιστή την ανάγκη να δεσμεύει και να ελευθερώνει μνήμη για τα αντικείμενα που χρησιμοποιεί.

Για τη δημιουργία του εξυπηρετητή SOAP, χρησιμοποιήσαμε τη βιβλιοθήκη PySimpleSOAP (διαθέσιμη στο <https://code.google.com/p/pysimplesoap/>), η οποία παρέχει τις απαραίτητες δυνατότητες για τη δημιουργία και χρήση ενός εξυπηρετητή SOAP. Προσφέρει ευκολία προγραμματισμού, μπορεί να χρησιμοποιήσει διαλέκτους SOAP που χρησιμοποιούν διαφορετικές πλατφόρμες (Java Axis, .Net, WCF, JBoss, "jetty") και είναι επεκτάσιμη, χρησιμοποιώντας διαφορετικές βιβλιοθήκες επικοινωνίας με το δίκτυο, αναλόγως των αναγκών της εφαρμογής. Η ταχύτητα των εξυπηρετητών που δημιουργούνται με την εν λόγω βιβλιοθήκη είναι μεγάλη, με πολλές περιπτώσεις κατά τις οποίες είναι μία τάξη μεγέθους πιο γρήγορη από άλλες αντίστοιχες υλοποιήσεις.

Χρησιμοποιώντας τη βιβλιοθήκη PySimpleSOAP, η δημιουργία ενός εξυπηρετητή είναι πολύ απλή. Ορίζεται η διαδικασία που θα χρησιμοποιηθεί, δηλώνονται τα ορίσματά της και η βιβλιοθήκη αναλαμβάνει τα υπόλοιπα. Στο παρακάτω παράδειγμα, δηλώνεται η διαδικασία "adder", η οποία δέχεται δύο αριθμούς και επιστρέφει το άθροισμά τους. Δημιουργείται ένας εξυπηρετητής SOAP, ο οποίος δέχεται συνδέσεις στη θύρα 8008 και δηλώνεται η διαδικασία, τα ορίσματα και οι επιστρεφόμενες τιμές. Τέλος, ο εξυπηρετητής ενεργοποιείται και μπορεί πλέον να δεχτεί requests.

```

def adder(a,b):
    "Add two values"
    return a+b

dispatcher = SoapDispatcher(
    'my_dispatcher',
    location = "http://localhost:8008/",
    action = 'http://localhost:8008/', # SOAPAction
    namespace = "http://example.com/sample.wsdl", prefix="ns0",
    trace = True,
    ns = True)

# register the user function
dispatcher.register_function('Adder', adder,
    returns={'AddResult': int},
    args={'a': int, 'b': int})

httpd = HTTPServer(("", 8008), SOAPHandler)
httpd.dispatcher = dispatcher
httpd.serve_forever()

```

Ένα παράδειγμα επικοινωνίας ενός πελάτη με τον εξυπηρετητή είναι το παρακάτω. Αρχικά, ο πελάτης στέλνει ένα μήνυμα όπως το παρακάτω, με το οποίο ζητάει τη διαδικασία “Adder” του εξυπηρετητή, με ορίσματα a = 1 και b = 2.

```

POST http://localhost:8008/
SOAPAction: "http://localhost:8008/Adder"
Content-length: 329
Content-type: text/xml; charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
  <Adder xmlns="http://example.com/sample.wsdl">
    <a>1</a><b>2</b></Adder>
</soap:Body>
</soap:Envelope>

```

Η απάντηση του εξυπηρετητή φαίνεται παρακάτω, όπου ο εξυπηρετητής επιστρέφει ένα αντικείμενο AddResponse, του οποίου το πεδίο AddResult έχει την τιμή 3, που είναι και το αποτέλεσμα της κλήσης.

```

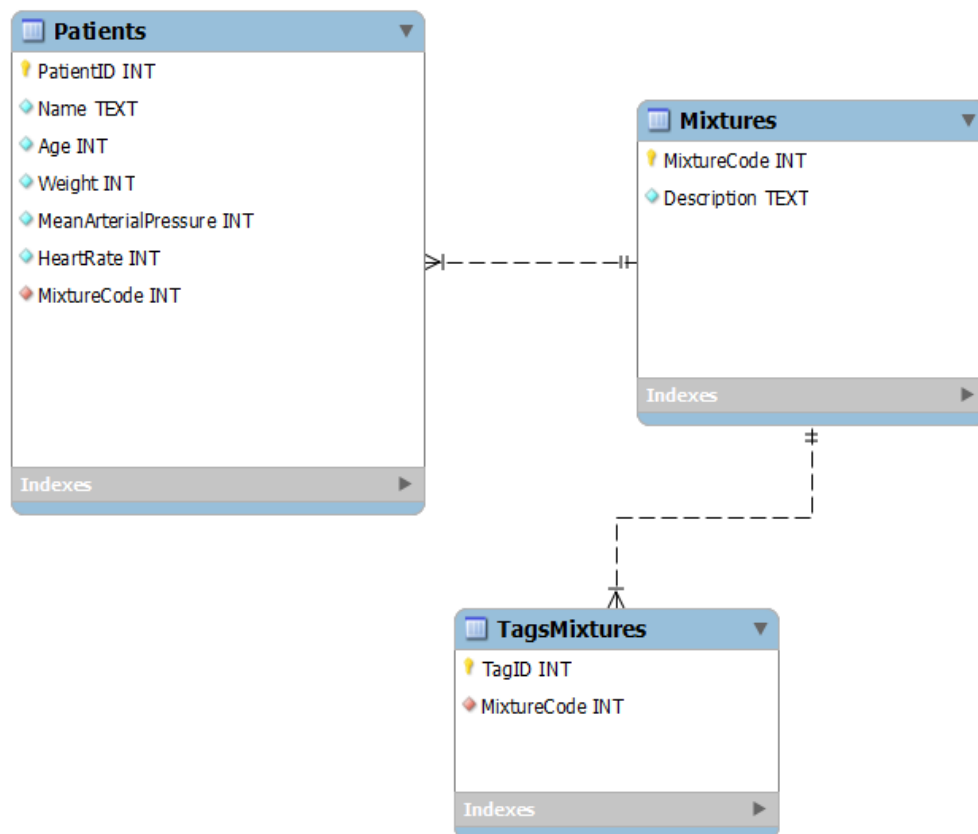
date: Tue, 20 Jul 2010 23:39:21 GMT
status: 200
content-type: text/xml
server: BaseHTTP/0.3 Python/2.5.4
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<AdderResponse xmlns="http://example.com/sample.wsdl">
<AddResult>3</AddResult>
</AdderResponse>
</soap:Body>
</soap:Envelope>

```

Βάση Δεδομένων

Για την αποθήκευση των δεδομένων των ασθενών, καθώς και των διαθέσιμων φαρμάκων, δημιουργήθηκε μια βάση δεδομένων. Για την υλοποίησή μας, χρησιμοποιήσαμε το σύστημα διαχείρισης βάσεων δεδομένων SQLite, το οποίο προσφέρει μια πλήρως λειτουργική, αυτόνομη βάση δεδομένων. Το σύστημα SQLite δε χρειάζεται ξεχωριστό εξυπηρετητή για να λειτουργήσει, αλλά ολόκληρη η βάση αποθηκεύεται σε ένα αρχείο, το οποίο μπορεί να λειτουργήσει σε διαφορετικές πλατφόρμες.

Το σχήμα (schema) της βάσης δεδομένων που δημιουργήσαμε φαίνεται στο Σχήμα 4.2. Αποτελείται από τρεις πίνακες. Ο πρώτος είναι ο πίνακας των ασθενών (Patients), ο οποίος περιλαμβάνει τα στοιχεία του ασθενή, καθώς και τον κωδικό του μείγματος αναισθησίας που ο αναισθησιολόγος έχει επιλέξει να χρησιμοποιηθεί. Ο δεύτερος πίνακας (Mixtures) περιλαμβάνει τα μείγματα αναισθησίας, με τον κωδικό του καθενός και την περιγραφή του (τα συστατικά και τις αναλογίες). Τέλος, ο τρίτος πίνακας (TagsMixtures) αντιστοιχεί τα RFID tags των μειγμάτων αναισθησίας με τον κωδικό του καθενός, δίνοντας έτσι τη δυνατότητα να υπάρχει το ίδιο μείγμα περισσότερες από μία φορές (και κατά συνέπεια σε διαφορετικά φιαλίδια με διαφορετικά tags).



Σχήμα 4.2: Σχήμα βάσης δεδομένων

Τα περιεχόμενα της βάσης δεδομένων δίνονται στο Παράρτημα Α, όπου δίνονται τα δεδομένα κάθε πίνακα και οι αντιστοιχίσεις ασθενών και μειγμάτων αναισθησίας.

Arduino Uno με RFID αναγνώστη και Ethernet Shield

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε όπως προαναφέρθηκε στο προηγούμενο κεφάλαιο η πλατφόρμα Arduino Uno συνδεδεμένη με Ethernet Shield και με RFID αναγνώστη. Οι λόγοι που επιλέχθηκαν τα συγκεκριμένα μοντέλα Arduino και Shield αντίστοιχα ήταν η απλότητα στη χρήση τους και ότι κάλυπταν όλες τις ανάγκες του συστήματος.

Ο RFID αναγνώστης που χρησιμοποιήθηκε είναι το μοντέλο RDM630 των 125 KHz ο οποίος υποστηρίζει εξωτερική κεραία και διαβάζει Read-only και Read-Write tags που βασίζονται στο πρωτόκολλο EM4100.



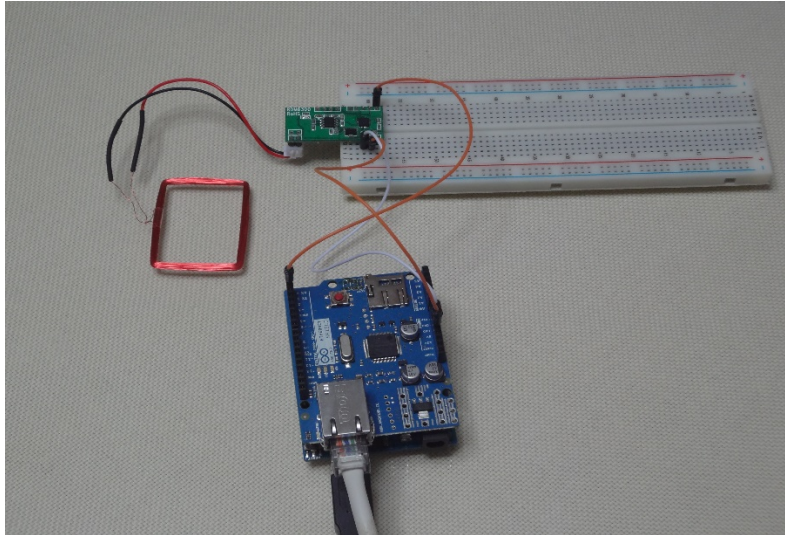
Σχήμα 4.3: RDM630 Reader

Τα χαρακτηριστικά του είναι τα εξής:

Πίνακας 4.1: Χαρακτηριστικά RDM630 Reader

Συχνότητα	125 kHz
Ρυθμός Baud	9600
Τύπος διεπαφής	επίπεδο TTL μορφή RS232
Εύρος ανάγνωσης	50mm
Τάση λειτουργίας	5V
Ρεύμα λειτουργίας	<50mA
Χρόνος αποκωδικοποίησης	<100ms

Η σύνδεση με το Arduino γίνεται σύμφωνα με το datasheet του RFID αναγνώστη (παράρτημα Β). Συγκεκριμένα ο δεύτερος και ο τρίτος ακροδέκτης του RFID αναγνώστη που βρίσκονται κάτω αριστερά (pins 2 και 3 στη θέση P3 του datasheet), συνδέονται με τους ακροδέκτες 5V και GND Arduino για την τροφοδοσία και τη γείωση αντίστοιχα, ενώ ο ακροδέκτης 1 του αναγνώστη (Tx) με τον ψηφιακό ακροδέκτη 2 του Arduino. Πάνω από την πλατφόρμα Arduino συνδέσαμε το Ethernet Shield, οπότε η παραπάνω συνδεσμολογία εφαρμόστηκε στους ακροδέκτες του Ethernet Shield. Στο σχήμα 4.4 φαίνεται η συνδεσμολογία που εφαρμόσαμε.



Σχήμα 4.4: Συνδεσμολογία Arduino-Ethernet Shield-RFID Reader

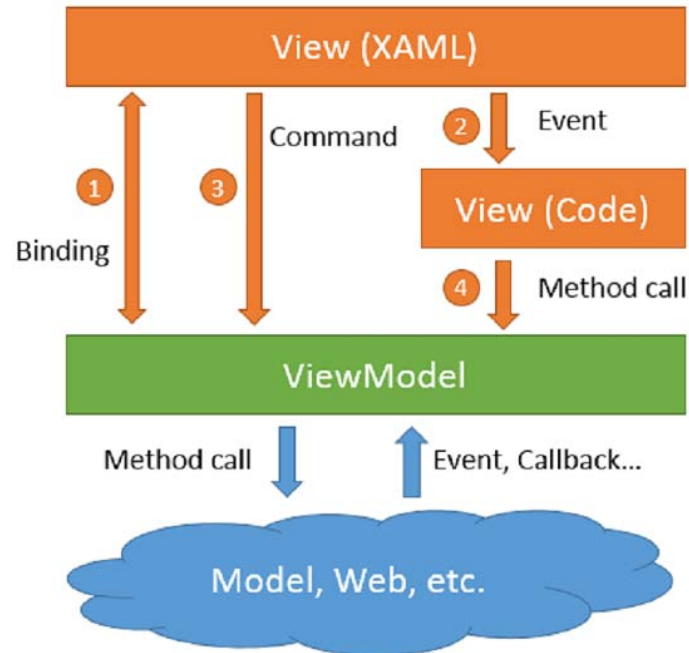
Ο RFID αναγνώστης χρησιμοποιήθηκε για να διαβάζει την ετικέτα που υπάρχει πάνω σε κάθε φιαλίδιο ενώ η πλατφόρμα Arduino για να αποστέλλει το ID της κάθε ετικέτας στον εξυπηρετητή SOAP Στο παράρτημα Γ παρατίθεται ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της λειτουργικότητας της πλατφόρμας Arduino.

Το Arduino επικοινωνεί με το δίκτυο μέσω του πρωτοκόλλου TCP. Το πρωτόκολλο TCP (Transmission Control Protocol - Πρωτόκολλο Ελέγχου Μεταφοράς) χαρακτηρίζεται από αξιόπιστη αποστολή και λήψη δεδομένων, καθώς και από τη μεταφορά τους χωρίς λάθη και με τη σωστή σειρά [37].

Διαδραστική Διεπαφή Χρήστη

Για τη διεπαφή χρήστη, προσανατολιστήκαμε σε μια εφαρμογή για οθόνες αφής, βασισμένη στην πλατφόρμα Windows, η οποία και μπορεί να εκτελεστεί τόσο σε all-in-one υπολογιστές, οι οποίοι ενσωματώνουν τον υπολογιστή στην οθόνη, όσο και σε tablets που χρησιμοποιούν λειτουργικό Windows. Η διεπαφή δημιουργήθηκε χρησιμοποιώντας το γραφικό υποσύστημα Windows Presentation Foundation (WPF), το οποίο είναι ένα υποσύστημα γραφικών που χρησιμοποιεί τις διεπαφές του DirectX για τη σχεδίαση των γραφικών. Είναι μέρος του .NET Framework και σχεδιάστηκε ώστε να ξεχωρίζει την δημιουργία της διεπαφής χρήστη από τη λογική της εφαρμογής. Το αρχιτεκτονικό υπόδειγμα που εισήχθη για να εκμεταλλευτεί τις δυνατότητες του WPF, με την ονομασία MVVM (Model View ViewModel) στοχεύει στο διαχωρισμό του γραφικού περιβάλλοντος (View) από τη λογική της εφαρμογής (Model), χρησιμοποιώντας ένα ενδιάμεσο επίπεδο (ViewModel) (Σχήμα 4.5). Έτσι, η διεπαφή δημιουργείται δηλώνοντας αντικείμενα σε μια γλώσσα βασισμένη στη γλώσσα XML, ενώ η λογική της εφαρμογής μπορεί να γραφεί σε οποιαδήποτε γλώσσα του .NET Framework. Για την υλοποίησή μας χρησιμοποιήσαμε τη γλώσσα C# για τον προγραμματισμό της λογικής της εφαρμογής. Η γλώσσα C# είναι μια γλώσσα προγραμματισμού που χρησιμοποιεί το .NET Framework και υποστηρίζει πολλούς διαφορετικούς τρόπους προγραμματισμού, όπως τον προστακτικό προγραμματισμό (imperative programming), τον αντικειμενοστραφή προγραμματισμό (object-oriented programming) και τον συναρτησιακό προγραμματισμό (functional programming). Χρησιμοποιώντας τις κλάσεις του .NET Framework, η επικοινωνία με τον

εξυπηρετητή SOAP είναι αυτοματοποιημένη, αποφεύγοντας έτσι την ανάγκη να χειριστούμε την επικοινωνία μέσω δικτύου.

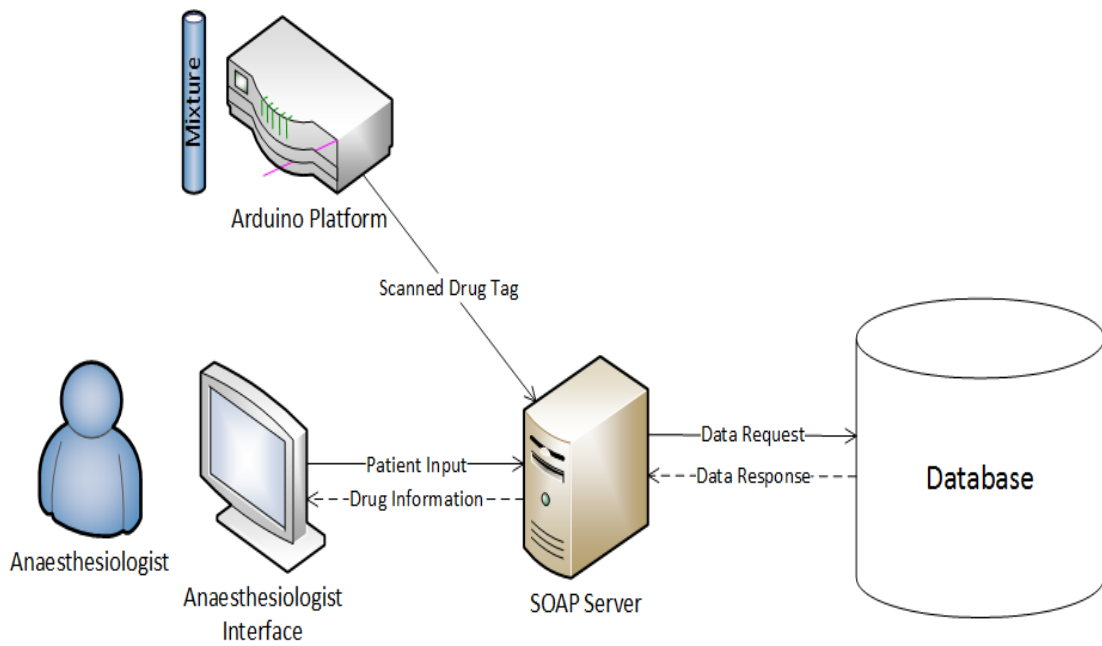


Σχήμα 4.5: Model View ViewModel

4.3 Αρχιτεκτονική του συστήματος

Το σύστημα που υλοποιήσαμε συνδυάζει όλες τις τεχνολογίες που παρουσιάστηκαν για να αυτοματοποιήσει τον έλεγχο των μειγμάτων αναισθησίας στο χειρουργείο. Στο Σχήμα 4.6 παρουσιάζουμε το σύστημα που δημιουργήθηκε. Αποτελείται από 4 μέρη:

- Μια διαδραστική διεπαφή χρήστη που χειρίζεται ο αναισθησιολόγος.
- Μια πλατφόρμα Arduino με έναν αναγνώστη RFID.
- Έναν εξυπηρετητή SOAP.
- Μία βάση δεδομένων.



Σχήμα 4.6: Αρχιτεκτονική συστήματος

Η διεπαφή χρήστη αποστέλλει πληροφορίες για την εισαγωγή ενός ασθενή στο χειρουργείο, δέχεται πληροφορίες για το μείγμα αναισθησίας που πρέπει να χρησιμοποιηθεί και τέλος ενημερώνει τον αναισθησιολόγο για το μείγμα που πρέπει να χρησιμοποιηθεί, καθώς και για το μείγμα που πέρασε από τον αναγνώστη RFID, όταν αυτό συμβεί. Η πλατφόρμα Arduino διαβάζει το RFID tag του μείγματος που θα χρησιμοποιηθεί και αποστέλλει το tag στον εξυπηρετητή SOAP. Τέλος ο εξυπηρετητής SOAP κάνει αιτήματα δεδομένων στη βάση δεδομένων, δέχεται πίσω τα στοιχεία ασθενών και φαρμάκων και ενημερώνει κατάλληλα την διεπαφή χρήστη.

4.4 Παράδειγμα χρήσης

Το σύστημα εφαρμόστηκε ενδεικτικά για 20 ασθενείς νοσοκομείου οι οποίοι υποβάλλονται σε ωτορινολαρυγγολογικές χειρουργικές επεμβάσεις την ίδια ημέρα. Τα χαρακτηριστικά των ασθενών (ηλικία, βάρος, μέση αρτηριακή πίεση και καρδιακές σφίξεις) καταγράφονται τη μέρα πριν την επέμβαση από τον αναισθησιολόγο σε βάση δεδομένων του νοσοκομείου. Μετά από την προαναισθητική εξέταση, ο ίδιος αναισθησιολόγος είναι υπεύθυνος για την επιλογή του κατάλληλου μείγματος αναισθησίας για την εισαγωγή στην αναισθησία. Η επιλογή του μείγματος για τις συγκεκριμένες επεμβάσεις γίνεται αναλόγως τα χαρακτηριστικά των ασθενών σύμφωνα με τον παρακάτω πίνακα.

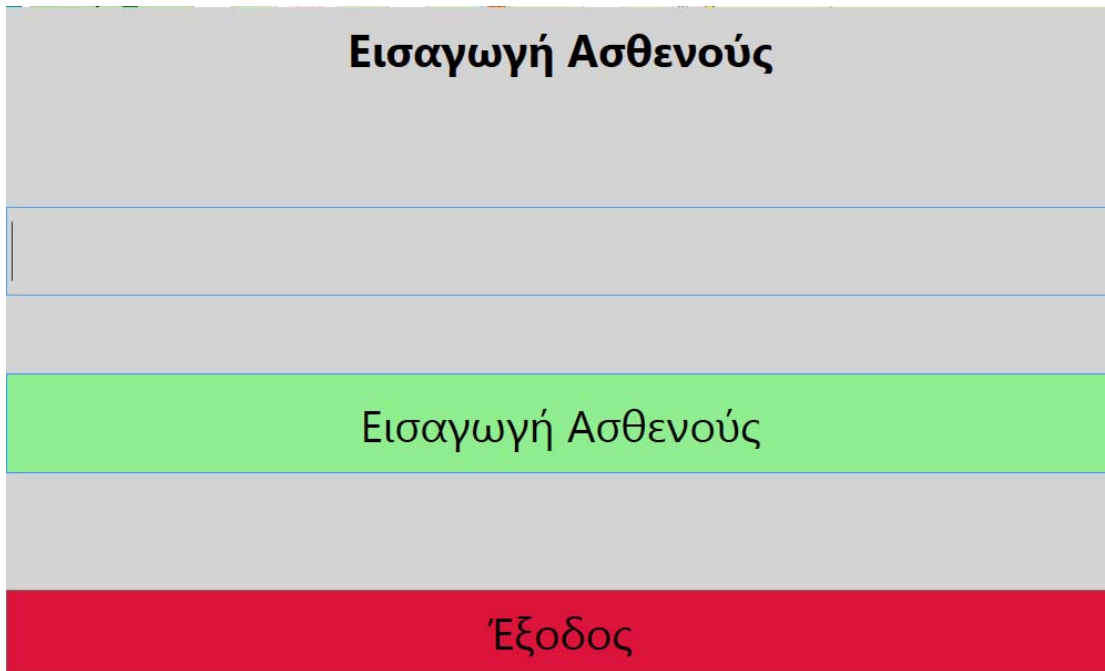
Πίνακας 4.2: Δοσολογία ενδοφλέβιων αναισθητικών φαρμάκων

ΟΜΑΔΑ Α		ΟΜΑΔΑ Β	
Μιδαζολάμη	2,5 mg	Μιδαζολάμη	2,5 mg
Προποφόλη	2-3 mg/kg	Ετομιδάτη	0,2-0,3 mg/kg
Ρεμιφεντανύλη	1 mcg/kg	Φεντανύλη	5-6 mcg/kg
Ροκουρόνιο	0,6 mg/kg	Ροκουρόνιο	0,6 mg/kg

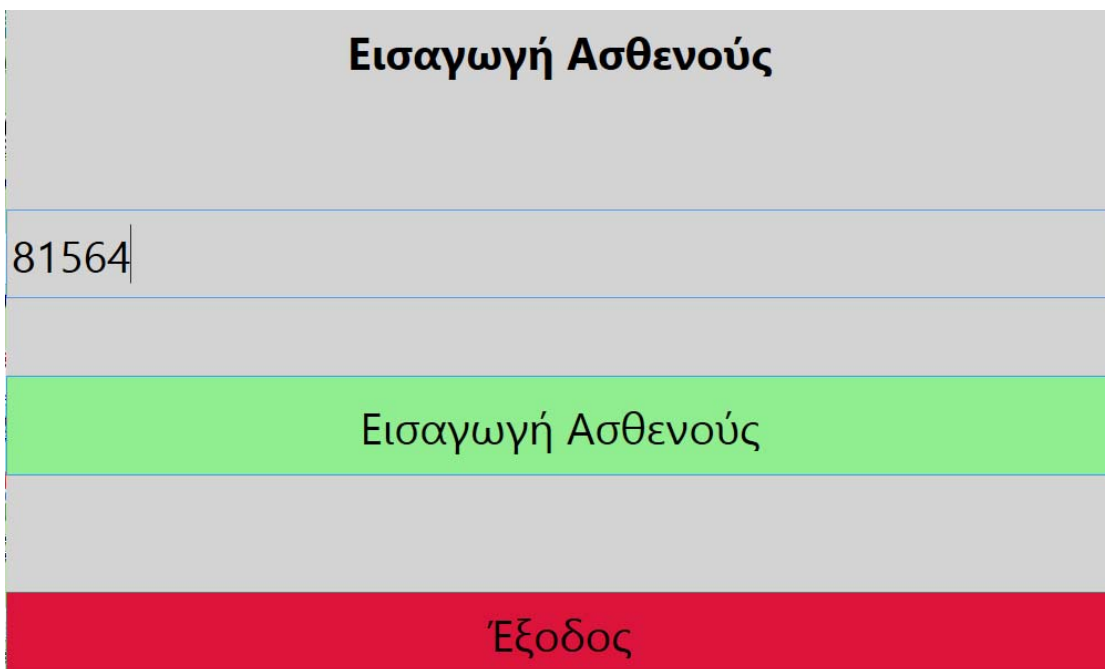
Ως ομάδα Β θεωρούνται οι ασθενείς άνω των 75 ετών και οι αιμοδυναμικά ασταθείς ασθενείς δηλαδή αυτοί που έχουν Μέση Αρτηριακή Πίεση (ΜΑΠ) > 80 mmHg και καρδιακές σφίξεις <50 ή >90 ανά λεπτό. Έτσι επιλέχθηκαν 20 μείγματα αναισθησίας, ένα για καθένα ασθενή. Το κάθε φιαλίδιο μείγματος φέρει RFID tag με ένα μοναδικό σειριακό αριθμό.

Σύμφωνα με τα παραπάνω, στο παράρτημα Α παρουσιάζεται ο πίνακας με τα στοιχεία των 20 ασθενών, οι αριθμοί μητρώου τους καθώς και ο κωδικός του σωστού μείγματος που αντιστοιχεί στον καθένα όπως αυτό επιλέχθηκε από τον αναισθησιολόγο. Επίσης παρουσιάζεται και ο πίνακας με τη δοσολογία του κάθε μείγματος καθώς και με την αντιστοίχιση κωδικού μείγματος και του σειριακού αριθμού του κάθε tag.

Ανήμερα της επέμβασης, ο ασθενής μεταφέρεται στην αίθουσα αναισθησίας όπου θα του γίνει η χορήγηση της ενδοφλέβιας αναισθησίας. Ο γιατρός εισάγει στην εφαρμογή τον αριθμό μητρώου του ασθενή. Στο σχήμα 4.7 βλέπουμε τι εμφανίζεται στην οθόνη. Υποθέτουμε ότι εισάγει τον αριθμό μητρώου 81564 που αντιστοιχεί στην ασθενή Ιωάννα Ηλιοπούλου (σχήμα 4.8) και επιλέγει «Εισαγωγή ασθενούς». Στη συνέχεια εμφανίζονται στην οθόνη τα στοιχεία της ασθενούς, ο κωδικός του σωστού μείγματος αναισθησίας που της αντιστοιχεί καθώς και τα συστατικά του μείγματος (Σχήμα 4.9) Η συγκεκριμένη ασθενής όπως βλέπουμε ανήκει στην ομάδα Β καθώς είναι αιμοδυναμικά ασταθής και είναι άνω των 75 ετών. Το φάρμακο που της αντιστοιχεί είναι αυτό με τον κωδικό 36.



Σχήμα 4.7: Αρχικό μήνυμα οθόνης



Σχήμα 4.8: Εισαγωγή αριθμού μητρώου ασθενούς

<u>Όνομα:</u>	Ιωάννα Ηλιοπούλου
<u>Ηλικία:</u>	88 ετών
<u>Βάρος:</u>	91 kg
<u>Μέση Αρτηριακή Πίεση:</u>	69 mmHg
<u>Καρδιακές Σφίξεις:</u>	107 /min
Φάρμακο	
Κωδικός:	<u>36</u>
Μείγμα:	Μιδαζολάμη 2,5 mg Ετομιδάτη 22,75 mg Φεντανύλη 500,5 mcg Ροκουρόνιο 54,6 mg
Επιστροφή	

Σχήμα 4.9: Προβολή στοιχείων ασθενούς και μείγματος φαρμάκου που του αντιστοιχεί

Για την ενημέρωση της βάσης και τη λήψη των δεδομένων του ασθενή και του μείγματος που πρέπει να χρησιμοποιηθεί, τα μηνύματα που ανταλλάσσει η διεπαφή με τον εξυπηρετητή είναι τα εξής:

```
POST http://pythonserver.dyndns.org:8008/
DEBUG:pysimplesoap.client:SOAPAction:
"http://pythonserver.dyndns.org:8008/RegisterPatient"
Content-length: 376
Content-type: text/xml; charset="UTF-8"
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header/>
<soap:Body>
  <RegisterPatient xmlns="http://pythonserver.dyndns.org">
    <patientID>81564</patientID></RegisterPatient>
</soap:Body>
</soap:Envelope>
```

```
date: Fri, 05 Sep 2014 20:03:27 GMT
content-type: text/xml
server: BaseHTTP/0.3 Python/2.7.3
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<RegisterPatientResponse xmlns="http://pythonserver.dyndns.org">
<PatientRegistered><MixtureDescription>Μιδαζολάμη 2,5 mg|Ετομιδάτη 22,75
mg|Φεντανύλη 500,5 mcg|Ροκουρόνιο 54,6 mg</MixtureDescription>
```

```

<Weight>91</Weight>
<MeanArterialPressure>69</MeanArterialPressure>
<PatientRegistrationSuccessful>>true</PatientRegistrationSuccessful>
<Name>Ιωάννα Ηλιοπούλου</Name>
<PatientExists>>true</PatientExists>
<MixtureCode>36</MixtureCode>
<Age>88</Age>
<HeartRate>107</HeartRate>
</PatientRegistered>
</RegisterPatientResponse></soap:Body></soap:Envelope>

```

Ο γιατρός παίρνει το φιαλίδιο που θεωρεί ότι είναι το σωστό και το περνάει μπροστά από τον RFID αναγνώστη. Ο αναγνώστης στέλνει το σειριακό αριθμό που έλαβε στον εξυπηρετητή, με το μήνυμα που φαίνεται παρακάτω:

```

POST http://pythonserver.dyndns.org:8008/
SOAPAction: "http://pythonserver.dyndns.org:8008/MixtureScanned"
Content-length: 387
Content-type: text/xml; charset="UTF-8"
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header/>
<soap:Body>
  <MixtureScanned xmlns="http://pythonserver.dyndns.org">
    <tagID>25352484848516651514868523</tagID></MixtureScanned>
</soap:Body>
</soap:Envelope>

```

Η διεπαφή χρήστη περιοδικά στέλνει ερωτήματα στον εξυπηρετητή, ώστε να λάβει τον κωδικό και τη περιγραφή του μίγματος που ενδεχομένως έχει περάσει από τον αναγνώστη ο αναισθησιολόγος. Τα μηνύματα που ανταλλάσσονται με τον εξυπηρετητή είναι τα παρακάτω:

```

POST http://pythonserver.dyndns.org:8008/
SOAPAction: "http://pythonserver.dyndns.org:8008/MixtureAdministered"
Content-length: 356
Content-type: text/xml; charset="UTF-8":<?xml version="1.0"
encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header/>
<soap:Body>
  <MixtureAdministered xmlns="http://pythonserver.dyndns.org">
    </MixtureAdministered>
</soap:Body>
</soap:Envelope>

```

```

date: Fri, 05 Sep 2014 20:10:22 GMT
content-type: text/xml
server: BaseHTTP/0.3 Python/2.7.3
<?xml version="1.0" encoding="UTF-8"?>

```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<MixtureAdministeredResponse xmlns="http://pythonserver.dyndns.org">
<MixtureAdministered><MixtureCode>36</MixtureCode>
<MixtureDescription>Μιδαζολάμη 2,5 mg|Ετομιδάτη 22,75 mg|Φεντανύλη 500,5
mcg|Ροκουρόνιο 54,6 mg</MixtureDescription>
<MixtureScanned>>true</MixtureScanned>
</MixtureAdministered></MixtureAdministeredResponse>
</soap:Body></soap:Envelope>

```

Αν το μίγμα που χρησιμοποιείται είναι το σωστό μείγμα, τότε εμφανίζεται στην οθόνη το εξής:

<p><u>Όνομα:</u> Ιωάννα Ηλιοπούλου</p> <p><u>Ηλικία:</u> 88 ετών</p> <p><u>Βάρος:</u> 91 kg</p> <p><u>Μέση Αρτηριακή Πίεση:</u> 69 mmHg</p> <p><u>Καρδιακές Σφίξεις:</u> 107 /min</p>	
<p>Φάρμακο</p> <p>Κωδικός: <u>36</u></p> <p>Μείγμα: Μιδαζολάμη 2,5 mg Ετομιδάτη 22,75 mg Φεντανύλη 500,5 mcg Ροκουρόνιο 54,6 mg</p>	<p>Σωστό Φάρμακο</p> <p><u>36</u></p> <p>Μιδαζολάμη 2,5 mg Ετομιδάτη 22,75 mg Φεντανύλη 500,5 mcg Ροκουρόνιο 54,6 mg</p>
<p>Επιστροφή</p>	

Σχήμα 4.10: Μήνυμα σε περίπτωση επιλογής σωστού μείγματος

Έτσι ξέρει ότι διάλεξε το σωστό μείγμα και ξεκινάει την ενδοφλέβια χορήγηση. Σε περίπτωση επιλογής λάθους φαρμάκου τότε εμφανίζεται το εξής:

Όνομα: Ιωάννα Ηλιοπούλου	
Ηλικία: 88 ετών	
Βάρος: 91 kg	
Μέση Αρτηριακή Πίεση: 69 mmHg	
Καρδιακές Σφίξεις: 107 /min	
Φάρμακο	Λανθασμένο Φάρμακο
Κωδικός: <u>36</u>	<u>46</u>
Μείγμα: Μιδαζολάμη 2,5 mg Ετομιδάτη 22,75 mg Φεντανύλη 500,5 mcg Ροκουρόνιο 54,6 mg	Μιδαζολάμη 2,5 mg Ετομιδάτη 25,5 mg Φεντανύλη 561 mcg Ροκουρόνιο 61,2 mg
Επιστροφή	

Σχήμα 4.11: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος

Τότε ο γιατρός επιλέγει διαφορετικό φιαλίδιο και το περνάει ξανά μπροστά από τον RFID αναγνώστη. Η διαδικασία επαναλαμβάνεται μέχρι να βγει το μήνυμα στην οθόνη ότι έχει επιλέξει το σωστό μείγμα. Αφού γίνει αυτό μπορεί να επιλέξει «Επιστροφή» για να γυρίσει στην αρχική οθόνη και να επιλέξει άλλο ασθενή ή για να κλείσει την εφαρμογή επιλέγοντας «Έξοδος». Ακολουθεί ένα άλλο παράδειγμα με ασθενή που ανήκει όμως τώρα στην ομάδα Α.

Βήμα 1^ο: Ο γιατρός εισάγει τον αριθμό μητρώου που αντιστοιχεί στον ασθενή Στυλιανό Γεωργίου.

Εισαγωγή Ασθενούς
81224
Εισαγωγή Ασθενούς
Έξοδος

Σχήμα 4.12: Εισαγωγή αριθμού μητρώου ασθενούς

Βήμα 2°: Εμφανίζονται τα στοιχεία του ασθενή και ο κωδικός του σωστού μείγματος.

<p><u>Όνομα:</u> Στυλιανός Γεωργίου</p> <p><u>Ηλικία:</u> 19 ετών</p> <p><u>Βάρος:</u> 75 kg</p> <p><u>Μέση Αρτηριακή Πίεση:</u> 92 mmHg</p> <p><u>Καρδιακές Σφίξεις:</u> 65 /min</p>	
<p>Φάρμακο</p> <p>Κωδικός: <u>31</u></p> <p>Μείγμα: Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg</p>	
<p>Επιστροφή</p>	

Σχήμα 4.13: Προβολή στοιχείων ασθενούς και μείγματος που του αντιστοιχεί

Βήμα 3°: Ο γιατρός επιλέγει το φιαλίδιο που θεωρεί ότι αντιστοιχεί στον ασθενή και το περνάει μπροστά από τον RFID αναγνώστη. Είναι όμως το λάθος μείγμα.

<p><u>Όνομα:</u> Στυλιανός Γεωργίου</p> <p><u>Ηλικία:</u> 19 ετών</p> <p><u>Βάρος:</u> 75 kg</p> <p><u>Μέση Αρτηριακή Πίεση:</u> 92 mmHg</p> <p><u>Καρδιακές Σφίξεις:</u> 65 /min</p>	
<p>Φάρμακο</p> <p>Κωδικός: <u>31</u></p> <p>Μείγμα: Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg</p>	<p>Λανθασμένο Φάρμακο</p> <p><u>54</u></p> <p>Μιδαζολάμη 2,5 mg Προποφόλη 212,5 mg Ρεμιφεντανύλη 85 mcg Ροκουρόνιο 51 mg</p>
<p>Επιστροφή</p>	

Σχήμα 4.14: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος

Βήμα 4°: Επιλέγει ξανά λάθος μείγμα.

Όνομα: Στυλιανός Γεωργίου	
Ηλικία: 19 ετών	
Βάρος: 75 kg	
Μέση Αρτηριακή Πίεση: 92 mmHg	
Καρδιακές Σφίξεις: 65 /min	
Φάρμακο	Λανθασμένο Φάρμακο
Κωδικός: <u>31</u>	<u>46</u>
Μείγμα: Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg	Μιδαζολάμη 2,5 mg Ετομιδάτη 25,5 mg Φεντανύλη 561 mcg Ροκουρόνιο 61,2 mg
Επιστροφή	

Σχήμα 4.15: Μήνυμα σε περίπτωση επιλογής λανθασμένου μείγματος

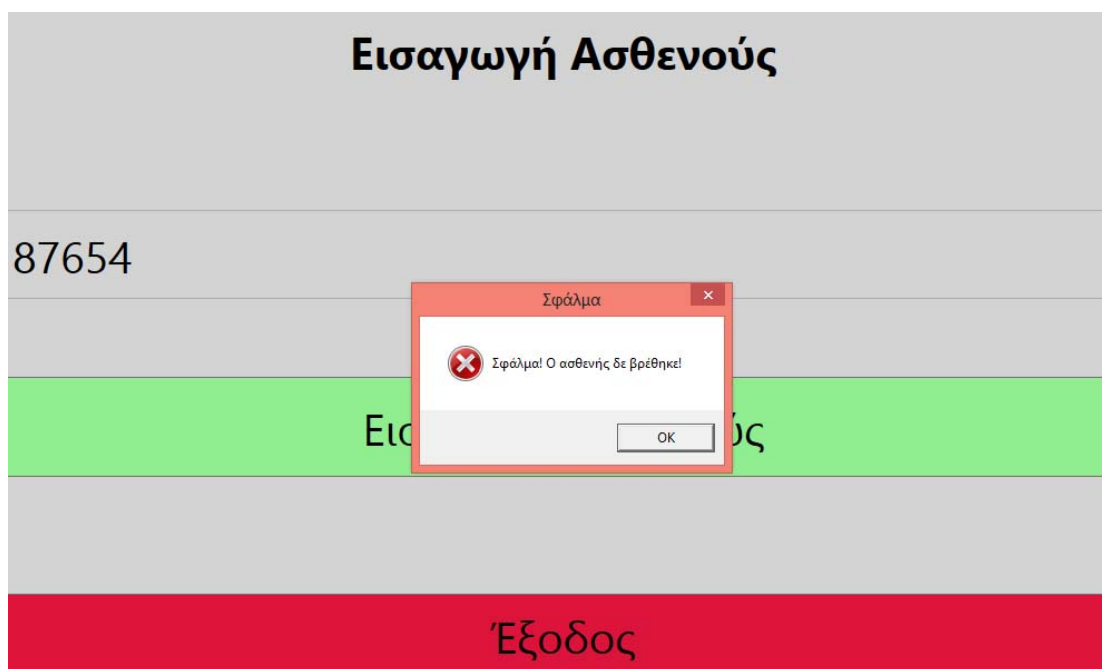
Βήμα 5°: Επιλέγει το σωστό μείγμα.

Όνομα: Στυλιανός Γεωργίου	
Ηλικία: 19 ετών	
Βάρος: 75 kg	
Μέση Αρτηριακή Πίεση: 92 mmHg	
Καρδιακές Σφίξεις: 65 /min	
Φάρμακο	Σωστό Φάρμακο
Κωδικός: <u>31</u>	<u>31</u>
Μείγμα: Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg	Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg
Επιστροφή	

Σχήμα 4.16: Μήνυμα σε περίπτωση επιλογής σωστού μείγματος

Βήμα 6°: Επιλέγει «Επιστροφή» και γυρίζει στην αρχική σελίδα. Έπειτα είτε εισάγει νέο Αριθμό Μητρώου και επαναλαμβάνει την ίδια διαδικασία για άλλο ασθενή άρα και για άλλη επέμβαση είτε επιλέγει «Έξοδος» και κλείνει η εφαρμογή.

Σε αυτό το σημείο, θα πρέπει να αναφέρουμε ότι σε περίπτωση που ο γιατρός εισάγει αριθμό μητρώου που δεν αντιστοιχεί σε κάποιο ασθενή στην οθόνη εμφανίζεται αντίστοιχο μήνυμα (Σχήμα 4.17).



Σχήμα 4.17: Μήνυμα σε περίπτωση εισαγωγής Α.Μ που δεν αντιστοιχεί σε κάποιον ασθενή

4.5 Συμπεράσματα

Σε αυτό το κεφάλαιο δείξαμε πως μπορούν να συνδυαστούν η τεχνολογία RFID μαζί με ένα υπολογιστικό σύστημα που χρησιμοποιεί την πλατφόρμα Arduino, το πρωτόκολλο SOAP και τη βάση δεδομένων του νοσοκομείου. Σκοπός του συστήματος είναι να ελαχιστοποιηθεί η πιθανότητα σφάλματος που υπάρχει στις τρέχουσες μεθόδους χορήγησης αναισθησίας εξαιτίας του εκτενούς ρόλου που παίζει ο ανθρώπινος παράγοντας.

Οι τεχνολογίες που χρησιμοποιούνται σήμερα για την πρόληψη του ανθρώπινου λάθους στην αναισθησία δυστυχώς είναι ελάχιστες και περιορίζονται στη χρήση barcodes, χρωματιστών ετικετών και προγεμισμένων συρίγγων. Λίγα συστήματα κάνουν χρήση υπολογιστικών συστημάτων και όταν γίνεται αυτό, περιορίζονται σε απλά καταγραφή των λαθών ή σε έλεγχο της ροής ενδοφλέβιων φαρμάκων. Εν αντιθέσει, στην υλοποίησή μας το υπολογιστικό σύστημα παίζει ένα κεντρικό ρόλο, τόσο στην αποθήκευση και μετάδοση πληροφοριών, όσο και στην επικοινωνία με τον αναισθησιολόγο και την ενημέρωσή του για τυχόντα λάθη.

Υλοποιήσαμε ένα υπόδειγμα ενός τέτοιου συστήματος που μπορεί να ελέγχει αν γίνεται σωστή επιλογή μείγματος ενδοφλέβιων αναισθητικών φαρμάκων κατά τη διάρκεια χειρουργικών επεμβάσεων και να αποτρέπει κάποιες κατηγορίες σφαλμάτων. Με αυτό το σύστημα, ο γιατρός επωμίζεται λιγότερες ευθύνες και δεν μπορεί να κάνει λάθος στη χορήγηση του μείγματος καθώς η διεπαφή όχι μόνο τον ειδοποιεί σε περίπτωση λάθους, αλλά και τον ενημερώνει για το σωστό μείγμα. Έτσι, μειώνεται η πιθανότητα λάθους και κατά συνέπεια και ο αριθμός των επιπλοκών.

Το εθνικό σύστημα υγείας επίσης θα μπορούσε να επωφεληθεί από το συγκεκριμένο σύστημα, καθώς τα ποσοστά επιπλοκών χορήγησης λάθους μείγματος μειώνονται. Αυτό θα είχε σαν αποτέλεσμα οι ασθενείς να αισθάνονται πιο ασφαλείς και να ενισχυθεί η αξιοπιστία των νοσοκομείων. Επιπλέον τα οικονομικά οφέλη θα είναι τεράστια μιας και τα νοσοκομεία θα αποφεύγουν την καταβολή υπέρογκων ποσών ως αποζημιώσεις σε περιπτώσεις λαθών των αναισθησιολόγων.

Το σύστημα που υλοποιήθηκε προλαμβάνει μια κατηγορία σφαλμάτων μόνο. Πιο συγκεκριμένα, δεν μπορεί να αποτρέψει σφάλματα στην επιλογή του μείγματος από τον αναισθησιολόγο. Μια πιθανή επέκταση του συστήματος θα μπορούσε να κωδικοποιήσει και τον τρόπο υπολογισμού του σωστού μείγματος, ειδοποιώντας τον αναισθησιολόγο κατά τον προεγχειρητικό έλεγχο για το ενδεικνύμενο μείγμα. Μια ακόμα επέκταση θα μπορούσε να αυτοματοποιήσει τη διαδικασία χορήγησης της αναισθησίας, χρησιμοποιώντας ένα πιο ολοκληρωμένο σύστημα που περιλαμβάνει μηχανικά μέρη.

ΠΑΡΑΡΤΗΜΑ Α: Πίνακες βάσης δεδομένων

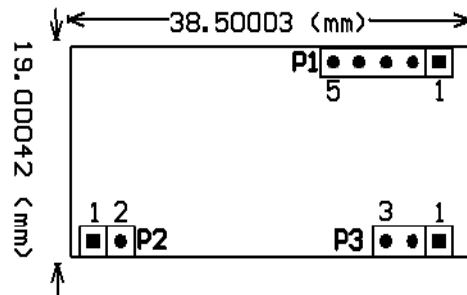
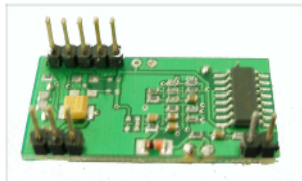
PatientID	Name	Age	Weight	MeanArterialPressure	HeartRate	MixtureCode
88254	Κυριάκος Παπαδόπουλος	25	85	90	60	54
87354	Ιωάννης Κυριακίδης	47	102	58	121	46
86564	Ελένη Πολυχρονίου	77	88	60	108	23
81224	Στυλιανός Γεωργίου	19	75	92	65	31
81564	Ιωάννα Ηλιοπούλου	88	91	69	107	36
84732	Μαρία Παπαντωνίου	35	55	85	55	62
82616	Σοφία Πέτρου	54	68	90	60	57
88163	Γεώργιος Μακρής	91	79	68	121	12
86343	Παναγιώτης Σωτηρίου	79	82	72	42	22
89638	Γεωργία Πικράκη	52	81	90	65	24
84583	Αντώνιος Αναστασίου	82	92	62	128	59
88634	Κρυσταλία Σταματάκη	27	57	91	72	78
88368	Αλεξία Μουστάκα	38	67	84	78	29
88965	Μάριος Γρίβας	78	85	57	110	69
83865	Δημήτριος Αλεξάνδρου	35	78	95	88	39
82836	Κωνσταντίνος Γκίκας	77	95	69	132	17
82863	Ιφιγένεια Κυπραίου	87	93	64	129	88
88693	Αθηνά Ιακωβάκη	45	72	86	60	87
88653	Γεώργιος Φωκάς	31	84	93	71	28
87865	Βασιλική Λέχου	83	81	61	125	44

MixtureCode	Description
54	Μιδαζολάμη 2,5 mg Προποφόλη 212,5 mg Ρεμιφεντανύλη 85 mcg Ροκουρόνιο 51 mg
46	Μιδαζολάμη 2,5 mg Ετομιδάτη 25,5 mg Φεντανύλη 561 mcg Ροκουρόνιο 61,2 mg
23	Μιδαζολάμη 2,5 mg Ετομιδάτη 22 mg Φεντανύλη 484 mcg Ροκουρόνιο 52,8 mg
31	Μιδαζολάμη 2,5 mg Προποφόλη 187,5 mg Ρεμιφεντανύλη 75 mcg Ροκουρόνιο 45 mg
36	Μιδαζολάμη 2,5 mg Ετομιδάτη 22,75 mg Φεντανύλη 500,5 mcg Ροκουρόνιο 54,6 mg
62	Μιδαζολάμη 2,5 mg Προποφόλη 137,5 mg Ρεμιφεντανύλη 55 mcg Ροκουρόνιο 33 mg
57	Μιδαζολάμη 2,5 mg Προποφόλη 170 mg Ρεμιφεντανύλη 68 mcg Ροκουρόνιο 40,8 mg
12	Μιδαζολάμη 2,5 mg Ετομιδάτη 19,75 mg Φεντανύλη 434,5 mcg Ροκουρόνιο 47,4 mg
22	Μιδαζολάμη 2,5 mg Ετομιδάτη 20,5 mg Φεντανύλη 451 mcg Ροκουρόνιο 49,2 mg
24	Μιδαζολάμη 2,5 mg Προποφόλη 202,5 mg Ρεμιφεντανύλη 81mcg Ροκουρόνιο 48,6 mg
59	Μιδαζολάμη 2,5 mg Ετομιδάτη 23 mg Φεντανύλη 506 mcg Ροκουρόνιο 55,2 mg
78	Μιδαζολάμη 2,5 mg Προποφόλη 142,5 mg Ρεμιφεντανύλη 57 mcg Ροκουρόνιο 34,2 mg
29	Μιδαζολάμη 2,5 mg Προποφόλη 167,5 mg Ρεμιφεντανύλη 67 mcg Ροκουρόνιο 40,2 mg
69	Μιδαζολάμη 2,5 mg Ετομιδάτη 21,25 mg Φεντανύλη 467,5 mcg Ροκουρόνιο 51 mg
39	Μιδαζολάμη 2,5 mg Προποφόλη 195 mg Ρεμιφεντανύλη 78 mcg Ροκουρόνιο 46,8 mg
17	Μιδαζολάμη 2,5 mg Ετομιδάτη 23,75 mg Φεντανύλη 522,5 mcg Ροκουρόνιο 57 mg
88	Μιδαζολάμη 2,5 mg Ετομιδάτη 23,25 mg Φεντανύλη 511,5 mcg Ροκουρόνιο 55,8 mg
87	Μιδαζολάμη 2,5 mg Προποφόλη 180 mg Ρεμιφεντανύλη 72 mcg Ροκουρόνιο 43,2 mg
28	Μιδαζολάμη 2,5 mg Προποφόλη 210 mg Ρεμιφεντανύλη 84 mcg Ροκουρόνιο 50,4 mg
44	Μιδαζολάμη 2,5 mg Ετομιδάτη 20,25 mg Φεντανύλη 445,5 mcg Ροκουρόνιο 48,6 mg

TagID	MixtureCode
25352484848516656554857703	54
25352484848515569655556693	46
25352484848515555657056703	23
25352484848515649496767653	31
25352484848516651514868523	36
25352484848516627575275275	62
25352484848516678678187171	57
25352484848516678781785178	12
25352484848516617815785887	22
25352484848516687851757181	24
25352484848516615751251512	59
25352484848516652151252151	78
25352484848516652152155421	29
25352484848516651515125252	69
25352484848516625251252152	39
25352484848516615211211951	17
25352484848516612151252521	88
25352484848516612521521521	87
25352484848516651252152152	28
25352484848516621512521541	44

ΠΑΡΑΡΤΗΜΑ Β: Datasheet RDM630

RDM630 Specification



1. Pin Definition (WEIGAND):

P1:

PIN1	DATA0
PIN2	DATA1
PIN3	
PIN4	GND
PIN5	+5V(DC)

P2:

PIN1	ANT1
PIN2	ANT2

P3:

PIN1	LED
PIN2	+5V(DC)
PIN3	GND

2. Pin definition (TTL interface RS232 data format):

P1:

PIN1	TX
PIN2	RX
PIN3	
PIN4	GND
PIN5	+5V(DC)

P2:

PIN1	ANT1
PIN2	ANT2

P3:

PIN1	LED
PIN2	+5V(DC)
PIN3	GND

ΠΑΡΑΡΤΗΜΑ Γ: Κώδικας Arduino

```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xD5, 0x66 };
char server[] = "pythonserver.dyndns.org"; //server address

IPAddress ip(192,168,1,6);

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to:
EthernetClient client;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    Ethernet.begin(mac, ip);
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);
}

void readTags()
{
  if (Serial.available() > 0)
  {
    String tagStr = "";
    // read tag numbers
    delay(100); // needed to allow time for the data to come in from the
serial buffer.
    int data1;
    for (int z = 0 ; z < 14 ; z++) // read the rest of the tag
    {
      data1 = Serial.read();
      tagStr += String(data1);
    }
    Serial.flush(); // stops multiple reads

    Serial.println(tagStr);

    Serial.println("connecting...");

    // if you get a connection, report back via serial:
    if(client.connect(server, 8008)) {
      Serial.println("Connected to server, preparing request");

      String body = "<?xml version=\"1.0\" encoding=\"UTF-
8\"?><soap:Envelope
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
```

```

instance\ "><soap:Header/><soap:Body><MixtureScanned
xmlns=\ "http://pythonserver.dyndns.org\ "><tagID>";
    String body2 =
"</tagID></MixtureScanned></soap:Body></soap:Envelope>";

    client.println("POST http://pythonserver.dyndns.org:8008/ HTTP/1.1");
    client.println("SOAPAction:
\"http://pythonserver.dyndns.org:8008/MixtureScanned\");
    int length = body.length() + tagStr.length() + body2.length();
    String secondString = String(length);
    String firstString = "Content-Length: ";

    String contentlength = firstString + secondString;

    client.println("Content-Type: text:xml; charset=\ "UTF-8\");
    client.print(firstString);
    client.println(secondString);
    client.println("Host: pythonserver.dyndns.org:8008");
    client.println("Connection: close");
    client.println();
    client.print(body);
    client.print(tagStr);
    client.println(body2);
    client.println();
    Serial.println("OK");
    client.stop();
    }else {
    Serial.println("Connection Faild");
    }

    delay(1000);
}

}

void loop()
{
    readTags()
}

```


ΠΑΡΑΡΤΗΜΑ Δ: Κώδικας SOAP Server

```
server.py

#!/usr/bin/python

# import libs that we use
import sqlite3 as lite
from pysimplesoap.server import SoapDispatcher, SOAPHandler
from BaseHTTPServer import HTTPServer

# one db connection throughout
dbName = 'hospitalDB.db'
con = lite.connect(dbName)

# we use these to store scanned mixture for future responses
mixtureScanned = 0
mixtureScannedCode = ""
mixtureScannedDescription = ""

# we use this to control whether there is an active patient
# and what functionality should be available to clients
patientNum = 0
patientName = ""
mixtureToAdministerCode = ""
mixtureToAdministerDescription = ""

# the function that gets invoked when a RFID tag has been scanned
def mixtureScanned(tagID):
    "Mark a mixture as Scanned"
    global patientNum
    global mixtureScanned
    global mixtureScannedCode
    global mixtureScannedDescription
    global con
    if (patientNum != 0): # if there is an active patient
        with con:
            cur = con.cursor()
            cur.execute("SELECT Mixtures.MixtureCode, Mixtures.Description
FROM Mixtures LEFT OUTER JOIN TagsMixtures ON Mixtures.MixtureCode =
TagsMixtures.MixtureCode WHERE TagsMixtures.TagID = :TAGID", {"TAGID":
tagID})
            rows = cur.fetchall()
            mixtureScannedCode = rows[0][0] # get the mixture code that
corresponds to the tag and keep it in a global variable
            mixtureScannedDescription = rows[0][1]
            mixtureScanned = 1
            return {'MixtureScannedResponse': True}
    else:
        return {'MixtureScannedResponse': False}

# this function gets periodically invoked from the doctor interface to
return whether a mixture has been scanned
def mixtureAdministered():
    "Return the scanned mixture"
    global mixtureScanned
    global patientNum
    global mixtureScannedCode
    global mixtureScannedDescription
```

```

    if (mixtureScanned == 1 and patientNum != 0): # if there is an active
patient and a mixture has been scanned, return the global variable
content
        return {'MixtureAdministered' : {'MixtureScanned' : True,
'MixtureCode' : mixtureScannedCode, 'MixtureDescription' :
mixtureScannedDescription}}
    else:
        return {'MixtureAdministered' : {'MixtureScanned' : False,
'MixtureCode' : "", 'MixtureDescription' : ""}}

# this funtion gets invoked when setting the active patient
def registerPatient(patientID):
    "Register a patient"
    global con
    global patientNum
    global patientName
    global mixtureToAdministerCode
    global mixtureToAdministerDescription
    if (patientNum == 0): # if there is no active patient already
        with con:
            cur = con.cursor()
            cur.execute("SELECT
Name, Age, Weight, MeanArterialPressure, HeartRate, MixtureCode FROM Patients
WHERE PatientID = :PATID", {"PATID": patientID})
            rows = cur.fetchall()
            # try to get the patient's data from the DB
            if (len(rows) > 0): # if the patient exists, store the info to
local variables
                patientNum = patientID
                patientName = rows[0][0]
                age = rows[0][1]
                weight = rows[0][2]
                pressure = rows[0][3]
                heartrate = rows[0][4]
                mixtureToAdministerCode = rows[0][5]
                with con: # try to get the corresponding mixture description
from the DB
                    cur = con.cursor()
                    cur.execute("SELECT Description FROM Mixtures WHERE
MixtureCode = :MIXCODE", {"MIXCODE": mixtureToAdministerCode})
                    rows = cur.fetchall()
                    mixtureToAdministerDescription = rows[0][0]
                    return {'PatientRegistered' : {'PatientRegistrationSuccessful':
True, 'PatientExists': True, 'Name': patientName, 'Age': age, 'Weight':
weight, 'MeanArterialPressure': pressure, 'HeartRate': heartrate,
'MixtureCode': mixtureToAdministerCode, 'MixtureDescription':
mixtureToAdministerDescription}}
                else:
                    return {'PatientRegistered' : {'PatientRegistrationSuccessful':
False, 'PatientExists': False, 'Name': "", 'Age': 0, 'Weight': 0,
'MeanArterialPressure': 0, 'HeartRate': 0, 'MixtureCode': "",
'MixtureDescription': ""}}
            else:
                return {'PatientRegistered' : {'PatientRegistrationSuccessful':
False, 'PatientExists': True, 'Name': "", 'Age': 0, 'Weight': 0,
'MeanArterialPressure': 0, 'HeartRate': 0, 'MixtureCode': "",
'MixtureDescription': ""}}

# this function gets invoked when clearing the active patient
def cancelPatient():

```

```

    "Calculate mixture and dosage"
    global patientNum
    global patientName
    global mixtureToAdministerCode
    global mixtureToAdministerDescription
    global mixtureScanned
    global mixtureScannedCode
    global mixtureScannedDescription
    if (patientNum != 0): # if there is an active patient, clear all the
global variables
        patientNum = 0
        patientName = ""
        mixtureScanned = 0
        mixtureToAdministerCode = ""
        mixtureToAdministerDescription = ""
        mixtureScannedCode = ""
        mixtureScannedDescription = ""
        return {'CancelPatient' : True }
    else:
        return {'CancelPatient' : False }

# define the dispatcher for the requests, with the host it uses
dispatcher = SoapDispatcher(
    'dispatcher',
    location = "http://pythonserver.dyndns.org:8008/",
    action = 'http://pythonserver.dyndns.org:8008/', # SOAPAction
    namespace = "http://pythonserver.dyndns.org", prefix="ns0",
    debug = True,
    ns = True)

# register the user functions with their arguments and return values
dispatcher.register_function('MixtureScanned', mixtureScanned,
    returns={'MixtureScannedResponse': bool},
    args={'tagID': str})

dispatcher.register_function('MixtureAdministered', mixtureAdministered,
    returns={'MixtureAdministered' : {'MixtureScanned' : bool,
'MixtureCode' : str, 'MixtureDescription' : str}},
    args={})

dispatcher.register_function('RegisterPatient', registerPatient,
    returns={'PatientRegistered' : {'PatientRegistrationSuccessful':
bool, 'PatientExists': bool, 'Name': str, 'Age': int, 'Weight': int,
'MeanArterialPressure': int, 'HeartRate': int, 'MixtureCode': str,
'MixtureDescription': str}},
    args={'patientID': int})

dispatcher.register_function('CancelPatient', cancelPatient,
    returns={'CancelPatient' : bool},
    args={})

print "Starting server..."
httpd = HTTPServer(("", 8008), SOAPHandler) # create the HTTP server
httpd.dispatcher = dispatcher # assign the dispatcher to the server
httpd.serve_forever() # serve forever

```


ΠΑΡΑΡΤΗΜΑ Ε: Κώδικας διεπαφής χρήστη

- **MainWindow.xaml**

```
<Window x:Class="DoctorInterface.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:extToolkit="http://schemas.xceed.com/wpf/xaml/toolkit"
    Title="Doctor Interface" Height="600" Width="1000"
    WindowStyle="None" ResizeMode="NoResize" Icon="patient.ico"
    WindowStartupLocation="CenterScreen" WindowState="Normal">
    <DockPanel>
        <Frame x:Name="_mainFrame" />
    </DockPanel>
</Window>
```

- **MainWindow.xaml.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace DoctorInterface
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        // the fields of the main window
        private PatientMixtureData _datacontext = null;
        private Pages.MainPage _mainpage = null;
        private Pages.MixturePage _mixturepage = null;

        public MainWindow()
        {
            // create all the pages, and register to the event in PatientMixtureData
            // this event fires when we admit a new active patient or cancel an
            active patient
            InitializeComponent();
            _mainFrame.NavigationUIVisibility = NavigationUIVisibility.Hidden;
            _datacontext = new PatientMixtureData();
            _datacontext.PatientRegisteredEv += datacontext_PatRegisteredEv;
            _mainpage = new Pages.MainPage(_datacontext);
            _mixturepage = new Pages.MixturePage(_datacontext);
        }
    }
}
```

```

        // navigate to the main page
        _mainFrame.Navigate(_mainpage);
    }

    void datacontext_PatRegisteredEv(object sender, bool registered)
    {
        // if there is a new active patient go to the mixture page
        // otherwise navigate to the main page

        if (registered)
        {
            _mainFrame.Navigate(_mixturepage);
        }
        else
        {
            _mainFrame.Navigate(_mainpage);
        }
    }
}
}
}

```

- **MainPage.xaml**

```

<Page x:Class="DoctorInterface.Pages.MainPage"
      x:Name="FirstPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:extToolkit="http://schemas.xceed.com/wpf/xaml/toolkit"
      DataContext="{Binding ElementName=FirstPage}"
      mc:Ignorable="d"
      d:DesignHeight="600" d:DesignWidth="1000"
      Title="Page1" Background="LightGray">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*"/>
            <RowDefinition Height="6*"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>
        <TextBlock Grid.Row="0" Text="Εισαγωγή Ασθενούς"
            HorizontalAlignment="Center" Margin="0,0,0,0" VerticalAlignment="Center"
            FontSize="40" FontWeight="Bold"/>
        <Grid Grid.Row="1" Height="Auto" HorizontalAlignment="Stretch"
            VerticalAlignment="Center">
            <Grid.RowDefinitions>
                <RowDefinition Height="150"/>
                <RowDefinition Height="90"/>
            </Grid.RowDefinitions>
            <extToolkit:WatermarkTextBox AcceptsReturn="False" Grid.Row="0"
                Background="LightGray" Text="{Binding Path=PatientID}" Watermark="Εισαγωγή Αριθμού
                Ασθενούς" HorizontalAlignment="Stretch" Height="80" Margin="0,0,0,0"
                TextAlignment="Left" VerticalAlignment="Top" VerticalContentAlignment="Center"
                FontSize="40"/>
            <Button Grid.Row="1" Content="Εισαγωγή Ασθενούς" Height="90"
                HorizontalAlignment="Stretch" Margin="0,0,0,0" VerticalAlignment="Top"
                Background="LightGreen" FontSize="40" Command="{Binding
                Path=RegisterPatientCommand}" IsDefault="True"/>
        </Grid>
    </Grid>

```

```

        <Button Grid.Row="2" Content="Εξοδος" Margin="0,0,0,0"
VerticalAlignment="Stretch" Background="Crimson" FontSize="40"
                Command="{Binding Path=QuitCommand}"/>
    </Grid>
</Page>

```

- **MainPage.xaml.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using DoctorInterface.PythonServiceReference;

namespace DoctorInterface.Pages
{
    /// <summary>
    /// Interaction logic for Page1.xaml
    /// </summary>
    public partial class MainPage : Page
    {
        // properties and fields of the class

        // The id that the user inputs
        private string _patientID;
        private PatientMixtureData _mixtureData = null;

        public string PatientID
        {
            get { return _patientID; }
            set { _patientID = value; }
        }

        // the commands bound to the keys
        public ICommand QuitCommand
        {
            get;
            internal set;
        }

        public ICommand RegisterPatientCommand
        {
            get;
            internal set;
        }

        // the method that shuts down the application
        public void QuitApplication()
        {

```

```

        if (System.Windows.MessageBox.Show("Είστε σίγουροι ότι θέλετε να
κλείσετε την εφαρμογή?",
        "Έξοδος", System.Windows.MessageBoxButton.YesNo) !=
MessageBoxResult.Yes)
    {
        return;
    }
    else
    {
        Application.Current.Shutdown(0);
    }
}

// the method that registers a patient
public void RegisterPatient()
{
    int patientID;
    if (Int32.TryParse(_patientID, out patientID))
    {
        // get a new client
        dispatcherPortTypeClient client = new dispatcherPortTypeClient();
        try
        {
            // do the rpc call
            var response = client.RegisterPatient(patientID);
            if (response.PatientRegistrationSuccessful)
            {
                // if the response was ok, hold the info of the patient and
the corresponding mixture in the _mixtureData object
                _mixtureData.MixtureCode = response.MixtureCode;
                _mixtureData.MixtureDescription =
response.MixtureDescription.Replace('|', '\n'); // we store any string in the DB
with lines denoted by the pipe character
                _mixtureData.PatientName = response.Name;

                _mixtureData.PatientAge = response.Age;

                _mixtureData.PatientWeight = response.Weight;
                _mixtureData.PatientMeanArterialPressure =
response.MeanArterialPressure;
                _mixtureData.PatientHeartRate = response.HeartRate;

                _mixtureData.PatientID = _patientID;
                _mixtureData.PatientRegistered = true;
            }
            else if (!response.PatientExists)
            {
                System.Windows.MessageBox.Show("Σφάλμα! Ο ασθενής δε
βρέθηκε!", "Σφάλμα", MessageBoxButton.OK, MessageBoxImage.Error);
            }
            else
            {
                System.Windows.MessageBox.Show("Σφάλμα! Παρακαλώ
επικοινωνήστε με το τεχνικό προσωπικό!", "Σφάλμα", MessageBoxButton.OK,
MessageBoxImage.Error);
            }
        }
        catch (Exception)
        {
            System.Windows.MessageBox.Show("Σφάλμα! Παρακαλώ επικοινωνήστε
με το τεχνικό προσωπικό!", "Σφάλμα", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }
}

```



```

    }
}

public MainPage(PatientMixtureData mixturedata)
{
    // bind the commands to their methods (the binding to the buttons is
done in the xaml file)
    QuitCommand = new RelayCommand(QuitApplication);
    RegisterPatientCommand = new RelayCommand(RegisterPatient);
    _mixtureData = mixturedata;
    InitializeComponent();
}
}
}

```

- **MixturePage.xaml**

```

<Page x:Class="DoctorInterface.Pages.MixturePage"
x:Name="MixturePag"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
mc:Ignorable="d"
DataContext="{Binding ElementName=MixturePag}"
d:DesignHeight="600" d:DesignWidth="1000"
Title="MixturePage" Loaded="MixturePag_Loaded" Background="LightGray" >
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="11.5*" />
        <RowDefinition Height="20*" />
        <RowDefinition Height="4*" />
    </Grid.RowDefinitions>
    <Border Grid.Row="0" BorderBrush="Black" BorderThickness="1" />
    <Grid Grid.Row="0" Margin="5,5,5,5">
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto" />
            <ColumnDefinition Width="Auto" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <TextBlock Grid.Row="0" Grid.Column="0" Text="Όνομα:" Margin="5,5,5,5"
HorizontalAlignment="Right" VerticalAlignment="Center" TextDecorations="Underline"
FontSize="20" />
        <TextBlock Grid.Row="1" Grid.Column="0" Text="Ηλικία:" Margin="5,5,5,5"
HorizontalAlignment="Right" VerticalAlignment="Center" TextDecorations="Underline"
FontSize="20" />
        <TextBlock Grid.Row="2" Grid.Column="0" Text="Βάρος:" Margin="5,5,5,5"
HorizontalAlignment="Right" VerticalAlignment="Center" TextDecorations="Underline"
FontSize="20" />
        <TextBlock Grid.Row="3" Grid.Column="0" Text="Μέση Αρτηριακή Πίεση:"
Margin="5,5,5,5" HorizontalAlignment="Right" VerticalAlignment="Center"
TextDecorations="Underline" FontSize="20" />

```

```

        <TextBlock Grid.Row="4" Grid.Column="0" Text="Καρδιακές Σφίξεις:"
Margin="5,5,5,5" HorizontalAlignment="Right" VerticalAlignment="Center"
TextDecorations="Underline" FontSize="20"/>

        <TextBlock Grid.Row="0" Grid.Column="1" Text="{Binding
Path=MixtureData.PatientName}" Margin="10,5,10,5" HorizontalAlignment="Right"
VerticalAlignment="Center" FontSize="20"/>
        <TextBlock Grid.Row="1" Grid.Column="1" Text="{Binding
Path=MixtureData.PatientAge, StringFormat={}{0} ετών}" Margin="10,5,10,5"
HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20"/>
        <TextBlock Grid.Row="2" Grid.Column="1" Text="{Binding
Path=MixtureData.PatientWeight, StringFormat={}{0} kg}" Margin="10,5,10,5"
HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20"/>
        <TextBlock Grid.Row="3" Grid.Column="1" Text="{Binding
Path=MixtureData.PatientMeanArterialPressure, StringFormat={}{0} mmHg}"
Margin="10,5,10,5" HorizontalAlignment="Center" VerticalAlignment="Center"
FontSize="20"/>
        <TextBlock Grid.Row="4" Grid.Column="1" Text="{Binding
Path=MixtureData.PatientHeartRate, StringFormat={}{0} /min}" Margin="10,5,10,5"
HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20"/>
    </Grid>
    <Grid Grid.Row="1">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
        <Border Grid.Column="0" BorderBrush="Black" BorderThickness="1" />
        <Border Grid.Column="1" BorderBrush="Black" BorderThickness="1" />

        <Grid Grid.Column="0">
            <Grid.RowDefinitions>
                <RowDefinition Height="40"/>
                <RowDefinition Height="*"/>
            </Grid.RowDefinitions>
            <Grid Grid.Row="0">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="120"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <TextBlock Name="CorrMixtureTxt" Grid.Column="1" Text="Φάρμακο"
Height="Auto" TextAlignment="Center" FontSize="30" />
            </Grid>
            <Grid Grid.Row="1" Margin="5,0,5,0" VerticalAlignment="Center"
Height="Auto">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="120"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="80"/>
                    <RowDefinition Height="*"/>
                </Grid.RowDefinitions>
                <TextBlock Grid.Column="0" Grid.Row="0" Text="Κωδικός:"
FontSize="30" Height="Auto" TextAlignment="Right" TextWrapping="WrapWithOverflow" />
                <TextBlock Grid.Column="0" Grid.Row="1" Text="Μείγμα:"
FontSize="30" Height="Auto" TextAlignment="Right"/>
                <TextBlock Grid.Column="1" Grid.Row="0" Text="{Binding
Path=MixtureData.MixtureCode}" FontSize="30" TextDecorations="Underline"
Height="Auto" TextAlignment="Center" />
                <TextBlock Grid.Column="1" Grid.Row="1" Text="{Binding
Path=MixtureData.MixtureDescription}" FontSize="30" Height="Auto"
TextAlignment="Center"/>
            </Grid>
        </Grid>
    </Grid>

```

```

        </Grid>
    </Grid>

    <Grid Grid.Column="1">
        <Grid.RowDefinitions>
            <RowDefinition Height="40"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>
        <TextBlock Name="WrongMixtureTxt" Grid.Row="0" Text="" Height="Auto"
TextAlignment="Center" FontSize="30"/>
        <Grid Grid.Row="1" Margin="5,0,5,0" VerticalAlignment="Center"
Height="Auto">
            <Grid.RowDefinitions>
                <RowDefinition Height="80"/>
                <RowDefinition Height="*/>
            </Grid.RowDefinitions>
            <TextBlock Grid.Row="0" Name="ScannedCodeTxt" Text="{Binding
Path=MixtureData.MixtureScannedCode}" FontSize="30" TextDecorations="Underline"
Height="Auto" TextAlignment="Center"/>
            <TextBlock Grid.Row="1" Name="ScannedDescriptionTxt"
Text="{Binding Path=MixtureData.MixtureScannedDescription}" FontSize="30"
Height="Auto" TextAlignment="Center" />
        </Grid>
    </Grid>
</Grid>

    <Button Grid.Row="2" Command="{Binding Path=CancelPatientCommand}"
Content="Επιστροφή" FontSize="40" Background="Crimson"/>
</Grid>
</Page>

```

- **MixturePage.xaml.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.ComponentModel;
using DoctorInterface.PythonServiceReference;

```

```

namespace DoctorInterface.Pages
{
    /// <summary>
    /// Interaction logic for MixturePage.xaml
    /// </summary>
    public partial class MixturePage : Page
    {

```

```

// properties and fields of the class

private PatientMixtureData _mixtureData = null;
private System.Timers.Timer mixtureCheck = null;
public ICommand CancelPatientCommand
{
    get;
    internal set;
}
public PatientMixtureData MixtureData
{
    get { return _mixtureData; }
    set { _mixtureData = value; }
}

// the method that cancels an active patient
public void CancelPatient()
{
    // get the client
    dispatcherPortTypeClient client = new dispatcherPortTypeClient();
    // do the rpc call
    var response = client.CancelPatient();
    if (response)
    {
        // clear all the fields with patient info, since we start all over
        mixtureCheck.Enabled = false;
        MixtureData.MixtureScannedDescription = null;
        MixtureData.MixtureScannedCode = null;
        WrongMixtureTxt.Text = "";

        MixtureData.PatientRegistered = false;
        MixtureData.MixtureCode = null;
        MixtureData.MixtureDescription = null;
        MixtureData.PatientName = null;
        MixtureData.PatientID = null;
    }
    else
    {
        System.Windows.MessageBox.Show("Σφάλμα! Παρακαλώ επικοινωνήστε με το  

τεχνικό προσωπικό!", "Σφάλμα", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

public MixturePage(PatientMixtureData mixturedata)
{
    // initialize the fields
    MixtureData = mixturedata;
    CancelPatientCommand = new RelayCommand(CancelPatient);
    // set up the timer that requests for any scanned mixture
    mixtureCheck = new System.Timers.Timer();
    mixtureCheck.Interval = 6000;
    mixtureCheck.Elapsed += mixtureCheck_Elapsed;
    InitializeComponent();
}

// when the timer ticks, do the rpc call
void mixtureCheck_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    // get a client
    dispatcherPortTypeClient client = new dispatcherPortTypeClient();
    try
    {

```

```

        // do the rpc call
        var response = client.MixtureAdministered();
        if (response.MixtureScanned)
        {
            //get the info from the response, if a mixture has been scanned
            MixtureData.MixtureScannedCode = response.MixtureCode;
            MixtureData.MixtureScannedDescription =
response.MixtureDescription;

            if (MixtureData == null || MixtureData.MixtureCode == null ||
MixtureData.MixtureScannedCode == null)
            {
                // if the mixture is null, do nothing. We use the dispatcher
because the event is handled (possibly)
                // by a different thread than the UI one and cannot change
the UI element properties
                this.Dispatcher.Invoke(new Action(() =>
                {
                    ScannedCodeTxt.Foreground = new
SolidColorBrush(Colors.Green);
                    ScannedDescriptionTxt.Foreground = new
SolidColorBrush(Colors.Green);
                    WrongMixtureTxt.Text = "";
                    WrongMixtureTxt.Foreground = new
SolidColorBrush(Colors.Red);
                }));
            }
            else if
(MixtureData.MixtureCode.Equals(MixtureData.MixtureScannedCode))
            {
                // if the mixture is the correct one, draw it in green
                this.Dispatcher.Invoke(new Action(() =>
                {
                    ScannedCodeTxt.Foreground = new
SolidColorBrush(Colors.Green);
                    ScannedDescriptionTxt.Foreground = new
SolidColorBrush(Colors.Green);
                    WrongMixtureTxt.Text = "Σωστό Φάρμακο";
                    WrongMixtureTxt.Foreground = new
SolidColorBrush(Colors.Green);
                }));
            }
            else
            {
                // the mixture is wrong, put it in red and alert the user
                this.Dispatcher.Invoke(new Action(() =>
                {
                    ScannedCodeTxt.Foreground = new
SolidColorBrush(Colors.Red);
                    ScannedDescriptionTxt.Foreground = new
SolidColorBrush(Colors.Red);
                    WrongMixtureTxt.Text = "Λανθασμένο Φάρμακο";
                    WrongMixtureTxt.Foreground = new
SolidColorBrush(Colors.Red);
                }));
            }
        }
    }
}
catch(Exception)

```

```

        {
        }
    }

    private void MixturePag_Loaded(object sender, RoutedEventArgs e)
    {
        mixtureCheck.Enabled = true;
    }
}
}

```

- **PatientMixtureData.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;

namespace DoctorInterface
{
    // the event that notifies the main window to navigate to the appropriate page
    public delegate void RegisterPatientEvent(object sender, bool Registered);

    public class PatientMixtureData : INotifyPropertyChanged
    {
        // we use properties and the RaisePropertyChanged method because
        // the UI need to know when something has changed, for performance reasons

        public event RegisterPatientEvent PatientRegisteredEv;

        private string _mixtureScannedDescription = null;
        public string MixtureScannedDescription
        {
            get { return _mixtureScannedDescription; }
            set
            {
                if (value != null)
                {
                    _mixtureScannedDescription = value.Replace('|', '\n');
                }
                else
                {
                    _mixtureScannedDescription = value;
                }
                RaisePropertyChanged("MixtureScannedDescription");
            }
        }

        private string _mixtureScannedCode = null;
        public string MixtureScannedCode
        {
            get { return _mixtureScannedCode; }
            set
            {
                _mixtureScannedCode = value;
            }
        }
    }
}

```

```

        RaisePropertyChanged("MixtureScannedCode");
    }
}

private string _mixtureDescription = null;
public string MixtureDescription
{
    get { return _mixtureDescription; }
    set
    {
        if (value != null)
        {
            _mixtureDescription = value.Replace('|', '\n');
        }
        else
        {
            _mixtureDescription = value;
        }
        RaisePropertyChanged("MixtureDescription");
    }
}

private string _mixtureCode = null;
public string MixtureCode
{
    get { return _mixtureCode; }
    set
    {
        _mixtureCode = value;
        RaisePropertyChanged("MixtureCode");
    }
}

private string _patientName = null;
public string PatientName
{
    get { return _patientName; }
    set
    {
        _patientName = value;
        RaisePropertyChanged("PatientName");
    }
}

private int _patientAge = -1;
public int PatientAge
{
    get { return _patientAge; }
    set
    {
        _patientAge = value;
        RaisePropertyChanged("PatientAge");
    }
}
}

```

```

private int _patientWeight = -1;
public int PatientWeight
{
    get { return _patientWeight; }
    set
    {
        _patientWeight = value;
        RaisePropertyChanged("PatientWeight");
    }
}

private int _patientMeanArterialPressure = -1;
public int PatientMeanArterialPressure
{
    get { return _patientMeanArterialPressure; }
    set
    {
        _patientMeanArterialPressure = value;
        RaisePropertyChanged("PatientMeanArterialPressure");
    }
}

private int _patientHeartRate = -1;
public int PatientHeartRate
{
    get { return _patientHeartRate; }
    set
    {
        _patientHeartRate = value;
        RaisePropertyChanged("PatientHeartRate");
    }
}

public string PatientID = null;

private bool _patientRegistered;
public bool PatientRegistered
{
    get { return _patientRegistered; }
    set {
        _patientRegistered = value;

        // depending on the value, fire the event with the correct arguments
        if (PatientRegisteredEv != null)
        {
            PatientRegisteredEv(this, value);
        }
    }
}

private void RaisePropertyChanged(string propName)
{
    if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs(propName));
}
public event PropertyChangedEventHandler PropertyChanged;
}
}

```


Βιβλιογραφία

- [1] Kritikos, P. G., and S. P. Papadaki. "The history of the poppy and of opium and their expansion in antiquity in the eastern Mediterranean area." *Bull Narc* 19.4 (1967): 5-10.
- [2] Packard, Francis R. "History of medicine in the United States." *The American Journal of the Medical Sciences* 184.4 (1932): 569.
- [3] <http://vml.med.uoc.gr/moodle/lessons/88-anaesthesiology/arxeia/kef3-anaesthesia-drugs.pdf>, Τμήμα Ιατρικής Σχολή Επιστημών Υγείας - Πανεπιστήμιο Κρήτης, Εικονικά Ιατρικά Εργαστήρια - Virtual Medical Lab.
- [4] ΙΦΕΤ <http://www.ifet.gr/>.
- [5] Αργυρώ Ι. Φασουλάκη: "Αναισθησιολογία" Ιατρικές Εκδόσεις Π.Χ. Πασχαλίδης.
- [6] Abeysekera, A., et al. "Drug error in anaesthetic practice: a review of 896 reports from the Australian Incident Monitoring Study database." *Anaesthesia* 60.3 (2005): 220-227.
- [7] <http://vml.med.uoc.gr/moodle/lessons/88-anaesthesiology/arxeia/kef15-anaesthesia-delirium.pdf>, Τμήμα Ιατρικής Σχολή Επιστημών Υγείας - Πανεπιστήμιο Κρήτης, Εικονικά Ιατρικά Εργαστήρια - Virtual Medical Lab.
- [8] Kawashima, Y., et al. "Anesthesia-related mortality and morbidity over a 5-year period in 2,363,038 patients in Japan." *Acta Anaesthesiologica Scandinavica* 47.7 (2003): 809-817.
- [9] Yamamoto, Mamoru, Seiji Ishikawa, and Koshi Makita. "Medication errors in anesthesia: an 8-year retrospective analysis at an urban university hospital." *Journal of anesthesia* 22.3 (2008): 248-252.
- [10] Hove, Lars Dahlgaard, et al. "Analysis of deaths related to anesthesia in the period 1996–2004 from closed claims registered by the Danish Patient Insurance Association." *Anesthesiology* 106.4 (2007): 675-680.
- [11] Lienhart, Andre, et al. "Survey of anesthesia-related mortality in France." *Anesthesiology* 105.6 (2006): 1087-1097.
- [12] Tiret, L., et al. "Complications associated with anaesthesia—prospective survey in France." *Canadian Anaesthetists' Society Journal* 33.3 (1986): 336-344.
- [13] Catchpole, K., M. D. D. Bell, and S. Johnson. "Safety in anaesthesia: a study of 12 606 reported incidents from the UK National Reporting and Learning System." *Anaesthesia* 63.4 (2008): 340-346.

- [14] Webster, C. S., et al. "**The frequency and nature of drug administration error during anaesthesia.**" *Anaesthesia and intensive care* 29.5 (2001): 494-500.
- [15] Orser, Beverley A., Robert JB Chen, and Doreen A. Yee. "**Medication errors in anesthetic practice: a survey of 687 practitioners.**" *Canadian journal of anaesthesia* 48.2 (2001): 139-146.
- [16] Cooper, Jeffrey B., Ronald S. Newbower, and Richard J. Kitz. "**An analysis of major errors and equipment failures in anesthesia management: considerations for prevention and detection.**" *Anesthesiology* 60.1 (1984): 34-42.
- [17] Cooper, Jeffrey B., et al. "**Preventable anesthesia mishaps: a study of human factors.**" *Anesthesiology* 49.6 (1978): 399-406.
- [18] Braz, Leandro Gobbo, et al. "**Mortality in anesthesia: a systematic review.**" *Clinics* 64.10 (2009): 999-1006.
- [19] Fasting, Sigurd, and Sven Erik Gisvold. "**Adverse drug errors in anesthesia, and the impact of coloured syringe labels.**" *Canadian journal of anaesthesia* 47.11 (2000): 1060-1067.
- [20] Mark Robert. "**The History of RFID Technology.**" *RFID Journal*, 2005.
- [21] Landt, Jeremy. "**The history of RFID.**" *Potentials, IEEE* 24.4 (2005): 8-11.
- [22] Walton, Charles A. "**Portable radio frequency emitting identifier.**" U.S. Patent No. 4,384,288. 17 May 1983.
- [23] Klaus Finkenzeller. **RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication 3rd Edition.** John Wiley & Sons, 2010.
- [24] Sabri Serkan Basat, **Design and Characterization of RFID modules in Multilayer Configurations**, Master Thesis, Georgia Institute of Technology, December, 2006.
- [25] Impinj, Inc, Resources. <http://www.impinj.com//resources/about-rfid/the-different-types-of-rfid-systems/>.
- [26] Κατσικά Ναυσικά, **Η εφαρμογή της τεχνολογίας RFID σε εταιρείες Third Party Logistics**, Διπλωματική Εργασία, Πειραιάς, 2009.
- [27] Θεοδώρου Σταυρούλα, **Εισαγωγή Στα Συστήματα Ραδιοσυχνικής Αναγνώρισης (RFID) και Σχεδίαση ενός συγκεκριμένου συστήματος RFID για Εργοστάσιο Εμφιαλώσεως Νερού**, Διπλωματική Εργασία, Αθήνα, 2005.
- [28] Παναγοπούλου Νίκη, **RFID: Τεχνολογία, Θεσμικό Πλαίσιο και Μελέτη Εφαρμογών**, Διπλωματική Εργασία, Πειραιάς, 2007.

[29] Margaret Wasserman, "**RFID Reader Management Requirements.**" SLLRP IETF Pre-WG, 62nd IETF Meeting in Minneapolis, USA, 2005.

[30] Παπάζογλου Δήμητρα: **Σχεδίαση αναλογικού τμήματος παθητικής ετικέτας (tag) για σύστημα αναγνώρισης μέσω ραδιοσυχνότητων (RFID)**, Διπλωματική εργασία, Θεσσαλονίκη, 2013.

[31] Kushner, David. "**The making of Arduino.**" IEEE Spectrum 26, 2011.

[32] <http://arduino.cc/>

[33] Durda, Frank. "**Serial and UART Tutorial.**" FreeBSD Documentation, 1996.

[34] Paret, Dominique, and Carl Fenger. "**The I2C bus: from theory to practice.**" John Wiley & Sons, Inc., 1997.

[35] Box, Don, et al. "**Simple object access protocol (SOAP) 1.1.**" (2000).

[36] Kuhlman, Dave. **A Python Book: Beginning Python, Advanced Python, and Python Exercises.** Dave Kuhlman, 2009.

[37] Wright, Gary R., and W. Richard Stevens. **TCP/IP Illustrated.** Vol. 2. Addison-Wesley Professional, 1995.