



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	ΨΗΦΙΑΚΟΣ ΟΔΗΓΟΣ ΦΑΡΜΑΚΩΝ DIGITAL MEDICAL GUIDE
Όνοματεπώνυμο Φοιτητή	ΚΥΡΙΑΚΟΣ ΜΠΟΥΡΜΠΟΥΧΑΚΗΣ
Πατρώνυμο	ΙΩΑΝΝΗΣ ΕΥΑΓΓΕΛΟΣ
Αριθμός Μητρώου	ΜΠΠΛ/ 12042
Επιβλέπων	ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Οκτώβριος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο

Όνομα Επώνυμο

Όνομα Επώνυμο

Βαθμίδα

Βαθμίδα

Βαθμίδα

Table of Contents

Περίληψη	5
Εισαγωγή-Abstract	6
Ανασκόπηση πεδίου	7
Differential Diagnosis Software.....	7
Medical Imaging Software.....	8
Translational Bioinformatics	8
Consumer Health Informatics	8
Public Health Informatics	8
Αντίστοιχες Εφαρμογές.....	8
Ανάλυση Απαιτήσεων και Λειτουργιών.....	12
Βασικές λειτουργίες	12
Περιπτώσεις Χρήσης	12
Διάγραμμα περιπτώσεων χρήσης.....	14
Παρουσίαση και χρήση εφαρμογής(User Manual)	15
Επιλογή 1.....	15
Επιλογή 2.....	18
Επιλογή 3.....	20
Αρχιτεκτονική Συστήματος	20
Υλοποίηση INTRO.....	21
Υλοποίηση EXTRA.....	23
Υλοποίηση MAIN.....	25
Υλοποίηση MAIN2.....	26
Εξαιρέσεις και ισχυρισμοί.....	27
Η Βάση Δεδομένων	27
ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ	29
ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ.....	29
Υλοποίηση με MySQL Workbench	30
ΠΕΡΙΟΡΙΣΜΟΙ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	30
ΕΙΣΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ	30
ΕΡΩΤΗΣΕΙΣ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	31
Συμπεράσματα και Μελλοντικές Επεκτάσεις	32
Συμπεράσματα.....	32
Μελλοντικές εξελίξεις στον κλάδο της Βιοπληροφορικής	34
Πρωτοβουλία DNA Barcode.....	35
To portal Barcode of Life.....	37

Human Genome Project.....	38
Η ιστορία του HGP	39
Η έρευνα του HGP στην Βιοπληροφορική	39
Τρέχοντα προγράμματα Βιοπληροφορικής.....	40
Βιβλιογραφία	41
Παράρτημα.....	41

Ήταν μακρύς ο δρόμος ως εδώ — δύσκολος δρόμος.

Τώρα είναι δικός σου αυτός ο δρόμος.

Τον κρατάς όπως κρατάς το χέρι του φίλου σου και μετράς το σφυγμό του.

Γιάννης Ρίτσος

Όπως όλα τα πράγματα που κατάφερα τα μικρά και τα ασήμαντα έτσι και αυτό είναι πάνω από όλα προϊόν πείσματος και αγώνα που εμπνέεται και θα εμπνέεται από τον αγώνα των λαών του κόσμου για μια καλύτερη και πιο ανθρώπινη ζωή. Και ιδιαίτερα από τον αγώνα του λαού του τόπου μου με την βαριά κληρονομιά του ΕΑΜ-ΕΛΑΣ, του ΔΣΕ, του Πολυτεχνείου, των εργατικών και φοιτητικών αγώνων της γενιάς μου και την καθημερινή πάλη χιλιάδων ανθρώπων ενάντια στην λεηλασία του καπιταλισμού και του φασισμού.

Αφιερώνεται σε όσους κρυφά και φανερά περπάτησαν και περπατούν μαζί μου ... και πάνω από όλους στην γυναίκα της ζωής μου Κατερίνα.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον λέκτορα κ. Ευθύμιο Αλέπη για τις συμβουλές και παρατηρήσεις του και τον γιατρό κ. Δημήτριο Γουρλή για τον χρόνο και τις συμβουλές που μου διέθεσε ώστε να ξεκινήσω. Δεν θα μπορούσα να παραλείψω να ευχαριστήσω τον βιολόγο Ιωάννη Μπουρμπουχάκη που 34 χρόνια τώρα είναι σύμβουλος σε πολλά περισσότερα από την επιστήμη του

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή όπως προκύπτει και από τίτλο της ασχολείται με την διαδικασία ψηφιοποίησης ενός οδηγού φαρμάκων. Περιλαμβάνει ένα γραφικό περιβάλλον το οποίο επιτελεί τον ρόλο της επικοινωνίας ανθρώπου και υπολογιστή και την βάση δεδομένων η οποία είναι δομημένη έτσι ώστε να παρέχει πληροφορίες στον χρήστη μέσω κατάλληλων ερωτημάτων. Οι λειτουργίες που επιτελεί το λογισμικό είναι 1^ο Η καταχώρηση δεδομένων 2^ο Η εμφάνιση δεδομένων βάσει του ονόματος της δραστικής ουσίας 3^ο Ο έλεγχος επιπλοκής των δραστικών ουσιών δυο φαρμάκων βάσει του ονόματος των φαρμάκων.

Χρήστης του λογισμικού κυρίως είναι ο επαγγελματίας υγείας αλλά ορισμένες ιδιότητες του μπορεί να τις χρησιμοποιήσει οποιοσδήποτε. Σε κάθε περίπτωση ωστόσο το λογισμικό δημιουργήθηκε με γνώμονα την φιλικότητα προς τον χρήστη δεν παύει όμως το αντικείμενο του να απαιτεί πιο ειδικευμένες γνώσεις. Τέλος λόγω του αντικειμένου θα πρέπει το λογισμικό να είναι συνεχώς αναπτυσσόμενο και να επιδέχεται βελτιώσεις πράγμα το οποίο μπορεί να επιτευχθεί και μέσω της MySQL η οποία έχει ευχέρεια προσθήκης ή βελτίωσης χαρακτηριστικών και μέσω αντίστοιχα της JAVA για την διεπαφή ανθρώπου – υπολογιστή.

Η δομή της διατριβής αποτελείται από την ανασκόπηση του πεδίου, την παρουσίαση της εφαρμογής (οδηγός χρήστη), την αρχιτεκτονική του συστήματος τόσο από πλευρά βάσης δεδομένων όσο από την πλευρά διεπαφής αλλά και την σύνδεση μεταξύ τους και τέλος τα συμπεράσματα που προκύπτουν και τις μελλοντικές επεκτάσεις που μπορεί να έχει.

Abstract

This thesis is about a software which digitalize a medicine guide. It has a G..U.I environment which accomplish the human-machine communication and a database to store data. The database is constructed in such way that can provide information to the user with the use of queries. The user can store data, edit data and check if two medicines are having complications. The software is for medicine professionals but some features can be used from every user. The G.U.I is written in JAVA and database in MySql

Εισαγωγή

Το λογισμικό που δημιουργήθηκε και αναλύεται στην παρούσα διατριβή αποτελείται από μια βάση δεδομένων και μια διεπαφή. Οι δυνατότητες του είναι 1) Να δέχεται καταχωρήσεις φαρμάκων που περιλαμβάνουν βασικές πληροφορίες για αυτό 2) Να αναζητά στην βάση δεδομένων την δραστική ουσία του φαρμάκου και να εμφανίζει τα βασικά χαρακτηριστικά της δραστικής ουσίας ώστε να χρησιμοποιηθούν από τον επαγγελματία υγείας ή ακόμα και από τον απλό χρήστη και 3) Να δέχεται δύο εμπορικές επωνυμίες φαρμάκων και να ελέγχει αν η λήψη των δύο φαρμάκων δημιουργεί επιπλοκές και ποιες είναι αυτές. Η εφαρμογή μπορεί να φανεί χρήσιμη σε επαγγελματίες υγείας αφού μπορούν να έχουν ένα οδηγό αναφοράς για τα φάρμακα και της δραστικές ουσίες αυτών των οποίο μπορούν να εμπλουτίζουν και να συμβουλευούνται όποτε έχουν ανάγκη αλλά και ο απλός χρήστης μπορεί να συμβουλευτεί τον οδηγό για να αντλήσει κάποιες πληροφορίες σχετικά με την φαρμακευτική του αγωγή.

Ανασκόπηση πεδίου

Η χρήση της πληροφορικής σε συνδυασμό με την βιοτεχνολογία χρησιμοποιείται για τη διαχείριση της συνέχια αυξανόμενης γνωσιακής βάσης, της εξόρυξης δεδομένων, της διασύνδεσης ανθρώπου υπολογιστή, της ανάλυσης εικόνας και φυσικής γλώσσας. Από τον συνδυασμό των δυο επιστημονικών τομέων προκύπτει η Βιοπληροφορική. Ένας ορισμός της Βιοπληροφορικής είναι το πεδίο που ασχολείται με την αποτελεσματική χρήση βιοιατρικών δεδομένων, πληροφοριών και μεθόδων επίλυσης προβλημάτων-λήψης αποφάσεων για την προαγωγή της ανθρώπινης υγείας. Ο κυρίαρχος τομέας της Βιοπληροφορικής με τον οποίο συχνά συγχέεται είναι η Ιατρική Πληροφορική.

Ο τομέας της Ιατρικής Πληροφορικής είναι ένας ιδιαίτερα παραγωγικός τομέας εφαρμογών και ιδεών γενικότερα. Η Ιατρική Πληροφορική σαν τομέας είναι πολυκλαδικός περιλαμβάνει τομείς από τον χώρο των μηχανικών λογισμικού, ιατρών, νοσηλευτών, φαρμακοποιών και βιολόγων. Αλλά δεν σταματά εκεί γενικότερα η βιοτεχνολογία σαν επιστημονικός κλάδος, του οποίου τομέας είναι και η ιατρική πληροφορική, ενσωματώνει και ηλεκτρολόγους, ηλεκτρονικούς και μηχανολόγους μηχανικούς.

Είναι προφανές ότι πέρα από την πληροφορική η ιατρική και η βιολογία συνεργάζονται και με άλλους επιστημονικούς τομείς.

Ενδεικτικά κάποια πεδία τα οποία η ιατρική και η βιολογία έχει συνδυαστεί είναι :

- Βιοστατιστική
Σε αυτό τον κλάδο οι εφαρμογές της στατιστικής συνδυάζονται με την ιατρική και την βιολογία ώστε να ελέγχεται η εφαρμοσιμότητα διάφορων υποθέσεων. Επίσης ελέγχεται ο κίνδυνος, σχετικός κίνδυνος, η ευαισθησία και ειδικότητα.
- Ιατρική απεικόνιση
Σε αυτό τον κλάδο οι εφαρμογές της επεξεργασίας και της ανάλυσης εικόνας συνδυάζονται με την Ιατρική και την Βιολογία. Τα τεχνολογικά πρότυπα αναλύονται και συνδυάζονται με την απεικονιστική Ιατρική ώστε να επιτευχθεί η ευκολότερη και ασφαλέστερη εξαγωγή αποτελεσμάτων.
- Ιατρική Φυσική
Σε αυτό τον κλάδο οι αρχές και οι εφαρμογές της φυσικής και της Ιατρικής συνδυάζονται. Χαρακτηριστικές εφαρμογές είναι η Πυρηνική Ιατρική, η ραδιοθεραπεία (ονομάζεται επίσης και ράδιο-ογκολογία). Άλλη μια εφαρμογή είναι η νευροφυσική.
- Ιατρική Χημεία
Σε αυτό τον κλάδο τομείς της χημείας και κυρίως η οργανική χημεία συνδυάζονται με την ιατρική και την βιολογία για να μπορέσουν να μοντελοποιήσουν φάρμακα, ασθένειες και διάφορες διαδικασίες.
- Ιατρική κοινωνιολογία
Σε αυτό τον κλάδο συνδυάζονται οι τομείς της Ιατρικής και της κοινωνιολογίας ώστε να διαπιστώνεται κάθε φορά μια ανθρωπολογική και κοινωνιολογική αιτιότητα θεραπειών, ανοσία ή ευαισθησία σε ασθένειες όπως και στα θέματα της βιοηθικής.

Ειδικά η Ιατρική Πληροφορική ασχολείται με ένα μεγάλο εύρος εφαρμογών οι σημαντικότερες από τις οποίες είναι:

Differential Diagnosis Software

Λογισμικό το οποίο κυρίως διαχειρίζεται πληροφορίες από την κλινική εικόνα ασθενών σε αποθήκες δεδομένων από όπου γίνεται προσπάθεια να αντληθούν δεδομένα με την χρήση Έμπειρων Συστημάτων και γενικότερα τεχνητής νοημοσύνης. Μια τέτοια εφαρμογή βρίσκεται στην ηλεκτρονική διεύθυνση *en.diagnosispro.com* και ονομάζεται Diagnosis Pro.

Medical Imaging Software

Λογισμικό το οποίο ενσωματώνεται σε απεικονιστικές συσκευές όπως MRI's και απεικονίζει τα δεδομένα τα οποία συλλέγονται από τις απεικονιστικές συσκευές έτσι ώστε να μπορούν να χρησιμοποιηθούν από τους επαγγελματίες υγείας.

Translational Bioinformatics

Ανάπτυξη μεθόδων και λογισμικού για την χαρτογράφηση του γονιδιώματος και εξαγωγή συμπερασμάτων και θεραπειών με την χρήση data mining.

Consumer Health Informatics

Ανάπτυξη λογισμικού ευρύτερης χρήσης με το οποίο δίνεται η δυνατότητα σε απλούς χρήστες αλλά και επαγγελματίες υγείας να διαχειρίζονται θέματα υγείας που αφορούν τους ίδιους ή ασθενείς τους. Με αυτό το λογισμικό ο χρήστης μπορεί να διαχειρίζεται ιατρικούς φάκελους να παίρνει μετρήσεις σχετικά με δείκτες υγείας, να ρυθμίζει την φαρμακευτική του αγωγή και να ενημερώνεται για εξελίξεις σχετικά με θέματα υγείας που το απασχολούν.

Public Health Informatics

Ανάπτυξη λογισμικού για θέματα δημόσιας υγείας που περιλαμβάνει εφαρμογές επιτήρησης δημόσιας υγείας, πρόληψης και προαγωγής θεμάτων υγείας. Ο τομέας αυτός είναι ιδιαίτερα ευρύς και ασχολείται με το περιβάλλον τον εργασιακό χώρο και την κατοικία ατόμων η ομάδων ατόμων.

Αντίστοιχες Εφαρμογές

Η παρούσα εργασία κινείται κυρίως στον τομέα του consumer health informatics, κατά την έρευνα του γράφοντα στο διαδίκτυο βρέθηκαν δυο αντίστοιχες εφαρμογές και αρκετές που ενσωματώνουν εν μέρει χαρακτηριστικά. Στο διαδίκτυο υπάρχουν κυβερνητικές ιστοσελίδες οι οποίες αναγράφουν τις δραστικές ουσίες και τα χαρακτηριστικά των φαρμάκων που έχουν εγκριθεί στην εκάστοτε χώρα συνήθως είναι στατικές ιστοσελίδες με απλή καταγραφή των προϊόντων, χαρακτηριστικότερες είναι οι εξής:

- Medicines.org.uk
- www.nps.org.au
- www.fda.gov

Οι δυο εμπορικές εφαρμογές που αναφέρθηκαν παραπάνω αποτελούν ολοκληρωμένες λύσεις διαχείρισης φαρμάκων και κάνουν πολλά περισσότερα από απλή καταγραφή

Πρέπει να γίνει σαφές ότι οι αντίστοιχες εφαρμογές βρίσκονται σε ανώτερα στάδια επαγγελματισμού και είναι πιο ολοκληρωμένες από μια ακαδημαϊκή προσέγγιση όπως η παρούσα η οποία σε καμία περίπτωση δεν προσπαθεί να υποκαταστήσει καμία επαγγελματική εφαρμογή.

Η πρώτη εφαρμογή που λειτουργεί στο διαδίκτυο και είναι σχετική με την παρούσα είναι το www.galinos.gr

The screenshot shows the Galinos website interface. At the top, there is a navigation bar with the Galinos logo and the tagline 'Είναι καλός ο σωματότυπός σας;'. Below this, there are search and user profile options. The main content area features a large image of a classical bust and a section titled '4 χρόνια' with a sub-header 'Συμπερινομάζουμε: τέσσερα χρόνια λειτουργίας και σας παραστήσαμε για την εμπειρία που μας δέχεται καθημερινά.' Below this, there is a table titled 'Καλώς ήλθατε στο Γαλινό' listing the top 6 most sold pharmaceuticals in the last 30 days. The table includes columns for 'Θεση', 'Όνομα', and 'Διανομέας'.

Θεση	Όνομα	Διανομέας
1	CIPROXIN F.C.TAB.500MG.TAB.BTx10	Pfizer Hellas
2	VIBRAMYCIN DISP.TAB.100MG.TAB.BTx8	Pfizer Hellas A.E.
3	DALACIN C CAPS.300MG.CAP.BTx16(BLIST 2x8)	Pfizer Hellas A.E.
4	KLARICID F.C.TAB.500MG.TAB.BTx21(BLIST3x7)	Abbott Hellas A.E.
5	AUGMENTIN F.C.TAB.(875+125)MG.TAB.BTx12	GlaxoSmithKline
6	AEROLIN AER.MD.INH.100MCG.DOSE.ΣΥΣΚΕΥΗ	GlaxoSmithKline

Σύμφωνα με την περιγραφή που παρέχει η ίδια η ιστοσελίδα : « Ο «Γαλινός» είναι μια διαδικτυακή υπηρεσία πληροφόρησης για τα φάρμακα ανθρώπινης χρήσης.

Οι ενδεικτικές υπηρεσίες σύμφωνα πάντα με την ιστοσελίδα του είναι :

- **Ανάλυση χαρακτηριστικών δραστικών ουσιών**

Με επίκεντρο τη δραστική ουσία, θα βρείτε αναλυτικά όλες τις απαραίτητες πληροφορίες σχετικά με την ορθή δοσολογία, ανάλογα με την παρούσα νόσο, το φύλο, την ηλικία, τα σωματομετρικά μεγέθη του ασθενούς και την οδό χορήγησης.

Στις σελίδες των δραστικών ουσιών θα βρείτε επίσης αναλυτικές πληροφορίες για τις αντενδείξεις χορήγησης, τις αλληλεπιδράσεις (*drug-to-drug*, *drug-to-disease interactions*), το μηχανισμό δράσης, την φαρμακοδυναμική, τη φαρμακοκινητική της δραστικής ουσίας, καθώς και για τις ιδιαιτερότητες χρήσης σε καταστάσεις όπως η κύηση και η γαλουχία.

Αναζητώντας μία οποιαδήποτε ασθένεια σας προτείνονται αυτόματα οι δραστικές ουσίες με σχετική ένδειξη.

- **Έλεγχος συγχορήγησης φαρμάκων**

Ο ασφαλής έλεγχος των αλληλεπιδράσεων μεταξύ συγχορηγούμενων φαρμάκων (*drug-to-drug interaction check*) σας δίνει τη δυνατότητα να ελέγχετε εάν μεταξύ των συγχορηγούμενων φαρμάκων που λαμβάνει ένας ασθενής υπάρχουν αντενδείξεις, περιορισμοί ή αλληλεπιδράσεις, ώστε με ασφάλεια να τροποποιήσετε, εφ' όσον απαιτείται, τη φαρμακευτική του αγωγή για την πρόληψη της εμφάνισης ανεπιθύμητων ενεργειών και τη βελτίωση του επιπέδου φροντίδας υγείας.

- **Περιλήψεις χαρακτηριστικών προϊόντων (SPC)**

Για τη πλειονότητα των σκευασμάτων που κυκλοφορούν στην Ελληνική αγορά, μπορείτε να διαβάσετε την εγκεκριμένη από τον ΕΟΦ "περίληψη χαρακτηριστικών προϊόντος" (SPC), δηλαδή το επίσημο έγγραφο που περιγράφει με ακρίβεια τον τρόπο χρήσης και δράσης ενός φαρμάκου, βάσει του οποίου έχει λάβει άδεια κυκλοφορίας.

- **Διοικητική / οικονομική πληροφόρηση**

Όλα τα κυκλοφορούντα στην ελληνική αγορά φαρμακευτικά σκευάσματα παρουσιάζονται αναλυτικά με όλες τις απαραίτητες διοικητικές και οικονομικές πληροφορίες τους: Ευρετήρια εμπορικών ονομάτων φαρμάκων και δραστικών ουσιών

- Αντιστοίχιση δραστικών ουσιών και εμπορικών ονομάτων φαρμάκων
- Δελτίο τιμών φαρμάκων (χονδρική, λιανική και νοσοκομειακή τιμή) ανά φαρμακοτεχνική μορφή και συσκευασία
- Τρόπος συνταγογράφησης, διάθεσης και ασφαλιστικής κάλυψης φαρμάκων
- Φωτογραφίες και barcode συσκευασιών φαρμάκων
- **Φαρμακευτικές εταιρείες ημεδαπής και διανεμόμενα φάρμακα αυτών**

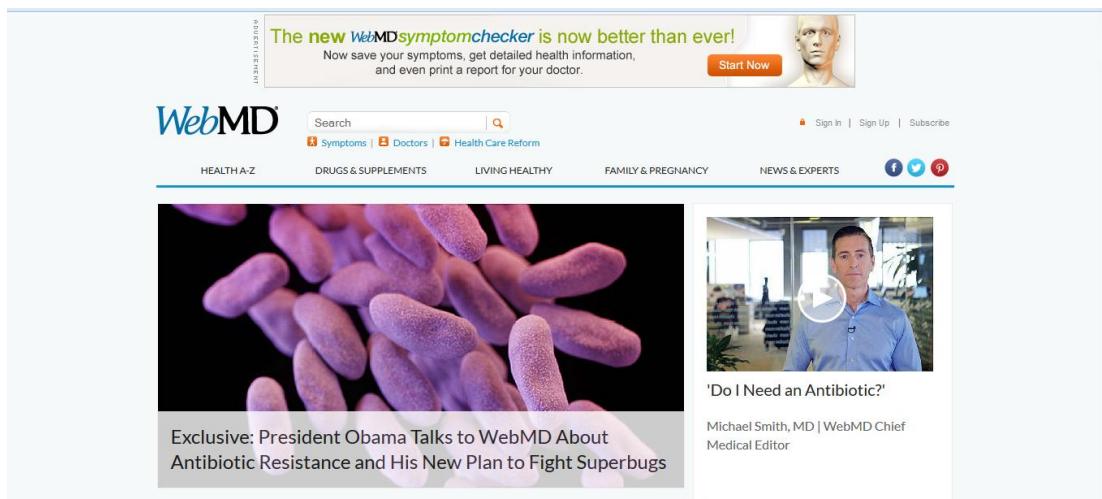
Πέρα από το εκάστοτε ισχύον δελτίο τιμών φαρμάκων, σας παρέχουμε τη δυνατότητα να βρείτε ποιο είναι το απόλυτα φθηνότερο εμπορικό σκεύασμα σε όρους κόστους ανά γραμμάριο δραστικής ουσίας.

- **Κωδικοποιήσεις**

Σας παρέχεται η δυνατότητα αναζήτησης σε διεθνείς κωδικοποιήσεις, όπως η *International Classification of Diseases (ICD10)*, η *Anatomical Therapeutic Chemical (ATC)* και εθνικές κωδικοποιήσεις, όπως η κωδικοποίηση ιατρικών πράξεων του Εθνικού Συστήματος Υγείας (πολύτιμο βοήθημα για την έκδοση ιατρικών και διοικητικών εγγράφων μονάδων υγείας) και η κωδικοποιημένη αναθεωρημένη έκδοση 2007 του Εθνικού Συνταγολογίου του Εθνικού Οργανισμού Φαρμάκων.»

Πηγή: www.galinos.gr

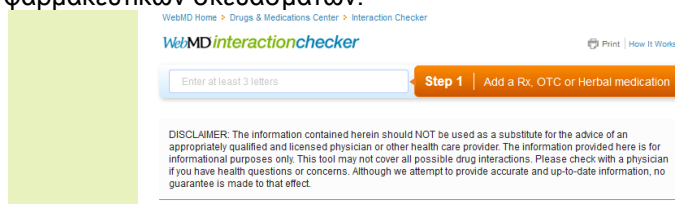
Η δεύτερη εφαρμογή που βρέθηκε είναι το www.webmd.com το οποίο αποτελεί μια πλατφόρμα με πληροφορίες σχετικά με θέματα υγείας.



Σύμφωνα πάντα με πληροφορίες ανηρημένες στην την ιστοσελίδα του το www.webmd.com είναι μια πλατφόρμα σχετικά με οτιδήποτε έχει να κάνει με θέματα υγείας. Είναι κομμάτι μια ευρύτερης εταιρείας που δραστηριοποιείται σε ιδιωτικές υπηρεσίες υγείας και πρόνοιας. Οι παροχές της ιστοσελίδας είναι αρκετά πλούσιες με κάποιες να αφορούν αποκλειστικά τους κατοίκους Η.Π.Α(πχ υπηρεσίες ιατρικών επισκέψεων) αλλά και αρκετές που αφορούν τον κάθε χρήστη εφόσον έχει ικανότητα κατανόησης της Αγγλικής γλώσσας και σχετίζονται με τον χώρο της ιατρικής πληροφορικής που μας αφορά. Αυτές είναι οι εξής:

- Health A-Z: Γλωσσάρι παθήσεων στο οποίο περιγράφονται αναλυτικά οι παθήσεις, άρθρα σχετικά με την κάθε πάθηση, τρόποι έλεγχου και αντιμετώπισης.
- Drugs A-Z: Γλωσσάρι φαρμακευτικών σκευασμάτων στο οποίο δίδεται η ικανότητα να γίνει αναζήτηση βάσει ονόματος φαρμάκου ή βάσει ασθένειας. Για κάθε φαρμακευτικό σκεύασμα παρέχονται πληροφορίες για:
 1. Την χρήση του φαρμάκου
 2. Της παρενέργειες
 3. Της προφυλάξεις του ασθενή
 4. Τις αλληλεπιδράσεις με άλλα φάρμακα
 5. Τις ενέργειες που πρέπει να γίνουν σε περίπτωση υπερδοσολογίας
 6. Τυχόν φωτογραφίες που υπάρχουν
 7. Κριτικές από ασθενείς

Επίσης υπάρχει εργαλείο με το οποίο ελέγχονται πιθανές αλληλεπιδράσεις μεταξύ φαρμακευτικών σκευασμάτων.



Η χρήση του εργαλείου είναι απλή, ο χρήστης καταχωρεί το όνομα του σκευάσματος που επιθυμεί στην συνέχεια το όνομα του δεύτερου σκευάσματος



Και στην συνέχεια εμφανίζει πιθανές επιπλοκές τις οποίες ταξινομεί με βαθμό επικινδυνότητας.

Τέλος στην ίδια ενότητα παρέχεται ενημέρωση σχετικά με τα νέα από τον χώρο των φαρμακευτικών εταιρειών.

- Η ιστοσελίδα επίσης παρέχει συμβουλές για διατροφή και άσκηση καθώς και για οικογενειακά θέματα και θέματα εγκυμοσύνης.

Ολοκληρώνοντας την ανασκόπηση πεδίου παρατηρείται μια άνθηση του τομέα της ιατρικής πληροφορικής που ασχολείται με τον τελικό καταναλωτή διότι είναι ένα αρκετά επικερδές πεδίο. Πρέπει ωστόσο να τονιστεί πως όλα τα συνδυαστικά αποτελέσματα των ιστοσελίδων – εφαρμογών λειτουργούν επικουρικά στην ιατρική γνώμатеυση και σε καμία περίπτωση δεν μπορούν να την αντικαταστήσουν. Επίσης οι εμπορικές ιστοσελίδες έχουν συνάψει συμφωνίες με φαρμακευτικές εταιρείες και προωθούν συχνά προϊόντα τους γεγονός που δεν πρέπει να επηρεάσει τον καταναλωτή από την στιγμή που δεν του συστήθηκε το σκεύασμα από τον ιατρό του.

Ανάλυση Απαιτήσεων και Λειτουργιών

Βασικές Λειτουργίες

Οι βασικές λειτουργίες του πληροφοριακού συστήματος που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας αναλύονται παρακάτω:

- 1.1 Καταγραφή στοιχείων φαρμάκων
- 1.2 Αναζήτηση χαρακτηριστικών των φαρμακευτικών σκευασμάτων βάσει ονόματος δραστικής ουσίας
- 1.3 Έλεγχος αλληλεπίδρασης μεταξύ δραστικών ουσιών βάσει εμπορικής ονομασία φαρμακευτικών σκευασμάτων
- 1.4 Επεξεργασία βάσης δεδομένων

Περιπτώσεις Χρήσης

Κατά την υλοποίηση του συστήματος προέκυψε ένας πίνακας ο οποίος προσδιορίζει τους χρήστες (Actors) του συστήματος και τις λειτουργίες που πραγματοποιούν στην εφαρμογή.

Δράστες(Actors)	Περιπτώσεις Χρήσης
Καταναλωτής-Πελάτης	Αναζήτηση & έλεγχος αλληλεπίδρασης δραστικής ουσίας
Επαγγελματίας Υγείας	Καταγραφή όλων των στοιχείων των φαρμάκων και καταχώρηση στη ΒΔ, Αναζήτηση βάσει δραστικής ουσίας & έλεγχος αλληλεπίδρασης
Διαχειριστής ΒΔ	Επεξεργασία, διαγραφή και συντήρηση δεδομένων στην ΒΔ

Στην συνέχεια θα αναλυθούν οι περιπτώσεις χρήσης υψηλού επιπέδου

Περίπτωση χρήσης: Αναζήτηση χαρακτηριστικών των δραστικών ουσιών.

Δράστες (Actors): Καταναλωτής, επαγγελματίες υγείας

Σκοπός: Η αναζήτηση των χαρακτηριστικών των φαρμακευτικών σκευασμάτων βάσει της ονομασίας της δραστικής ουσίας

Τύπος: Πρωταρχικός και απαραίτητος

Βασικές Λειτουργίες: 1.2

Περίπτωση χρήσης: Καταγραφή των χαρακτηριστικών των φαρμακευτικών σκευασμάτων

Δράστες (Actors): Επαγγελματίες υγείας

Σκοπός: Η καταγραφή όλων των δραστικών ουσιών και των χαρακτηριστικών τους που περιέχονται μέσα σε ένα φύλλο οδηγιών ώστε να μπορεί στην συνέχεια να ελεγχθεί.

Τύπος: Πρωταρχικός και απαραίτητος

Βασικές Λειτουργίες: 1.1

Περίπτωση χρήσης: Αναζήτηση αλληλεπίδρασης μεταξύ των δραστικών ουσιών.

Δράστες (Actors): Καταναλωτής, επαγγελματίες υγείας

Σκοπός: Η αναζήτηση επικίνδυνων για την υγεία του ασθενούς αλληλεπιδράσεων μεταξύ των χορηγούμενων φαρμακευτικών σκευασμάτων και εμφάνιση τις πιθανής επιπλοκής που θα προκύψει.

Τύπος: Πρωταρχικός και απαραίτητος

Βασικές Λειτουργίες: 1.3

Περίπτωση χρήσης: Επεξεργασία και συντήρηση της βάσης δεδομένων.

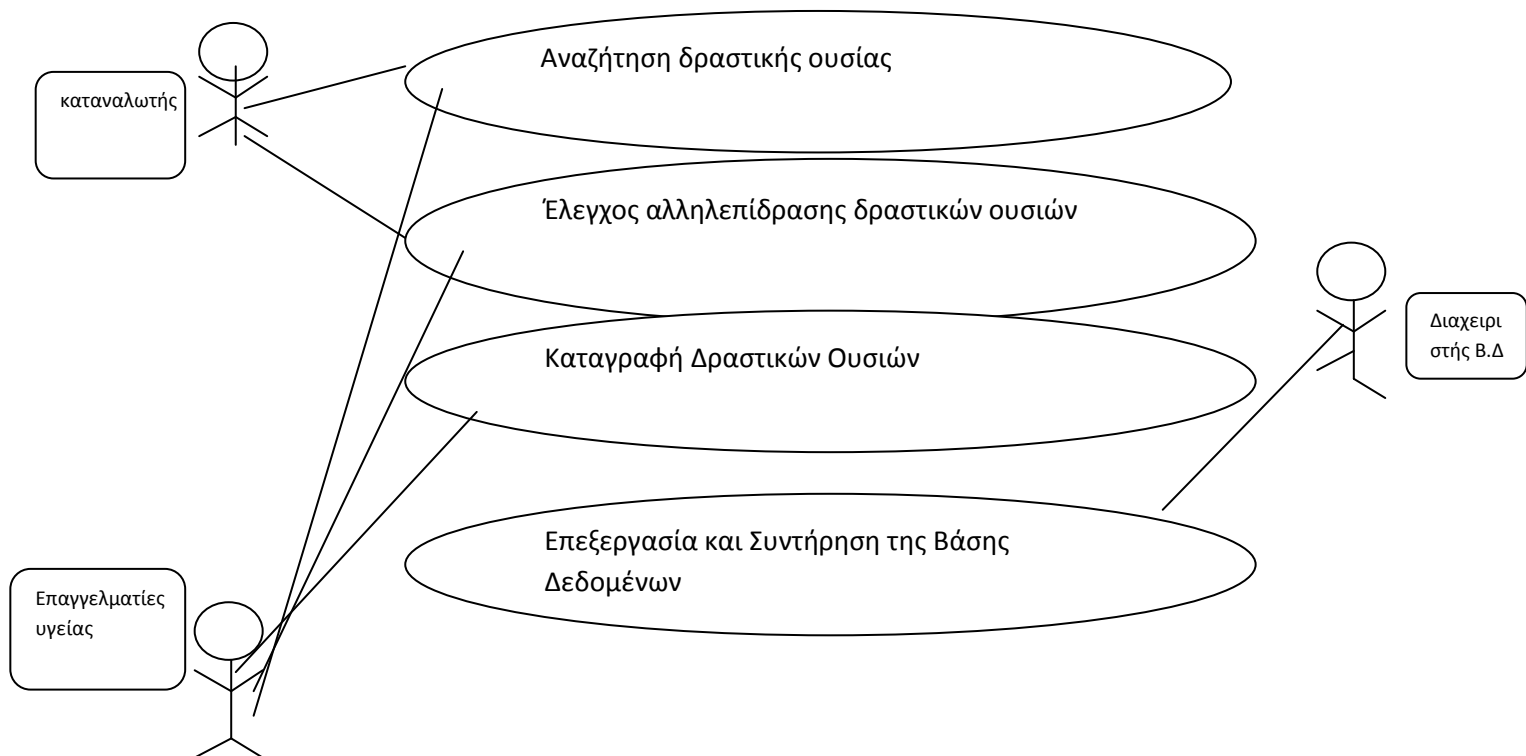
Δράστες (Actors): Διαχειριστής συστήματος

Σκοπός: Η περιοδική συντήρηση των δεδομένων που έχουν εισαχθεί στην βάση δεδομένων

Τύπος: Δευτερεύων και απαραίτητος

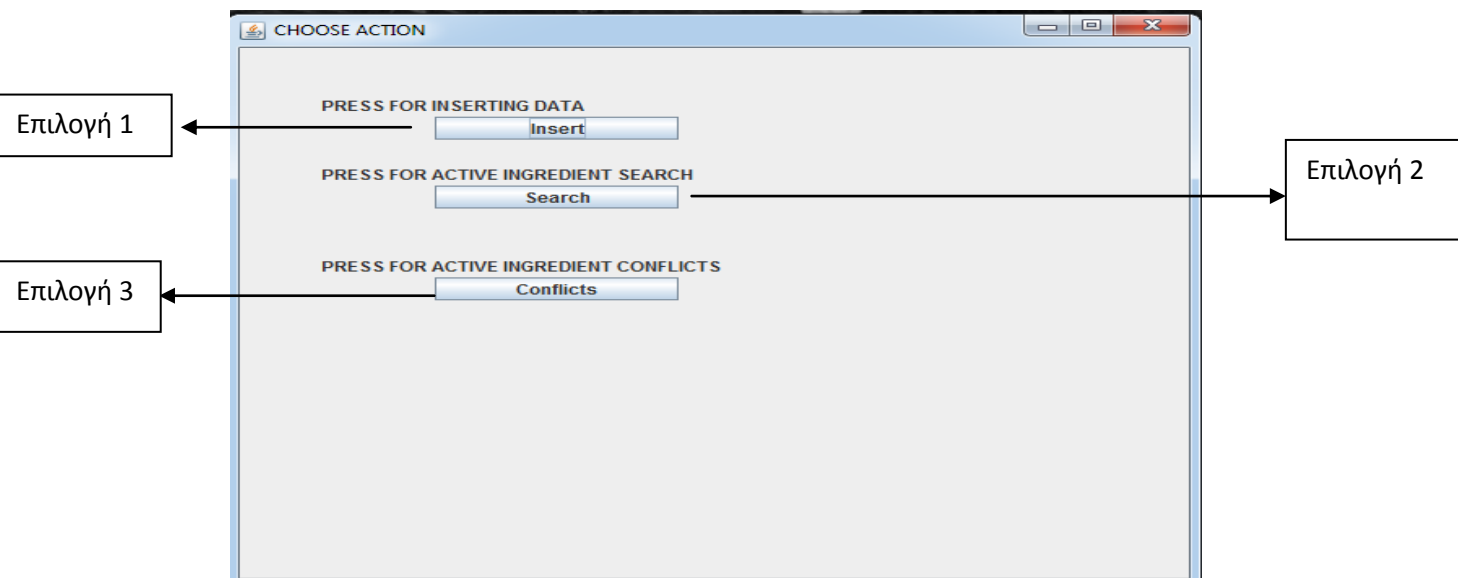
Βασικές Λειτουργίες: 1.4

Διάγραμμα περιπτώσεων χρήσης



Παρουσίαση και χρήση εφαρμογής (User Manual)

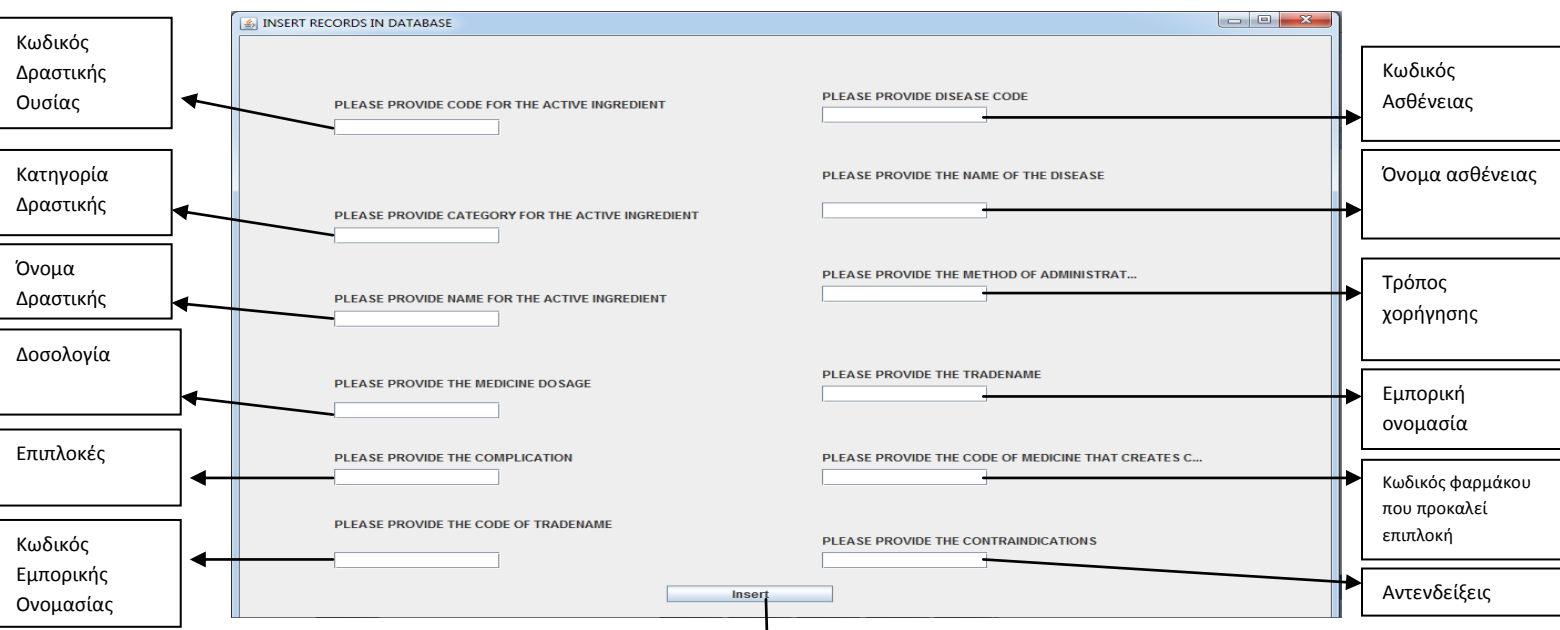
Κατά την εκκίνηση του το πρόγραμμα εμφανίζει την παρακάτω οθόνη η οποία δίνει στον χρήστη τρεις επιλογές



Επιλογή 1

Ο χρήστης πατώντας το κουμπί insert έχει την δυνατότητα να καταχωρήσει τα στοιχεία ενός καινούργιου φαρμάκου. Εδώ πρέπει να τονιστεί πως η εισαγωγή στοιχείων στη βάση δεδομένων πρέπει να γίνεται από επαγγελματίες υγείας αφού τα στοιχεία που καταχωρούνται απαιτούν εξειδικευμένες γνώσεις.

Εδώ πρέπει να τονιστεί πως στον παρακάτω πίνακα γίνεται μια προσπάθεια να κωδικοποιηθεί το φύλλο οδηγιών για τον χρήστη που περιέχεται σε ένα φαρμακευτικό σκεύασμα έτσι γίνεται προσπάθεια να ακολουθηθεί η ίδια σειρά καταγραφής.



Από την στιγμή που ο χρήστης είναι σίγουρος πως καταχώρησε σωστά όλα τα στοιχεία μπορεί να κάνει εισαγωγή στην βάση

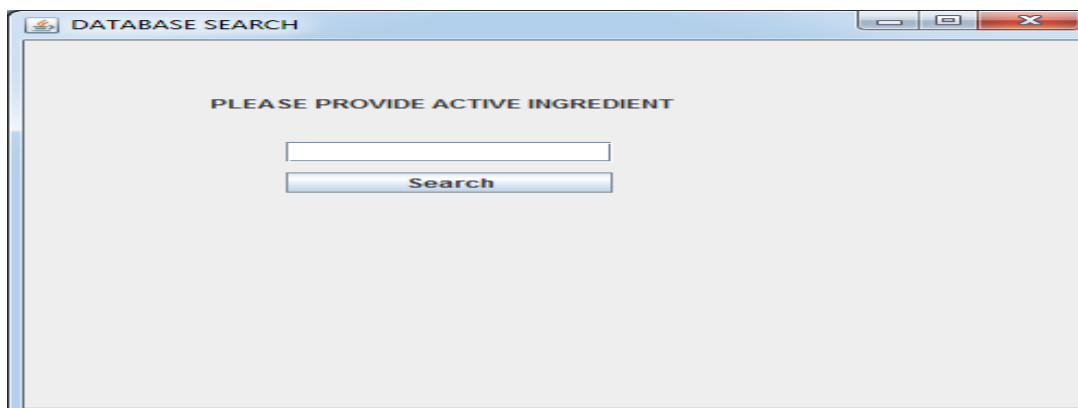
- Στο πεδίο «Κωδικός Ασθένειας» ο χρήστης επιλέγει τον μοναδικό κωδικό που χαρακτηρίζει την ασθένεια. Σε αυτό το πεδίο χρησιμοποιήθηκε το ICD (International Classification of Diseases) το οποίο είναι ένα εργαλείο κατάταξης για την επιδημιολογία, τη διαχείριση θεμάτων υγείας. Το διαχειρίζεται ο Παγκόσμιος Οργανισμός Υγείας. Το ICD καταγράφει την ασθένεια αλλά είναι σχεδιασμένο έτσι ώστε να μπορεί να κατατάξει συμπτώματα, ευρήματα, παράπωνα, κοινωνικές επιπτώσεις και αίτια που προκαλούν την ασθένεια.
- Στο πεδίο «Όνομα Ασθένειας» ο χρήστης καλείται να εισάγει την ονομασία της ασθένειας για τη οποία ενδείκνυται η δραστική ουσία όπως έχει καταγραφεί στην βιβλιογραφία.
- Στο πεδίο «Τρόπος Χορήγησης» ο χρήστης καλείται να εισάγει αν το σκεύασμα χορηγείται δια στόματος, παρεντερικά, ενέσιμα ή με άλλο τρόπο χορήγησης.
- Στο πεδίο «Εμπορική Ονομασία» ο χρήστης καλείται να εισάγει τις εμπορικές ονομασίες του σκευάσματος που περιέχει την δραστική ουσία. Εδώ πρέπει να σημειωθεί η εμπορική ονομασία μπορεί να είναι παραπάνω από μια.
- Στο πεδίο «Κωδικός Φαρμάκου που προκαλεί επιπλοκή» ο χρήστης καλείται να εισάγει το ATC του σκευάσματος που έχει καταγραφεί ότι προκαλεί ή ενδέχεται να προκαλεί επιπλοκές.

- Στο πεδίο «Αντενδείξεις» ο χρήστης καλείται να εισάγει τις καταγεγραμμένες αντενδείξεις που συνδέονται με το σκεύασμα .

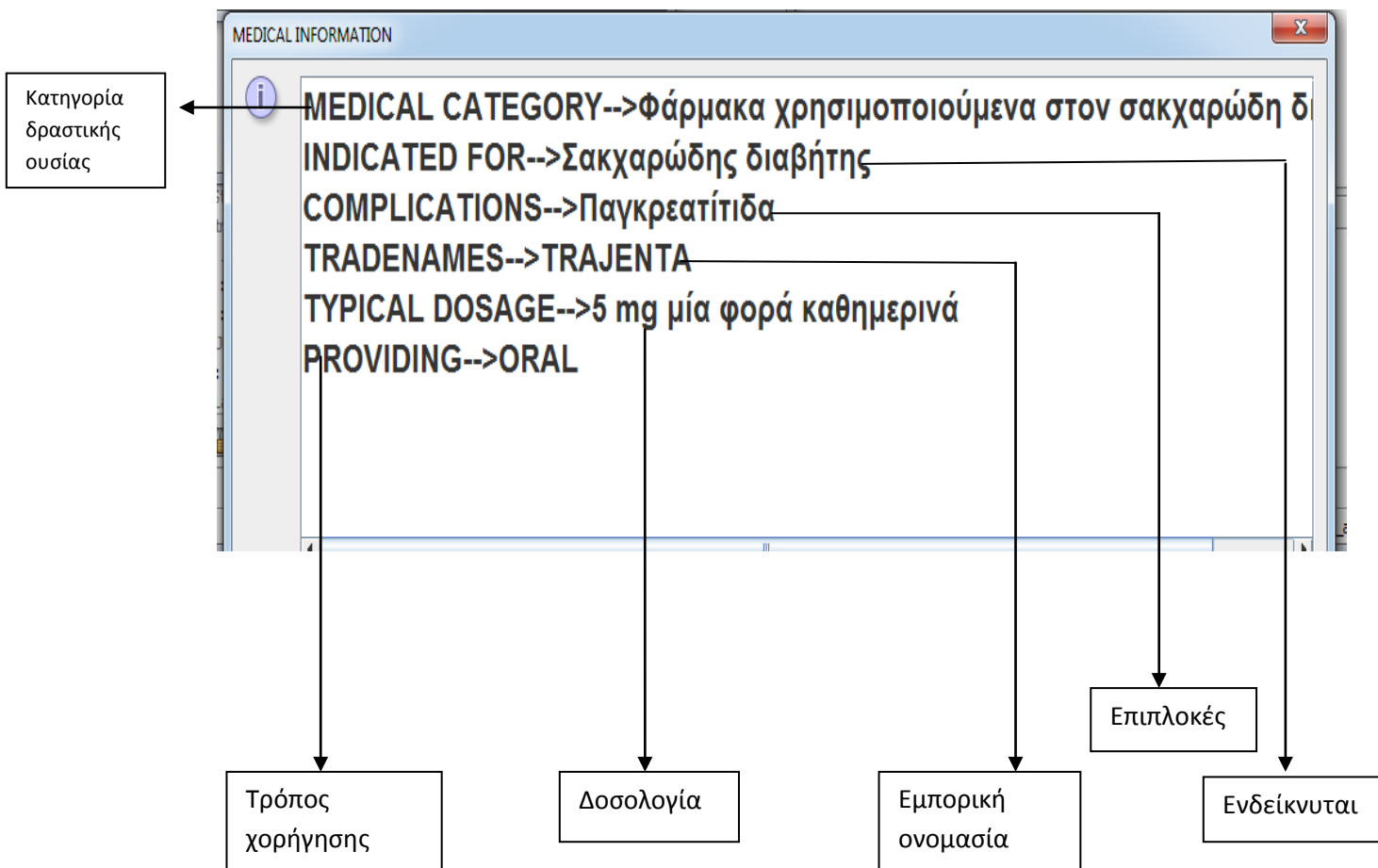
Αφού γίνει εισαγωγή των στοιχείων ο χρήστης πρέπει να πατήσει το κουμπί «Insert» ώστε να καταχωρηθούν τα στοιχεία στην βάση δεδομένων.

Επιλογή 2

Ο χρήστης πατώντας το κουμπί Search ο χρήστης μπορεί να αναζητήσει τις πληροφορίες που περιέχονται σε ένα φύλλο οδηγιών για τον χρήστη εισάγοντας το όνομα της δραστικής ουσίας.



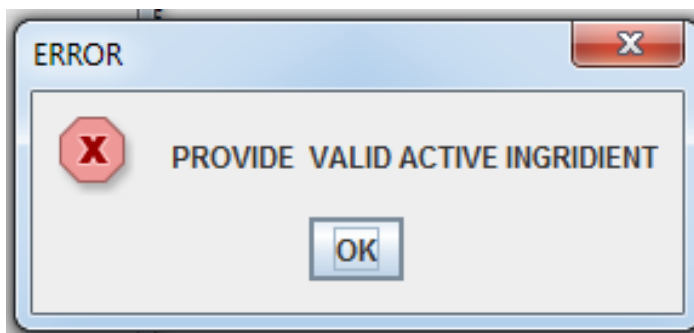
Πληκτρολογώντας την ονομασία της δραστικής ουσίας και πατώντας το κουμπί Search εμφανίζεται η επόμενη καρτέλα



Σε αυτή την καρτέλα εμφανίζονται με την ακόλουθη σειρά

- «Κατηγορία δραστικής ουσίας» : Όπου δίνονται στοιχεία για την φαρμακοθεραπευτική κατηγορία της δραστικής ουσίας
- «Ενδείκνυται» : Όπου εμφανίζεται η ονομασία της ασθένειας για την οποία ενδείκνυται η δραστική ουσία
- «Επιπλοκές» : Όπου εμφανίζεται το όνομα των επιπλοκών που έχουν καταγραφεί.
- «Εμπορικές ονομασίες» : Όπου εμφανίζονται οι εμπορικές ονομασίες που περιέχουν την δραστική ουσία.
- «Δοσολογία» : Όπου εμφανίζεται η συνιστώμενη δοσολογία.
- «Τρόπος χορήγησης» : Όπου εμφανίζεται ο τρόπος χορήγησης του σκευάσματος

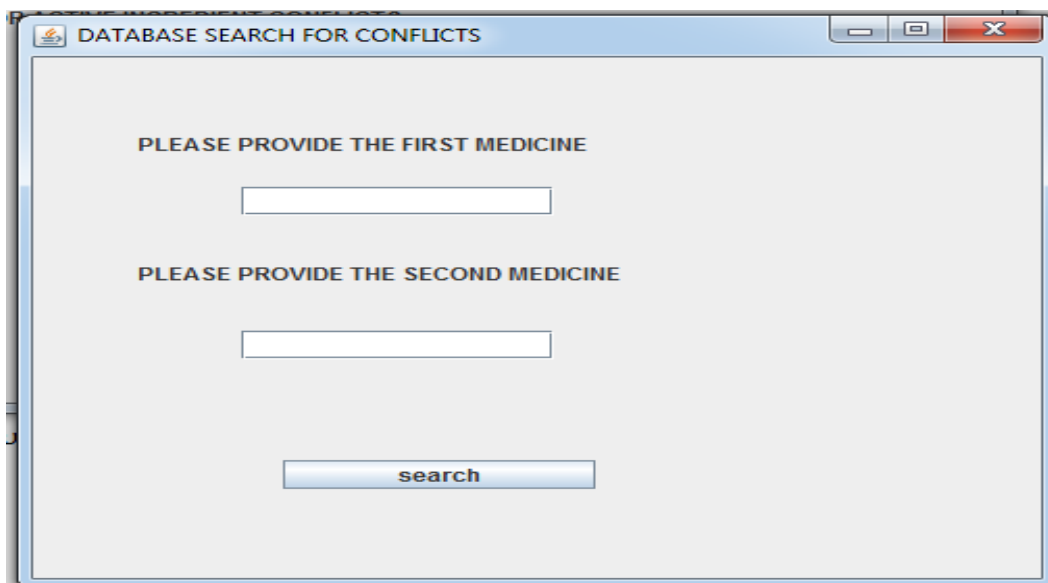
Σε περίπτωση που ο χρήστης δεν εισαγάγει σωστή δραστική ουσία το πρόγραμμα θα του εμφανίσει ένα μήνυμα λάθους



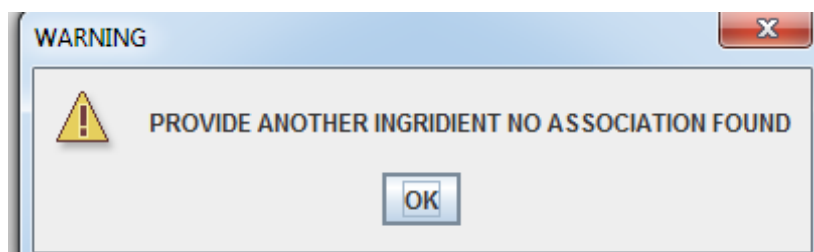
Εικόνα 3 Μήνυμα Λάθους

Επιλογή 3

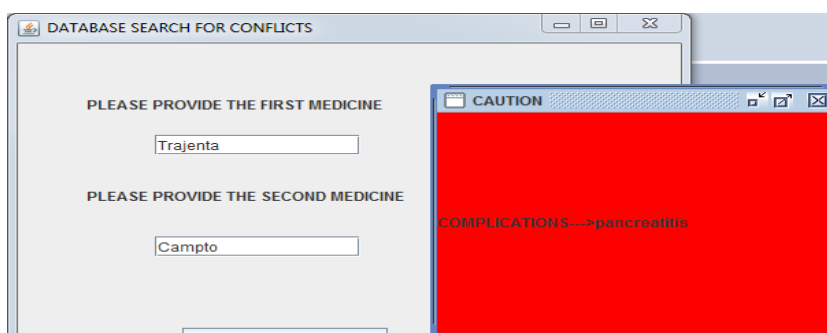
Ο χρήστης μπορεί να ελέγξει εάν τα φάρμακα που λαμβάνει προκαλούν μεταξύ τους επιπλοκές πατώντας το κουμπί Conflicts.



Στην καρτέλα που εμφανίζεται ο χρήστης μπορεί να καταχωρήσει την εμπορική ονομασία δυο φαρμάκων που χρησιμοποιεί ή πρόκειται να χρησιμοποιήσει. Αφού καταχωρήσει την ονομασία πρέπει να πατήσει το κουμπί Search. Εφόσον υπάρχει επιπλοκή θα εμφανιστεί μήνυμα όπου θα καταγράφεται η επιπλοκή αλλιώς θα εμφανιστεί μήνυμα στο οποίο θα αναφέρεται ότι δεν βρέθηκαν επιπλοκές.



Εικόνα 1 Η πινακίδα που ενημερώνει ότι δεν βρέθηκαν επιπλοκές



Εικόνα 2 Η πινακίδα όπου αναγράφεται η επιπλοκή

Αρχιτεκτονική Συστήματος

Το πρόγραμμα έχει συνταχθεί με την χρήση της αντικειμενοστραφούς γλώσσας προγραμματισμού JAVA με την χρήση του εργαλείου NETBEANS και της βάσης δεδομένων που υλοποιήθηκε με την MySql με το εργαλείο MySql Workbench.

Όσον αφορά τον κώδικα της JAVA δημιουργήθηκε ένα πακέτο με την ονομασία med_app. Μέσα σε αυτό δημιουργήθηκαν τέσσερις κλάσεις

Υλοποίηση INTRO

- Η κλάση INTRO είναι η εισαγωγική κλάση της εφαρμογής. Από αυτή την κλάση υλοποιείται η αρχική οθόνη της εφαρμογής όπου και αποτυπώνονται οι επιλογές που έχει ο χρήστης. Στην υλοποίηση της κλάσης χρησιμοποιήθηκε το Java Swing το οποίο αποτελεί ένα GUI εργαλείο με το οποίο ο προγραμματιστής μπορεί να σχεδιάσει γραφικά περιβάλλοντα. Με αυτό το εργαλείο σχεδιάστηκε όλο το γραφικό περιβάλλον της εφαρμογής. Για να επιτευχθεί το γραφικό περιβάλλον της κλάσης INTRO καλούνται οι υποκλάσεις του SWING:
JFRAME, JPANEL, JTABEL, JTEXTFIELD, JLABEL, JBUTTON.
JFRAME: Είναι η κλάση που δημιουργεί το αρχικό πλαίσιο για μια παραθυρική εφαρμογή. Τοποθετεί ένα πλαίσιο ένα τίτλο και ένα κουμπί με το οποίο κλείνει το παράθυρο
JPANEL: Είναι η κλάση που παρέχει ένα περιέκτη μέσα στο οποίο ο προγραμματιστής μπορεί να βάλει διάφορα γραφικά συστατικά. Ο περιέκτης θα πρέπει να βρίσκεται μέσα σε ένα πλαίσιο σαν αυτό που αναφέρθηκε προηγουμένως
JTABEL: Είναι η κλάση που χρησιμοποιείται για να απεικονίσει με τον καλύτερο τρόπο τα δεδομένα που θα εισαχθούν. Η κλάση αυτή δεν αποθηκεύει ούτε επεξεργάζεται τα δεδομένα είναι μόνο ένα απεικονιστικό εργαλείο.
JLABEL: Είναι η κλάση η οποία μπορεί να απεικονίσει κείμενο, φωτογραφία ή και τα δύο.
JBUTTON: Είναι η κλάση η οποία εισάγει ένα κουμπί το οποίο όταν πατηθεί θα παράξει ένα γεγονός.
JTEXTFIELD: Είναι η κλάση η οποία εισάγει ένα συστατικό το οποίο μπορεί να χρησιμοποιηθεί για την επεξεργασία μιας γραμμής κειμένου.

Η ιεραρχία με την καλούνται τα συστατικά του Swing πρέπει να είναι συγκεκριμένη αφού το αποτέλεσμα που παράγεται είναι στη πραγματικότητα η συνένωση διαφορετικών συστατικών. Έτσι καλείται πρώτα το πλαίσιο JFRAME μέσα στο οποίο τοποθετείται ο περιέκτης των συστατικών JPANEL μέσα στον οποίο τοποθετούνται τα συστατικά.

Το JTABEL είναι στατικό ενώ στην συνέχεια τοποθετούνται τα συστατικά

```
Label = new JLabel ("PRESS FOR ACTIVE INGREDIENT SEARCH");
label2 = new JLabel ("PRESS FOR ACTIVE INGREDIENT CONFLICTS");
label3 = new JLabel ("PRESS FOR INSERTING DATA");
label.setBounds(50, 100, 280, 20);
label2.setBounds(50, 150, 290, 80);
label3.setBounds(50,45,270,10);
button = new JButton("Search");
button2 = new JButton("Conflicts");
button3 = new JButton("Insert");
button.setBounds(120,120,150,20);
button2.setBounds(120,200,150,20);
button3.setBounds(120,60,150,20);
```

Κατά την εισαγωγή των συστατικών πρέπει να γίνουν τα εξής να οριστεί τι κείμενο θα περιέχουν και που θα τοποθετηθούν στον χώρο του περιέκτη. Έτσι κάθε φορά που καλείται μια κλάση δημιουργείται ένα καινούργιο αντικείμενο και δίνονται οι συντεταγμένες (X,Y,WIDTH,HEIGHT) από την μέθοδο setbounds

Ένα άλλο χαρακτηριστικό είναι ότι αφού πατηθεί το κουμπί που έχει οριστεί πρέπει να λάβει χώρα ένα γεγονός αυτό συμβαίνει με την χρήση του Action Listener.

- Γεγονότα
Τα γραφικά περιβάλλοντα δίνουν την δυνατότητα στον χρήστη να επιλέξει την πορεία της εργασίας του. Αυτό καθορίζεται από το πάτημα του ποντικιού ή από το πάτημα ενός κουμπιού στο πληκτρολόγιο κ.ο.κ όπου δημιουργείται ένα γεγονός. Όταν συμβαίνει ένα γεγονός δημιουργείται ένα αντικείμενο γεγονόςτος. Το αντικείμενο γεγονόςτος περιέχει πληροφορίες για το ίδιο το γεγονός καθώς και για την πηγή από την οποία προέρχεται. Επίσης ψάχνει μια ειδική μέθοδο η οποία θα χειριστεί το γεγονός αυτό.

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

Στην εφαρμογή γίνεται εισαγωγή της κλάσης Action Event και Action Listener το γεγονός που θα παραχθεί από το πάτημα του κουμπιού θα μεταφερθεί στο Action Listener αντικείμενο που του έχει ανατεθεί να διαχειριστεί το γεγονός.

```
button.addActionListener(this);
```

```
button2.addActionListener(this);
```

```
button3.addActionListener(this);
```

Όταν λοιπόν πατηθεί το αντίστοιχο κουμπί

```
public void actionPerformed(ActionEvent ae)
```

```
{
```

```
    Object src = ae.getSource();
```

```
    if(src == button){
```

```
        med_app.Main.main(args);
```

```
    }
```

```
    if(src == button2){
```

```
        med_app.Main2.main(args);
```

```
    }
```

```
    if(src == button3){
```

```
        med_app.Extra.main(args);
```

```
    }
```

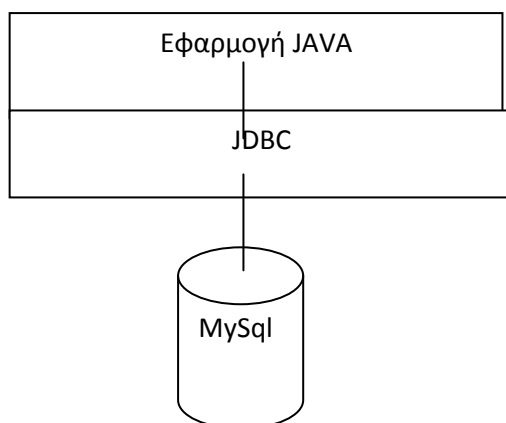
έχει οριστεί να ξεκινάει η αντίστοιχη εφαρμογή έτσι για το κουμπί 1 θα ξεκινήσει η κλάση MAIN για το κουμπί 2 η κλάση MAIN2 και για το κουμπί 3 η κλάση extra.

Ο χρήστης μπορεί να επιστρέφει κάθε φορά στην κλάση INTRO και να εκτελεί ξανά τις λειτουργίες που του παρέχονται και το πρόγραμμα κλείνει όταν πατηθεί το κουμπί που κλείνει το JFrame αυτής της κλάσης.

Υλοποίηση EXTRA

Στην κλάση EXTRA καινούργιο χαρακτηριστικό είναι η διασύνδεση με την βάση δεδομένων MySql. Η διασύνδεση επιτυγχάνεται με την χρήση της βιβλιοθήκης JDBC η οποία επιτρέπει στα προγράμματα της γλώσσας Java να εκτελούν εντολές SQL και να λαμβάνουν αποτελέσματα των εντελών αυτών από την βάση δεδομένων.

- JDBC: Ο βασικός στόχος της βιβλιοθήκης είναι να επιτρέπει σε προγράμματα της JAVA να εκτελούν εντολές SQL και να λαμβάνουν τα αποτελέσματα από την βάση δεδομένων. Οι κλάσεις που χρησιμοποιούνται από την JDBC είναι
 1. DRIVER MANAGER : Φορτώνει τους κατάλληλους οδηγούς και δημιουργεί σύνδεση με την βάση
 2. CONNECTION: Αντιπροσωπεύει μια σύνδεση σε μια συγκεκριμένη βάση δεδομένων. Η επικοινωνία της SQL με το πρόγραμμα JAVA πραγματοποιείται με τις μεθόδους της Connection
 3. STATEMENT: Αντιπροσωπεύει μια εντολή SQL. Οι κλήσεις για την εκτέλεση εντολών SQL πραγματοποιούνται με τις μεθόδους της Statement
 4. ResultSet: Αντιπροσωπεύει τα αποτελέσματα που λαμβάνονται από την εκτέλεση των εντολών SQL. Με την ResultSet μπορούμε να διαχειριστούμε τα δεδομένα που επιστρέφει η βάση δεδομένων



Η JDBC κατά την εφαρμογή της πρέπει να ακολουθείται από συγκεκριμένα βήματα ώστε να μπορεί να κάνει χρήση της βάσης δεδομένων και να μπορεί να στείλει Queries και να λάβει απαντήσεις.

- Εισαγωγή κατάλληλων κλάσεων
- Φόρτωση κατάλληλου οδηγού JDBC
- Προσδιορισμός πηγής δεδομένων
- Δημιουργία ενός αντικειμένου Connection
- Δημιουργία ενός αντικειμένου Statement
- Query στην βάση δεδομένων με την βοήθεια Statement
- Λήψη δεδομένων από το αντικείμενο ResultSet
- Κλείνει τα αντικείμενα ResultSet, Statement, Connection

```

public static void ConnectwithMysql(){
connection();
String host = "jdbc:mysql://localhost/medicine?useOldAliasMetadataBehavior=true";
String username = "root";
String password = "";
try {

```

```
Connection connect = DriverManager.getConnection(host, username, password) ;
```

```
Statement statement = connect.createStatement();
```

```
System.out.println("Connection established");
```

Στον κώδικα της εφαρμογής καλείται η connection και ορίζεται η διαδρομή η οποία ορίζει την θέση της βάσης. Στην συνέχεια παρέχονται τα credentials της βάσης και δημιουργούνται αντικείμενα των κλάσεων Connect και Statement.

Στην συνέχεια δημιουργούνται αλφαριθμητικά στα οποία καταχωρούνται τα queries προς την SQL

```
String sql = "INSERT INTO `ACTIVE INGREDIENT` (CODE,CATEGORY,NAME) VALUES (" +ac_in+" ,"+ac_in2+" ,"+ac_in3+" )";
```

```
String sql2 = "INSERT INTO `DISEASES` (DCODE,NAMES) VALUES (" +ac_in4+" ,"+ac_in5+" )";
```

```
String sql3 = "INSERT INTO `TRADE NAMES` (TRCODE,Medicine_Name,`active ingredient_CODE`) VALUES (" +ac_in12+" ,"+ac_in7+" ,"+ac_in+" )";
```

```
String sql4 = "INSERT INTO `active ingredient_indicated_for_diseases` (`active ingredient_CODE`, `DISEASES_DCODE`,DOSAGE,PROVIDING) VALUES (" +ac_in+" ,"+ac_in4+" ,"+ac_in8+" ,"+ac_in9+" )";
```

```
String sql5 = "INSERT INTO `active ingredient_interacts_active ingredient` (`active ingredient_CODE`, `active ingredient_CODE1`,Complications) VALUES (" +ac_in+" ,"+ac_in6+" ,"+ac_in10+" )";
```

```
String sql6 = "INSERT INTO `active ingredient_not_indicated_for_diseases` (`active ingredient_CODE`,Complication) VALUES (" +ac_in+" ,"+ac_in11+" )";
```

Στο συγκεκριμένο κομμάτι κώδικα δίνεται εντολή να εισαχθούν νέες καταχωρήσεις. Οι καταχωρήσεις αυτές έχουν εισαχθεί από τον χρήστη και το πρόγραμμα τις «τραβάει» από εκεί.

Επόμενη διαδικασία είναι η εκτέλεση των εντολών.

```
int result = statement.executeUpdate(sql);
```

```
int result2 = statement.executeUpdate(sql2);
```

```
int result3 = statement.executeUpdate(sql3);
```

```
int result4 = statement.executeUpdate(sql4);
```

```
int result5 = statement.executeUpdate(sql5);
```

```
int result6 = statement.executeUpdate(sql6);
```

Σε αυτό το κομμάτι κώδικα καλείται η μέθοδος execute update το αντικείμενο αυτό χρησιμοποιείται για να εκτελεστεί ένα SQL Statement και επιστρέφει ένα ακέραιο που σηματοδοτεί τον αριθμό των γραμμών που επηρεάστηκαν από τις εντολές που εκτελέστηκαν. Με αυτή την υλοποίηση επιτυγχάνεται η εισαγωγή δεδομένων από τον χρήστη στην βάση δεδομένων. Σε αυτό το σημείο θα πρέπει να τονιστεί ότι έχει ληφθεί μέριμνα ώστε να μην επιτρέπονται χαρακτήρες πέρα των αλφαριθμητικών δηλαδή όλα να επιτρέπονται όλα τα νούμερα και οι αριθμοί και ανάλογα την περίπτωση να μην επιτρέπονται ούτε οι αλφαβητικοί χαρακτήρες. Για να επιτευχθεί αυτό προστέθηκε μια παράμετρος που μόλις ανιχνεύσει μη αποδεκτούς χαρακτήρες του αφαιρεί ώστε να μην περαστούν στην βάση δεδομένων. Η υλοποίηση είναι σχετικά απλή από την στιγμή που ο χρήστης πάτησε ένα μην αποδεκτό χαρακτήρα θα αντικατασταθεί με κενό χαρακτήρα και έτσι δεν θα περαστεί στο σύστημα.


```
ac_in = textbox.getText();
```

```
ac_in = ac_in.replaceAll("[^a-zA-Z0-9]", "");
```

Επίσης έχει προβλεφθεί να ειδοποιείται ο χρήστης ώστε να μην αφήνει κενά πεδία με μήνυμα που τον ενημερώνει ποιο πεδίο είναι κενό

```
if (textbox3.getText().trim().length()== 0){
```

```
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 3", "ERROR", 0, null);
```

```
}
```

Υλοποίηση MAIN

Στην κλάση MAIN υλοποιείται η αναζήτηση βάσει ονόματος δραστικής ουσίας στην βάση δεδομένων. Η διαδικασία σύνδεσης και επικοινωνίας με την SQL ακολουθεί ακριβώς την ίδια μεθοδολογία με την προηγούμενη υλοποίηση αλλά το καινούργιο χαρακτηριστικό που εισάγεται εδώ είναι η διαδικασία απεικόνισης των αποτελεσμάτων που επιστρέφει η βάση δεδομένων.

```
ResultSet result = statement.executeQuery(sql);
```

```
displayResults(result);
```

Αφού δόθηκε το query στην βάση δεδομένων καλείται η μέθοδος executeQuery

- Executing Queries: Για να εκτελεστεί ένα query πρέπει να γίνει κλήση μιας μεθόδου execute από το σύστημα. Η executeQuery συγκεκριμένα επιστρέφει ένα αντικείμενο ResultSet.
- ResultSet: Η διεπαφή ResultSet παρέχει μεθόδους για ανάκτηση και επεξεργασία αποτελεσμάτων από queries που μόλις έχουν εκτελεστεί. Τα αντικείμενα που προκύπτουν μπορούν να έχουν διάφορες λειτουργίες.

Στο πρόγραμμα το αντικείμενο ResultSet ονομάζεται result και εισάγεται σαν παράμετρος στην κλάση displayResults.

```
ResultSetMetaData rmeta = result.getMetaData();
```

```
int numColumns=result.getMetaData().getColumnCount();
```

Στην κλάση displayResults γίνεται χρήση της μεθόδου getMetaData().

- ResultSetMetaData: Μια διεπαφή η οποία παρέχει μεθόδους για να γίνει συλλογή για τα χαρακτηριστικά των στηλών από ένα αντικείμενο ResultSet. Αφού παρθούν τα στοιχεία καλείται μια μέθοδος .getColumnCount(); η οποία επιστρέφει τον αριθμό των στηλών.

Ακολουθως με την χρήση while και της μεθόδου .next () ο cursor διατρέχει κάθε γραμμή από το αντικείμενο result.

```
while (result.next())
```

Στην συνέχεια και αφού γίνει έλεγχος ότι δεν είναι κενό από εγγραφές το αντικείμενο το διατρέχει της στήλης με μια for.

```
if (result.getString(1) != null) {
```

```
for (int i=1;i<numColumns;i++)
```

Με την χρήση των μεθόδων `getColumnName` και `getString` παίρνει το όνομα της εκάστοτε στήλης και του αντίστοιχου αλφαριθμητικού που περιέχεται

```
b[i]= rmeta.getColumnName(i);
a[i] = result.getString(i).trim()
```

Υλοποίηση MAIN2

Στην κλάση MAIN υλοποιείται η αναζήτηση επιπλοκών μεταξύ των φαρμάκων που δίνει ο χρήστης στο γραφικό περιβάλλον που του παρέχεται από το πρόγραμμα. Εδώ η υλοποίηση ακολουθεί τον τρόπο σύνδεσης με την βάση δεδομένων όπως ακριβώς και στην κλάση MAIN για αυτό και δεν αναλύεται ξανά και στην συνέχεια με την χρήση της `ResultSet` απεικονίζονται τα δεδομένα.

Πιο αναλυτικά

```
ResultSetMetaData rmeta = result.getMetaData();
```

```
int numColumns=result.getMetaData().getColumnCount();
```

Αφού πρώτα γίνει συλλογή των στοιχείων για τις στήλες που επιστραφεί από τα ερωτήματα προς την βάση δεδομένων στην συνέχεια γίνεται καταμέτρηση των στηλών.

if (`result.next()==true`) με την χρήση της if εξετάζουμε αν υπάρχει εγγραφή στην πρώτη μόνο γραμμή αφού εκεί βρίσκονται τα αποτελέσματα που μας ενδιαφέρουν.

```
do
```

```
{
```

```
    s1 = rmeta.getColumnName(1);
```

```
    s2 = result.getString(1);
```

```
    sf = s1+"--->" +s2;
```

```
JFrame.setDefaultLookAndFeelDecorated(true);
```

```
JFrame frame = new JFrame();
```

```
frame.setTitle("CAUTION");
```

```
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
JLabel label = new JLabel(sf);
```

```
frame.add(label);
```

```
frame.pack();
```

```
frame.setSize(new Dimension(300, 250));
```

```
frame.getContentPane().setBackground(Color.RED);
```

```
frame.setVisible(true)
```

```
while (result.next()) ;
```

Στην συνέχεια με την χρήση της `do – while` επιτυγχάνεται ότι τουλάχιστον μια φορά θα έχουμε εκτέλεση του κώδικα που βρίσκεται μέσα στο πεδίο δράσης του βρόγχου. Τα αποτελέσματα της εκτέλεσης της `meta.getColumn(1)`; αποθηκεύονται στο `s1` και της `result.getString(1)`; στο `s2`. Έπειτα γίνεται η δημιουργία του παραθύρου που θα εμφανιστεί στον χρήστη εδώ αξίζει να σημειωθεί η χρήση της `setDefaultLookAndFeelDecorated(true)`. Το εργαλείο για την δημιουργία γραφικού περιβάλλοντος που παρέχει η JAVA στις τελευταίες εκδόσεις της ονομάζεται `Swing` και η βασική ιδέα γύρω από την οποία χτίστηκε είναι η παραμετροποίηση του. Ορίζοντας σαν `true` την χρήση του επιτρέπεται στο εκάστοτε λογισμικό που χρησιμοποιείται να ορίζει τα χαρακτηριστικά του παράθυρου που δημιουργείται.

Εξαιρέσεις και ισχυρισμοί

Σε κάθε κλάση του προγράμματος γίνεται χρήση εξαιρέσεων και είναι σημαντικό να γίνει μια αναφορά σε αυτό το κομμάτι του κώδικα. Όταν συμβεί κάποιο πρόβλημα στην JAVA δημιουργείται μια εξαίρεση. Το πλεονέκτημα από την χρήση εξαιρέσεων είναι ότι μπορεί να διαχωριστεί ο κώδικας που χειρίζεται την εξαίρεση από τον υπόλοιπο κώδικα. Ο χειρισμός των εξαιρέσεων βασίζεται στο ότι όταν παραχθεί μια εξαίρεση δεν μπορεί πάντα να αντιμετωπιστεί αυτόματα. Εάν όμως υπάρχει κώδικας που μπορεί να διαχειριστεί εξαιρέσεις τότε η εξαίρεση συλλαμβάνεται (`catch`) και να υποστεί διαχείριση. Στον κώδικα του προγράμματος οι εξαιρέσεις που πρέπει να διαχειριστούν αφορούν κυρίως θέματα που έχουν να κάνουν με την επικοινωνία της JAVA-SQL αλλά όχι αποκλειστικά επίσης υπάρχουν εξαιρέσεις που έχουν να κάνουν με χειρισμό `pointers`.

Όταν χρειάζεται να συλληφθεί μια εξαίρεση δημιουργείται ένα τμήμα `try` μέσα στο οποίο πρέπει να τοποθετηθεί τον κώδικα της μεθόδου το οποίο ενδέχεται να δημιουργήσει μια εξαίρεση. Στην συνέχεια πρέπει να δημιουργηθεί ένα τμήμα `catch` το οποίο προσδιορίζει το τύπο της εξαίρεσης που ενδέχεται να συλληφθεί και τον κώδικα για τον χειρισμό της. Το τμήμα `catch` πρέπει να ακολουθήσει από μια παρένθεση όπου θα ορίζεται ο τύπος της εξαίρεσης με την οποία ασχολείται το `catch`.

```
try{...
```

```
catch (NullPointerException e){
```

```
JOptionPane.showMessageDialog( null, "PROVIDE VALID ACTIVE INGRIDIENT", "ERROR",
0, null);
```

```
e.printStackTrace();
```

Μετά το `try` ακολουθεί ο κώδικας υπό εξέταση στην συνέχεια υπάρχει ένα `catch` με την εξαίρεση που ενδέχεται να δημιουργηθεί και ο κώδικας που θα αντιμετωπίσει τη ενδεχόμενη εξαίρεση.

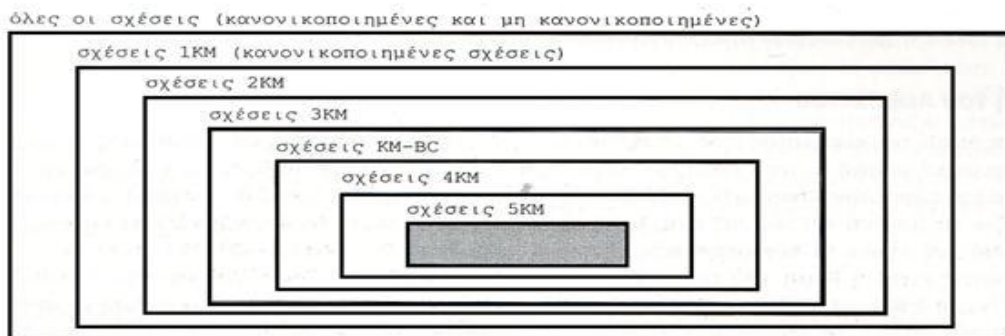
Η Βάση Δεδομένων

Με τον όρο βάση δεδομένων εννοείτε μια οργανωμένη συλλογή από δεδομένα τα οποία αναφέρονται σε ένα αντικείμενο. Το αντικείμενο μπορεί να είναι μια επιχείρηση, ένας οργανισμός ή ακόμα και ένα άτομο. Η λειτουργία μιας βάσης δεδομένων είναι η εξής κάθε πληροφορία που εισάγεται στην βάση δεδομένων αποθηκεύεται σε μια καθορισμένη θέση μέσα σε αυτή. Εάν η βάση είναι σωστά οργανωμένη δίνει την δυνατότητα να αντληθούν από αυτή σύνθετες πληροφορίες από την στιγμή που θα δοθούν τα σωστά ερωτήματα προς αυτή. Μια βάση δεδομένων δημιουργείται από ένα ειδικό πακέτο λογισμικού που ονομάζεται Σύστημα Διαχείρισης Βάσης Δεδομένων (Σ.Δ.Β.Δ). Το Σ.Δ.Β.Δ εκτός από να δημιουργεί την βάση έχει την δυνατότητα να αποθηκεύει στοιχεία σε αυτή να πραγματοποιεί ταξινομήσεις και να συντηρεί τα στοιχεία που υπάρχουν σε αυτή καθώς και να τα προστατεύει από λάθος χειρισμούς. Κάθε μεγάλη εταιρεία του κλάδου της ανάπτυξης λογισμικού έχει δημιουργήσει πακέτα Σ.Δ.Β.Δ. Υπάρχουν διάφορα μοντέλα βάσεων δεδομένων όμως το πλέον διαδεδομένο είναι το σχεσιακό μοντέλο. Στο σχεσιακό μοντέλο τα δεδομένα οργανώνονται σε πίνακες. Κάθε πίνακας αποτελείται από από γραμμές και στήλες. Κάθε γραμμή αντιστοιχεί σε μια οντότητα (πρόσωπο ,

αντικείμενο) ενώ κάθε στήλη προσδιορίζει χαρακτηριστικά αυτής της οντότητας. Κάθε εγγραφή αποτελείται από συγκεκριμένες τιμές πεδίων. Ένα από τα πεδία κάθε εγγραφής πρέπει να ταυτοποιεί μοναδικά την εγγραφή έτσι ώστε αυτή να ξεχωρίζει απόλυτα από τις υπόλοιπες. Το πεδίο αυτό ονομάζεται πρωτεύον κλειδί. Μια τυπική εφαρμογή αποτελείται από πολλούς πίνακες που συνδέονται μεταξύ τους. Η σύνδεση γίνεται με ένα κοινό πεδίο. Το πεδίο αυτό στον ένα πίνακα αποτελεί το πρωτεύον κλειδί ενώ στον άλλο το ξένο κλειδί. Στο σχεσιακό μοντέλο όλοι οι πίνακες πρέπει να είναι κανονικοποιημένοι .

Για να λειτουργεί σωστά μια βάση δεδομένων πρέπει να ακολουθεί τα εξής βήματα κανονικοποίησης.

1. Πρώτη κανονική μορφή: Εδώ επιβάλλεται να υπάρχουν μοναδικές τιμές για κάθε όνομα γνωρίσματος ,όλες οι τιμές κάθε γνωρίσματος να είναι του ίδιου τύπου, κάθε τιμή γνωρίσματος να είναι ατομική
2. Δεύτερη κανονική μορφή: Μια σχέση είναι σε δεύτερη κανονική μορφή όταν είναι σε πρώτη κανονική μορφή και κάθε ένα από τα γνωρίσματα της που δεν είναι το πρωτεύον κλειδί εξαρτώνται από το πρωτεύον κλειδί
3. Τρίτη κανονική μορφή: Μια σχέση είναι σε Τρίτη κανονική μορφή όταν είναι σε δεύτερη κανονική μορφή και δεν περιέχει μεταβατικές εξαρτήσεις
4. Για την βέλτιστη λειτουργία μιας βάσης δεδομένων πρέπει να βρίσκεται την Boyce Codd κανονική μορφή. Μια σχέση είναι σε Boyce Codd κανονική μορφή όταν είναι σε τρίτη κανονική μορφή και κανένα γνώρισμα της σχέσης δεν εξαρτάται συναρτησιακά από κανένα υποσύνολο των γνωρισμάτων της σχέσης που δεν είναι πρωτεύον κλειδί.



Εικόνα 3 Κανονικές μορφές

Η βάση δεδομένων που έχει σχεδιαστεί περιέχει πληροφορίες για ένα σύστημα διαχείρισης των δραστικών ουσιών των φαρμάκων. Συγκεκριμένα το σύστημα θα διαχειρίζεται τις ακόλουθες έννοιες:

Δραστική ουσία: Στο σύστημα θα αποθηκεύονται πληροφορίες για δραστικές ουσίες, για κάθε μια από τις οποίες θα καταγράφεται το όνομά της ,ο κωδικός της ο οποίος είναι μοναδικός, το πεδίο δράσης της και οι αντενδείξεις που έχει κάθε ουσία.

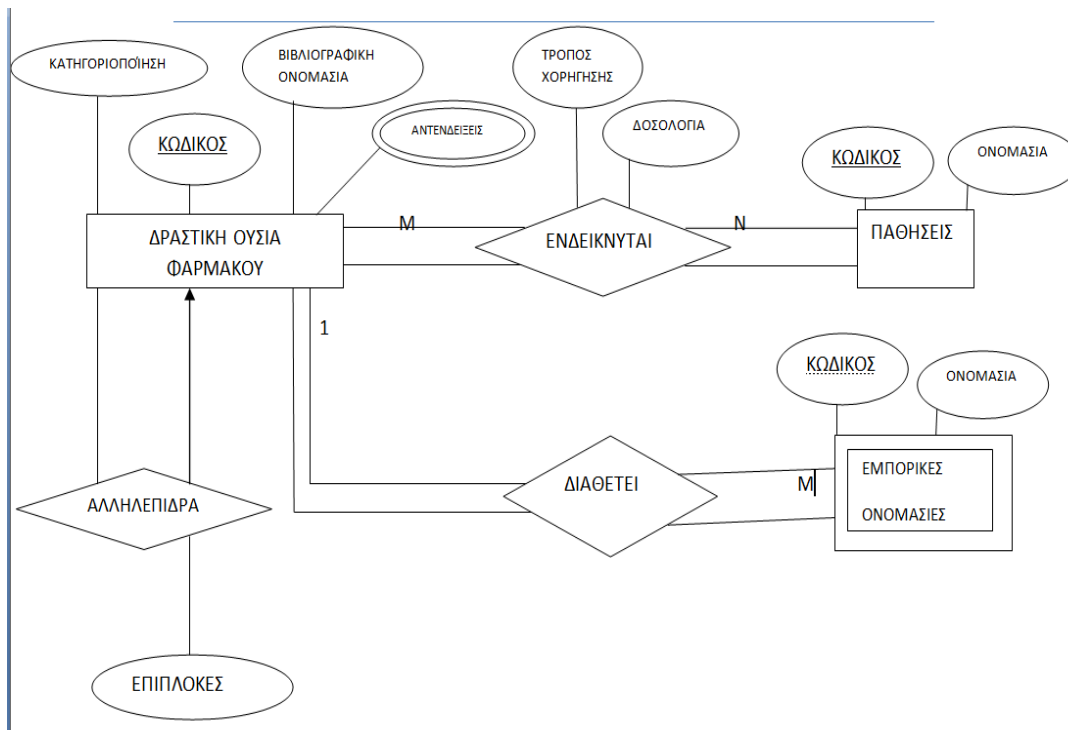
Ασθένειες: Καταγράφονται επίσης τα στοιχεία των ασθενειών. Συγκεκριμένα καταγράφεται ο μοναδικός κωδικός κάθε ασθένειας καθώς και η βιβλιογραφική της ονομασία.

Εμπορική Ονομασία: Καταγράφονται οι εμπορικές ονομασίες που μπορεί να έχει κάθε δραστική ουσία και συγκεκριμένα καταγράφεται ο μοναδικός κωδικός , ο κωδικός της δραστικής ουσίας και οι εμπορικές ονομασίες

Για τις σχέσεις μεταξύ των οντοτήτων ισχύουν οι εξής περιορισμοί:

- Κάθε δραστική ουσία έχει τουλάχιστον μια εμπορική ονομασία
- Οι δραστικές ουσίες αλληλεπιδρούν μεταξύ τους
- Τουλάχιστον μία δραστική θεραπεύει τουλάχιστον μια ασθένεια
- Κάθε εμπορική ονομασία πρέπει να αντιστοιχεί τουλάχιστον σε μια δραστική ουσία

ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ



ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Το σχεσιακό μοντέλο που προκύπτει είναι το εξής:

ΔΡΑΣΤΙΚΗ ΟΥΣΙΑ

ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ	<u>ΚΩΔΙΚΟΣ</u>	ΟΝΟΜΑΣΙΑ
-----------------	----------------	----------

ΑΝΤΕΝΔΕΙΞΕΙΣ

<u>ΑΝΤΕΝΔΕΙΞΗ</u>	<u>ΚΩΔ.ΔΡΑΣΤΙΚΗΣ</u>
-------------------	----------------------

ΕΝΔΕΙΚΝΥΤΑΙ

<u>ΚΩΔ.ΔΡΑΣΤΙΚΗΣ</u>	<u>ΚΩΔ.ΠΑΘΗΣΗΣ</u>	ΤΡΟΠΟΣ ΧΟΡΗΓΗΣΗΣ	ΔΟΣΟΛΟΓΙΑ
----------------------	--------------------	------------------	-----------

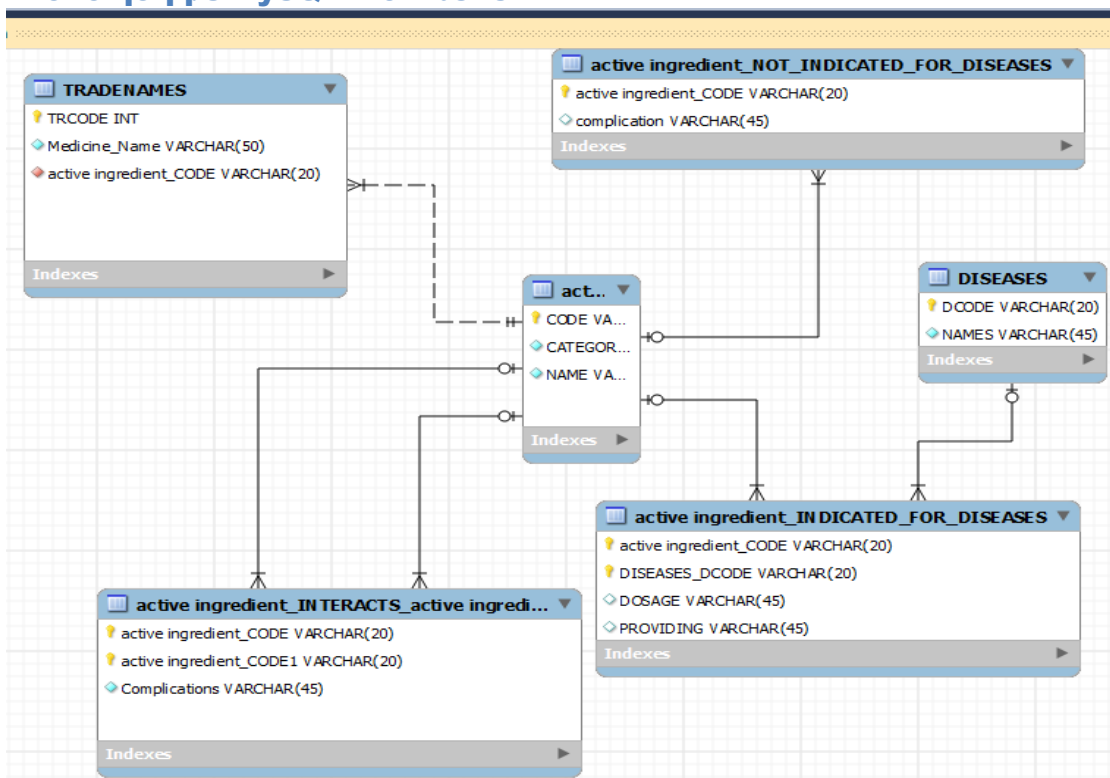
ΠΑΘΗΣΗ

<u>ΚΩΔΙΚΟΣ</u>	ΟΝΟΜΑΣΙΑ
----------------	----------

ΕΜΠΟΡΙΚΗ ΟΝΟΜΑΣΙΑ

ΟΝΟΜΑΣΙΑ	<u>ΚΩΔΙΚΟΣ</u>	<u>ΚΩΔΙΚΟΣ ΔΡΑΣΤΙΚΗΣ</u>
----------	----------------	--------------------------

Υλοποίηση με MySQL Workbench



ΠΕΡΙΟΡΙΣΜΟΙ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

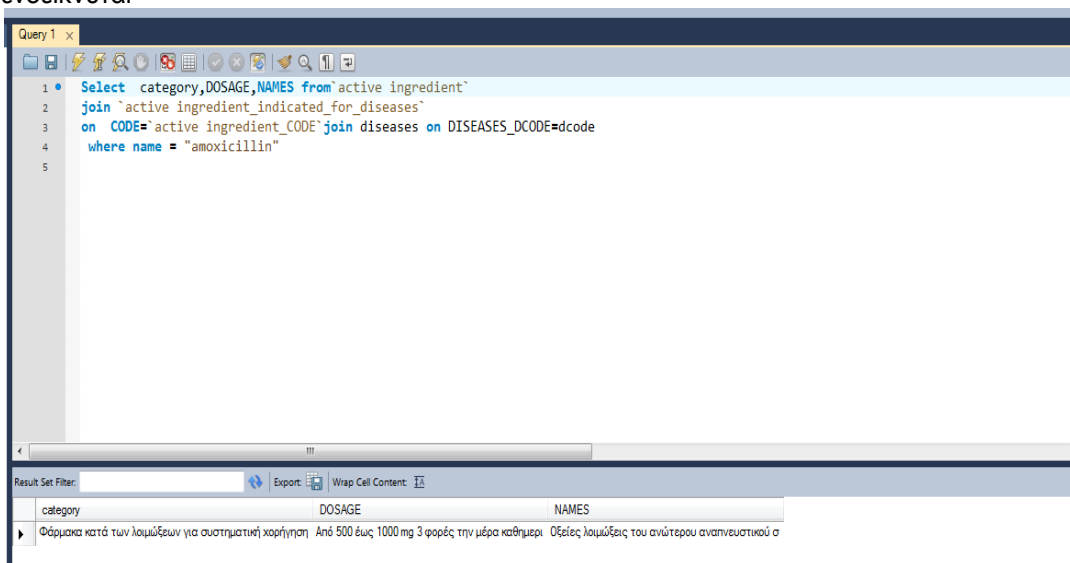
Τα γεγονότα που δεν μπορούν να αντικατοπτριστούν στη βάση δεδομένων είναι ο χρόνος που θα διαρκεί η θεραπεία καθώς και τυχόν συνδυασμός δραστικών που ίσως χρειάζεται μια πάθηση.

ΕΙΣΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

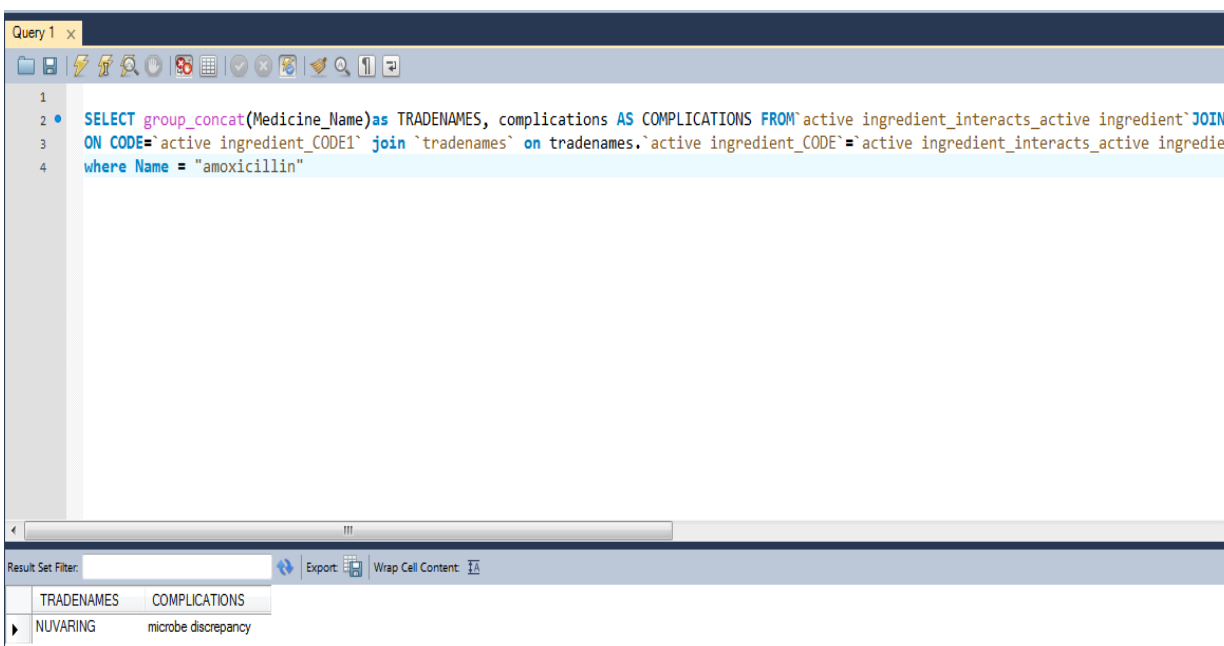
- Η οντότητα active ingredient θα έχει code που θα είναι και το κλειδί. Θα είναι τύπου varchar 20 θέσεων. Θα έχει CATEGORY επίσης και NAME τύπου χαρακτήρων (varchar) 1500 και 105 θέσεων.
- Η συσχέτιση ingredient_INDICATED_FOR_DISEASES θα έχει χαρακτηριστικό την DOSAGE και PROVIDING
- Η οντότητα DISEASES θα έχει DCODE κλειδί τύπου varchar 20 θέσεων. NAMES varchar 45 θέσεων
- Η συσχέτιση active ingredient_INTERACTS_active ingredient θα έχει το χαρακτηριστικό Complications varchar 100 θέσεων και θα είναι αναδρομική συσχέτιση
- Το πλείοτιμο χαρακτηριστικό ingredient_NOT_INDICATED_FOR_DISEASES θα έχει χαρακτηριστικό complication varchar 245 θέσεων
- Η αδύναμη οντότητα TRADENAMES θα έχει το TRCODE κλειδί της αδύναμης οντότητας, ακέραιος και το active ingredient_CODE varchar 20 θέσεων που θα είναι FOREIGN KEY καθώς και το πεδίο Medicine_Name varchar 50 θέσεων

ΕΡΩΤΗΣΕΙΣ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

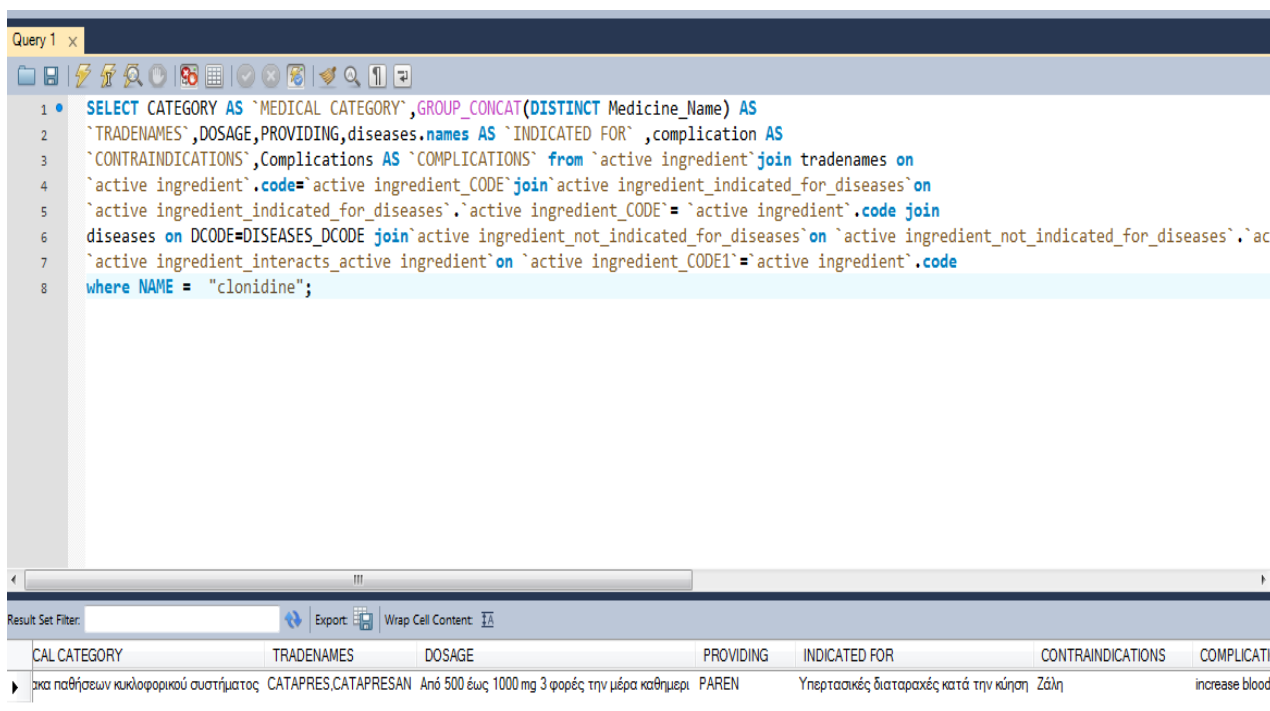
Εμφάνιση κατηγοριοποίησης, δοσολογίας της δραστικής ουσίας και ασθένειας για την οποία ενδείκνυται



Βάσει της ονομασίας μιας δραστικής ουσίας ποια εμπορική ονομασία έχει η δραστική ουσία που δεν πρέπει να χορηγείται μαζί της και τη επιπλοκή που παρουσιάζεται



Εμφάνιση της κατηγοριοποίησης, της εμπορικής ονομασίας ,της δοσολογίας, του τρόπου χορήγησης , της ασθένειας για την οποία ενδείκνυται, των αντενδείξεων και των επιπλοκών που μπορεί να προκαλέσει μια δραστικής ουσία.



Συμπεράσματα και Μελλοντικές Επεκτάσεις

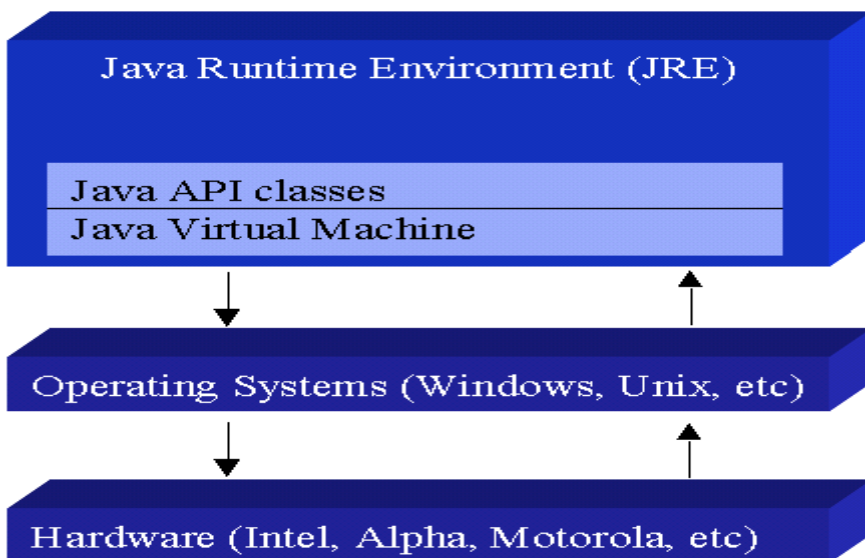
Συμπεράσματα

Τα συμπεράσματα που μπορούν να εξαχθούν από την κωδικοποίηση του λογισμικού με τα εργαλεία της JAVA και MySql είναι θετικά. Τα εργαλεία συνεργάζονται αρκετά καλά μεταξύ τους και η βιβλιογραφία που παρέχεται είναι αρκετά εκτενής όσον αφορά και τα δυο εργαλεία και τον μεταξύ του προγραμματισμό.

Ποιο συγκεκριμένα η JAVA παρέχει την δυνατότητα στον προγραμματιστή να δημιουργήσει προγράμματα που τρέχουν σε μια ποικιλία υπολογιστών και προγραμμάτων.

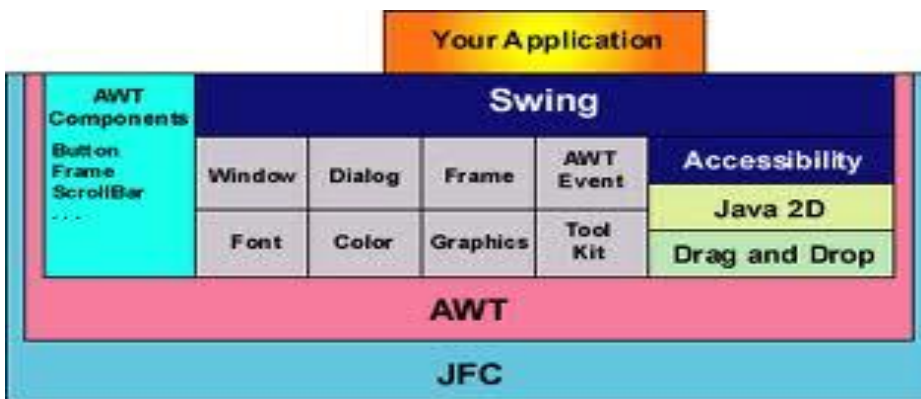
Σε γενικές γραμμές στην Java το περιβάλλον ανάπτυξης περιλαμβάνει δυο μέρη

- Τον μεταγλωττιστή ο οποίος αντί για δυαδικό αρχείο δημιουργεί ένα ανεξάρτητο από πλατφόρμα κώδικα byte
- Τον διερμηνευτή ο οποίος ονομάζεται και Java Virtual Machine και διαβάζει τον κώδικα byte και εκτελεί τις κατάλληλες εντολές γλώσσας μηχανής που αντιστοιχούν στο συγκεκριμένο επεξεργαστή. Ο διερμηνευτής προσαρμόζεται στο κάθε περιβάλλον.



Εικόνα 4 Java Virtual Machine

Η JAVA διαθέτει ισχυρό υποσύστημα γραφικών με το οποίο μπορεί να σχεδιαστούν παράθυρα και να βελτιωθεί η επικοινωνία να ανθρώπου υπολογιστή σε αρκετά ικανοποιητικό βαθμό. Η βιβλιοθήκη πακέτων με το όνομα Swing περιλαμβάνει τα απαραίτητα συστατικά για την δημιουργία του γραφικού περιβάλλοντος. Είναι δε η επέκταση του πακέτου γραφικών AWT το οποίο είχε χρήση μέχρι την έκδοση 1.2. Τα συστατικά της βιβλιοθήκης είναι ελαφρά και χρησιμοποιήσιμα σε οποιαδήποτε πλατφόρμα.



Εικόνα 5 Αρχιτεκτονική του Swing

Όσον αφορά την διασύνδεση με βάσεις δεδομένων η εταιρεία Sun ανέπτυξε την διασύνδεση JDBC η οποία καθορίζει τον τρόπο επικοινωνίας μιας εφαρμογής με σχεσιακές βάσεις δεδομένων. Με αυτό τον τρόπο η γλώσσα μπορεί να υποστηρίξει βάσεις δεδομένων οι οποίες προέρχονται από διαφορετικά εργαλεία.

Η βάση δεδομένων σχεδιάστηκε με το εργαλείο MySql workbench το οποίο είναι το γραφικό περιβάλλον το οποίο επιτρέπει την δημιουργία ,διαχείριση και επεξεργασία των βάσεων δεδομένων. Όχι ιδιαίτερα σταθερό αλλά αρκετά βολικό σύστημα για την σχεδίαση της βάσης δεν παύει να είναι ένα εργαλείο αποκλειστικά για εκπαιδευτικούς σκοπούς αφού στην πιο βαριά χρήση παρουσιάζει αρκετά προβλήματα. Η σχεδίαση της βάσης αποτελεί την καρδιά του λογισμικού αφού εκεί γίνεται η επεξεργασία των δεδομένων για την εξαγωγή των αποτελεσμάτων προς τον χρήστη. Ιδιαίτερη μνεία δόθηκε στην ανάπτυξη σωστών ερωτημάτων προς την βάση δεδομένων.

Σε αυτή την κατεύθυνση έγιναν αρκετές δοκιμές ώστε να γίνει βέβαιο ότι οι απαντήσεις που παράγονται καλύπτουν το σωστό φάσμα δεδομένων. Στα ερωτήματα προς την βάση έγινε

αρκετή χρήση της εντολής join που χρησιμοποιείτε κατά κύριο λόγο στον συνδυασμό γραμμών από διαφορετικούς πίνακες. Επίσης δόθηκε μεγάλη προσοχή στην σωστή αντιστοίχιση των στοιχείων που αφορούν τις φαρμακευτικές ουσίες. Σε αυτό ιδιαίτερη βοήθεια παρείχε ο ιατρός κος Γουρλής ωστόσο επειδή το δείγμα που έφτασε στα χέρια του γράφοντος ήταν μικρό από ένα σημείο και μετά και για λόγους ελέγχου της ορθής λειτουργίας του λογισμικού έχει συμπεριληφθεί και τυχαίο υλικό κυρίως από το διαδίκτυο.

Σε αυτό το σημείο πρέπει να τονιστεί το εξής στην Ελλάδα οποιαδήποτε προσπάθεια έρευνας και ανάπτυξης λογισμικού είναι μια επίπονη διαδικασία αφού ούτε διαθέσιμοι πόροι υπάρχουν αρκετοί ούτε αρκετό υλικό-λογισμικό υπάρχει διαθέσιμο ώστε να μπορεί να καθοδηγήσει τον προγραμματιστή σε μια κατεύθυνση. Κυρίως η πληροφορία έρχεται από το εξωτερικό και αυτή είναι συνήθως αποσπασματική, παρόλα αυτά ο κλάδος της ανάπτυξης λογισμικού που έχει σχέση με επαγγέλματα υγείας είναι δυναμικός και με ανοδική πορεία. Αξίζει να αναφερθεί πως στην Αθήνα λειτουργούν ήδη δυο Μεταπτυχιακά προγράμματα σπουδών που σχετίζονται με την βιοπληροφορική και την βιοτεχνολογία. Το ένα διεξάγεται στο Ε.Κ.Π.Α και ονομάζεται Τεχνολογίες Πληροφορικής στην Ιατρική και την Βιολογία και το άλλο διεξάγεται στο Τ.Ε.Ι Αθήνας και ονομάζεται Προηγμένα Συστήματα και Μέθοδοι στην Βιοιατρική Τεχνολογία. Επίσης περιφερειακά Πανεπιστημιακά ιδρύματα οργανώνουν και διεξάγουν αντίστοιχα μεταπτυχιακά προγράμματα. Επίσης στα προπτυχιακά προγράμματα μηχανικών και πληροφορικής έχουν ενσωματωθεί μαθήματα σχετικά με την Βιοπληροφορική-Ιατρική πληροφορική. Με την εξέλιξη των σπουδών προβλέπεται να υπάρξει ακόμη περισσότερη έρευνα και παράγωγη τεχνολογίας.

Σε γενικές γραμμές το μέλλον του συγκεκριμένου κλάδου είναι ραγδαία αναπτυσσόμενο σε παγκόσμιο επίπεδο οι μεγαλύτερες εταιρείες που ασχολούνται με τον κλάδο είναι:

- **Nucleiscs:** Αυστραλιανή εταιρεία που παράγει λογισμικό σχετικό με την ιατρική και τις επιστήμες υγείας καθώς και λογισμικό σχετικό με την αποκωδικοποίηση της αλυσίδας του DNA
- **Progenus :** Βέλγικη εταιρεία με ειδίκευση στην ανάλυση του DNA και λογισμικό σχετικό με την μοριακή βιολογία.
- **Algonomics:** Βέλγικη εταιρεία βιοτεχνολογίας-βιοπληροφορικής με ειδίκευση στην ανοσοθεραπεία δηλαδή ανάπτυξη θεραπειών με αντισώματα.
- **Bayer Bioscience:** Βέλγικη εταιρεία που ασχολείται κυρίως με βιοτεχνολογικά επιτεύγματα σε καλλιέργειες λαχανικών και σπαρτών. Επίσης προσφέρει μια σειρά γενετικά βελτιωμένων σπόρων για διάφορες καλλιέργειες
- **Caprion Proteomics:** Καναδική εταιρεία με κύρια ασχολία την ανάπτυξη λογισμικού αλλά και βιοτεχνολογίας για την μελέτη του ανθρώπινου γονιωματός και της πρωτεϊνικής σύνθεσης.
- **Zymeworks :** Καναδική εταιρεία που συνδυάζει την έρευνα για την ψηφιακή εξομοίωση με την υπολογιστική έρευνα για να ανάπτυξη θεραπευτικά ένζυμα και πρωτεΐνες. Επίσης ειδικεύεται στον σχεδιασμό και την μηχανική βίο καταλυτών και θεραπευτικών πρωτεϊνών.

Μελλοντικές εξελίξεις στον κλάδο της Βιοπληροφορικής

Αν και η πρόβλεψη εξελίξεων σε ένα επιστημονικό κλάδο είναι ένα παράτολμο εγχείρημα η εξέλιξη του κλάδου της πληροφορικής εξαρτάται σε μεγάλο βαθμό από τις δυο επιστήμες που σε συνδυασμό παράγουν και το όνομα της την βιολογία και την πληροφορική. Ένας τομέας στον οποίο αυτή την στιγμή διεξάγεται έρευνα για την παραγωγή εφαρμογών είναι η προσωποποιημένη ανάλυση του γονιδιωματός για την εξατομικευμένη παραγωγή ιατρικών συμπερασμάτων. Επίσης υπάρχει αρκετή κινητικότητα στην διαδικασία αποθήκευσης και ανάλυσης δεδομένων που εξάγονται από βιολογικούς παράγοντες η αποθήκευσή τους σε αποθήκες δεδομένων και η εξόρυξη συμπερασμάτων σχετικά με μια σειρά θεμάτων που απασχολούν την βιολογία για παράδειγμα η δημιουργία φαρμάκων, ή αντιμετώπιση ασθενειών. Η συνεχιζόμενη ανάπτυξη του κλάδου οφείλεται σε μεγάλο βαθμό στην ραγδαία ανάπτυξη του κλάδου της πληροφορικής τα τελευταία χρόνια, το διαδίκτυο, η μαζικοποίηση των προϊόντων

πληροφορικής σχεδόν σε κάθε σπίτι βοήθησε να ανοίξει το εύρος των προϊόντων που θα έβρισκαν εφαρμογή με μεγαλύτερη άνεση και σε λιγότερο χρόνο για τον κάθε άνθρωπο. Αλλά και οι πιο επαγγελματικές λύσεις οδήγησαν την διάγνωση και την θεραπεία σε νέα επίπεδα.

Αλλά και στην βιολογία η προσπάθεια να δημιουργηθεί ένα λεξικό πρωτεϊνών αλλά και η σχέση μεταξύ τους και έτσι να προχωρήσει και η διαδικασία μοντελοποίησης

Χαρακτηριστικό παράδειγμα χαρτογράφησης της δομής γονιδίων είναι η καταπολέμηση του καρκίνου άλλο παράδειγμα η καταπολέμηση του σταφυλόκοκκου ή άλλων μολυσματικών ασθενειών ανιχνεύοντας μικρές παραλλαγές στην ακολουθία του DNA.

Χαρακτηριστικά παραδείγματα όπου η πρωτοπόρα έρευνα στην βιολογία συνδυάζεται με την πληροφορική είναι ένα Project με την ονομασία BARCODE OF LIFE INITIATIVE.

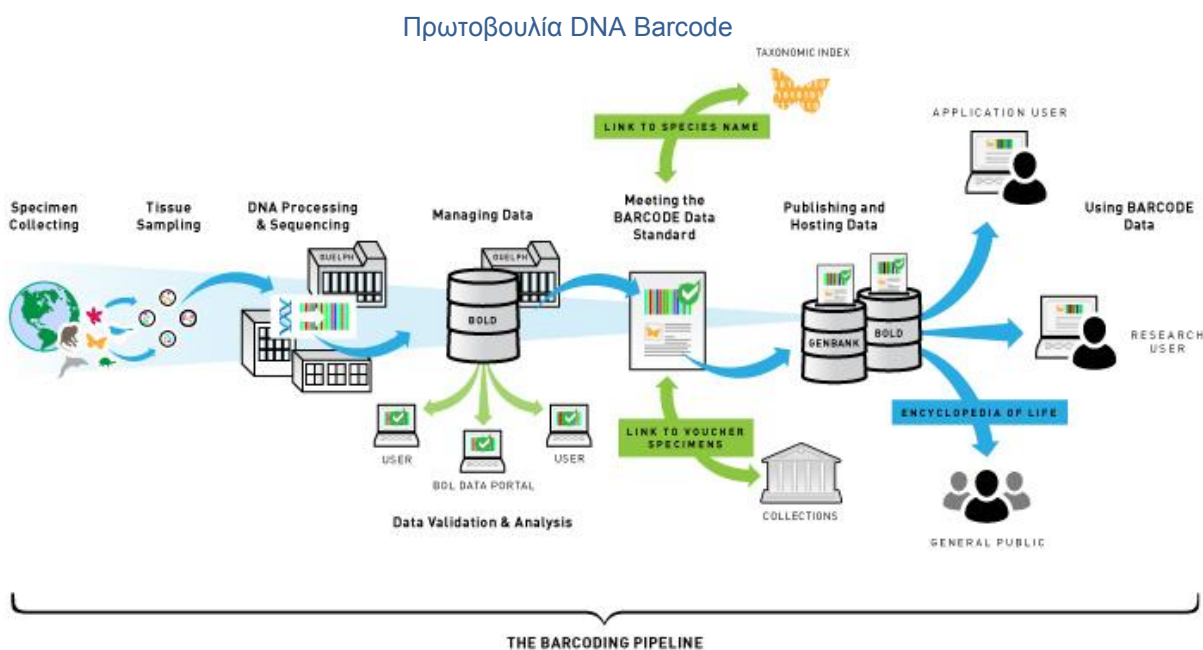


Figure 1 Από την ιστοσελίδα της πρωτοβουλίας

Σύμφωνα με την ιστοσελίδα της πρωτοβουλίας το 2003 ένα Καναδός ερευνητής ο Paul Hebert πρότεινε μια μέθοδο ταυτοποίησης των ειδών. Η ιδέα πίσω από αυτό είναι ότι θα χρησιμοποιείται μια μικρή γενετική ακολουθία από ένα προκαθορισμένο τμήμα του γονιδιώματος με τον ίδιο τρόπο που σε ένα πολυκατάστημα ταυτοποιούνται τα προϊόντα από το barcode. Ο τρόπος που γίνεται η ταυτοποίηση μέχρι σήμερα είναι βάσει μορφολογικών χαρακτηριστικών. Αυτό προϋποθέτει ότι το δείγμα προς ταξινόμηση θα πρέπει να έχει πλήρως ανεπτυγμένα χαρακτηριστικά έτσι ώστε να μπορεί να γίνει η ταξινόμηση, σε περίπτωση που κάτι τέτοιο δεν ισχύει η ταξινόμηση δεν μπορεί να επιτευχθεί ή μπορεί να είναι μια πολύ δύσκολη διαδικασία. Αν και αυτή η μέθοδος παραμένει η κυρίαρχη η χρήση του barcoding μπορεί να λειτουργήσει επικουρικά όπως επίσης να βοηθήσει ανθρώπους που δεν διαθέτουν ταξινομητικές γνώσεις να διακρίνουν τα είδη.

Το μέρος του γονιδίου που χρησιμοποιείται σαν το χαρακτηριστικό είναι η περιοχή 648 του γονιδίου **Cytochrome c oxidase I (COX1)** το οποίο είναι πάρα πολύ αποτελεσματικό στην ταυτοποίηση ζωικών ομάδων. Για την ταυτοποίηση φυτικών ομάδων θα χρησιμοποιείται μια ομάδα από άλλα γονίδια.

Το εγχείρημα του barcoding έχει 4 βασικούς άξονες

- Την συλλογή των δειγμάτων από μουσεία φυσικής ιστορίας, ζωολογικούς κήπους, αποθετήρια βιολογικών υλικών.

- Την εργαστηριακή ανάλυση ώστε να μπορέσει να ανακτηθεί υλικό για την δημιουργία του barcode στην συνέχεια αυτό το υλικό αποθηκεύεται σε βάσεις δεδομένων.
- Την βάση δεδομένων η οποία αποτελεί ένα από τα βασικά κομμάτια της πρωτοβουλίας εφόσον εκεί θα καταγράφονται τα στοιχεία ώστε να είναι διαθέσιμα δημόσια.
- Την ανάλυση δεδομένων η οποία επιτυγχάνεται μέσω του Barcode of Life Data Portal στο οποίο μπορούν να μπουν οι ερευνητές και να αποθηκεύσουν να αναλύσουν και να διαχειριστούν τα δεδομένα τους.

Οι βάσεις δεδομένων που χρησιμοποιεί η πρωτοβουλία είναι

- International Nucleotide Sequence Database Collaboration (INSDC) η οποία είναι μια συλλογική προσπάθεια για την καταγραφή και αποθήκευση ακολουθιών DNA και RNA. Την INSDC την απαρτίζουν 3 επιμέρους βάσεις η DNA Data Bank of Japan, GenBank (USA), European Nucleotide Archive (UK). Κάθε μέρα τα νέα δεδομένα που καταχωρούνται στις 3 επιμέρους βάσεις συγχρονίζονται στην INSDC.
- Barcode of life Database (BOLD) η οποία είναι μια υλοποίηση του Πανεπιστημίου Guelph του Οντάριο. Παρέχει στους χρήστες μια πλατφόρμα για ανάλυση ακολουθιών DNA. Το 2013 η BOLD περιελάμβανε 2,5 εκατομμύρια ακολουθίες από 190,000 είδη.

To portal Barcode of Life

Ένα από τα πιο ενδιαφέροντα κομμάτια της όλης προσπάθειας είναι η δημιουργία της πύλης Barcode of life στην οποία γίνεται στην ουσία ο συντονισμός όλων των δεδομένων που εισάγονται.

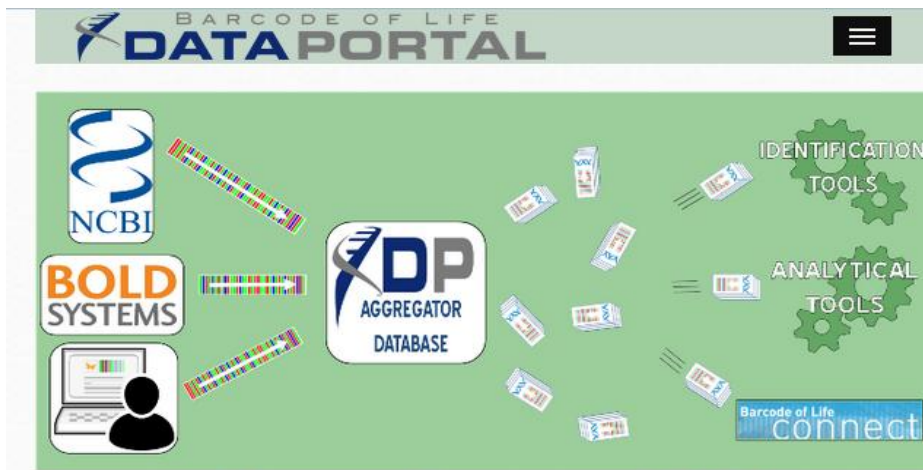


Figure 2 Η ροή εργασίας του Barcode of Life Data Portal

Αφού γίνει η εισαγωγή των δεδομένων από τις παραπάνω βάσεις δεδομένων αλλά και από μεμονωμένα μέλη της πρωτοβουλίας γίνεται αρχειοθέτηση από την Aggregator Database ώστε να γίνεται ευκολότερη η αναζήτηση.

Στην συνέχεια οι ομάδες δεδομένων που έχουν δημιουργηθεί μπορούν να χρησιμοποιηθούν από τους ερευνητές με τα εργαλεία που τους προσφέρει η πύλη.

Τα εργαλεία που προσφέρει η πύλη είναι τα εξής

1. Εργαλεία ταυτοποίησης με τα οποία οι server της πύλης συγκρίνουν μια ακολουθία ή μια ομάδα ακολουθιών με ήδη καταχωρημένες ακολουθίες και επιστρέφουν το βαθμό στον οποίο υπάρχει ταυτοποίηση.
2. Εργαλεία ανάλυσης με τα οποία γίνονται μετρήσεις όπως η γεωγραφική κατανομή.
3. Barcode of Life Connect network το οποίο αποτελεί ένα δίκτυο στο οποίο οι επιστήμονες μπορούν να δημοσιεύουν και να μοιράζονται πληροφορίες σχετικά με την ερευνητική τους δραστηριότητα.

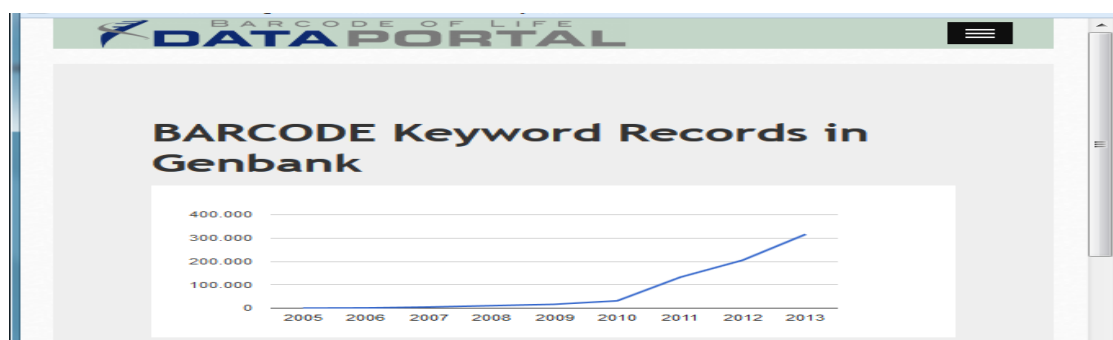


Figure 3 Αρχείο εγγραφών στην GenBank

Η πρωτοβουλία Barcode of Life έχει σαν σκοπό την ελεύθερη δωρεάν διάθεση των ευρημάτων της έρευνας που διεξάγεται. Οι ερευνητές που συμμετέχουν σε αυτή την πρωτοβουλία προέρχονται από διαφορετικά ερευνητικά κέντρα ή πρωτοβουλίες. Οι μεγαλύτερες ομάδες και πρωτοβουλίες που συμμετέχουν είναι οι εξής.

- iBOL International Barcode of life Project η οποία αποτελείται από ερευνητές από 25 χώρες σκοπός του είναι η δημιουργία 5 εκατομμυρίων καταγραφών DNA από 500,000 είδη σε 5 χρόνια για τον σκοπό αυτό έχουν δημιουργηθεί 10 ομάδες εργασίας.
- CBOL Consortium for the Barcode of Life η οποία προάγει την διαδικασία του barcoding μέσω συνεδρίων, θεματικών ομάδων και διάφορων δραστηριοτήτων.
- ECBOL European Consortium for the Barcode of Life η οποία ασχολείται με την ταξινόμηση.

Τα εργαστήρια CCDB Canadian Centre for DNA Barcoding του Πανεπιστημίου του Guelph είναι το μεγαλύτερο κέντρο έρευνας και τροφοδοτεί χιλιάδες αρχεία κάθε χρόνο, επίσης εκπαιδεύει ερευνητές στην διαδικασία του barcoding.

Τέλος η πρωτοβουλία διοργανώνει projects σε τοπικά ερευνητικά κέντρα με σκοπό την συλλογή ακόμα περισσότερες πληροφορίες και στοιχεία σχετικά με την διαδικασία του barcoding.

Human Genome Project

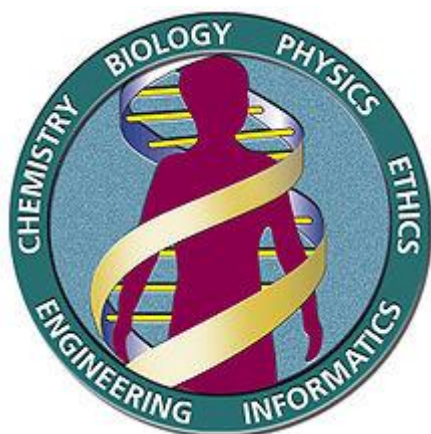


Figure 4 Το λογότυπο του Human Genome Project

Το **Human Genome Project (HGP)** είναι ένα διεθνές ερευνητικό πρόγραμμα με σκοπό να καθορίσει την ακολουθία των χημικών δεσμών οι οποίοι διαμορφώνουν το ανθρώπινο DNA και την ταυτοποίηση και χαρτογράφηση όλων των γονιδίων του ανθρώπου.

Αν και ολοκληρώθηκε παραμένει το μεγαλύτερο συνεργατικό εγχείρημα με την συμμετοχή μια πλειάδας επιστημόνων από διάφορους κλάδους συμπεριλαμβανομένης της πληροφορικής ενώ η ανάλυση των αποτελεσμάτων του θα συνεχιστεί για αρκετά χρόνια ακόμα .

Το HGP ασχολήθηκε με τα παρακάτω θέματα

- Η αναγνώριση όλων των περίπου 20,500 γονιδίων του ανθρώπινου DNA
- Η ανάλυση της ακολουθίας 3 δισεκατομμυρίων χημικών δεσμών που συνθέτουν το DNA
- Την αποθήκευση της πληροφορίας σε βάσεις δεδομένων
- Βελτίωση των εργαλείων για την ανάλυση δεδομένων
- Η μεταφορά της τεχνογνωσίας στον ιδιωτικό τομέα
- Ρύθμιση των ηθικών , κοινωνικών και νομικών θεμάτων που προκύπτουν από τα ευρήματα του HGP

Τα αποτελέσματα του HGP προέκυψαν και από αναλύσεις σε αρκετούς μη ανθρώπινους οργανισμούς όπως βακτήρια και ζωικοί οργανισμοί. Σημαντική είναι και η προσφορά του σε θέματα προώθησης μιας ηθικής αντιμετώπισης των ζητημάτων που άπτονται της χρήσης του ανθρώπινου DNA. Σημαντική συνεισφορά είναι η μεταφορά της επιστημονικής εμπειρίας σε

τρίτους ερευνητές και ερευνητικά κέντρα συνέβαλε καταλυτικά στην ιδιαίτερα κερδοφόρα βιομηχανία βιοτεχνολογίας των Ηνωμένων Πολιτειών

Η ιστορία του HGP

Το HGP προέκυψε το 1990 από την συμφωνία δυο μεγάλων οργανισμών του Εθνικού Ινστιτούτου για την Υγεία και του Υπουργείου Ενέργειας των Ηνωμένων Πολιτειών.

Το project στοίχισε 3 δισεκατομμύρια δολάρια και είχε ένα πλάνο για 15 χρόνια. Εκτός από τα ερευνητικά κέντρα των Η.Π.Α συμμετείχαν διεθνείς οργανισμοί και ερευνητές. Λόγω τις συμμετοχής πολλών αξιόλογων κέντρων αλλά και τις ανάπτυξης ιδιαίτερα της πληροφορικής η πρώτη «πρόχειρη» απεικόνιση του γονιδίου τελείωσε το 2000. Αυτή η πρώτη απεικόνιση ολοκληρώθηκε από τον ερευνητή Βιοπληροφορικής Jim Kent του Πανεπιστήμιου της Καλιφόρνια. Στην συνέχεια το project ολοκληρώθηκε το 2003 δυο χρόνια νωρίτερα από την αρχική πρόβλεψη και το Μάιο του 2006 ανακοινώθηκε η ανάλυση της ακολουθίας του τελευταίου χρωμοσώματος.

Το project δεν είχε σκοπό όλη την ακολουθία του DNA στο ανθρώπινο κύτταρο αλλά την ευχρωματίνη. Το DNA ενώνεται με πρωτεΐνες σχηματίζοντας μια ένωση που ονομάζεται χρωματίνη η ευχρωματίνη είναι η κατάσταση της χρωματίνης που περιέχει την γενετική πληροφορία αφού θεωρείται περιοχή ενεργών γονιδίων. Κατά την ολοκλήρωση της πρωταρχικής έρευνας του HGP είχε «αποκρυπτογραφηθεί» το 99,99% της ευχρωματικής.

Αναλύσεις και μελέτες πάνω στην ευχρωματική συνεχίζουν μέχρι σήμερα.

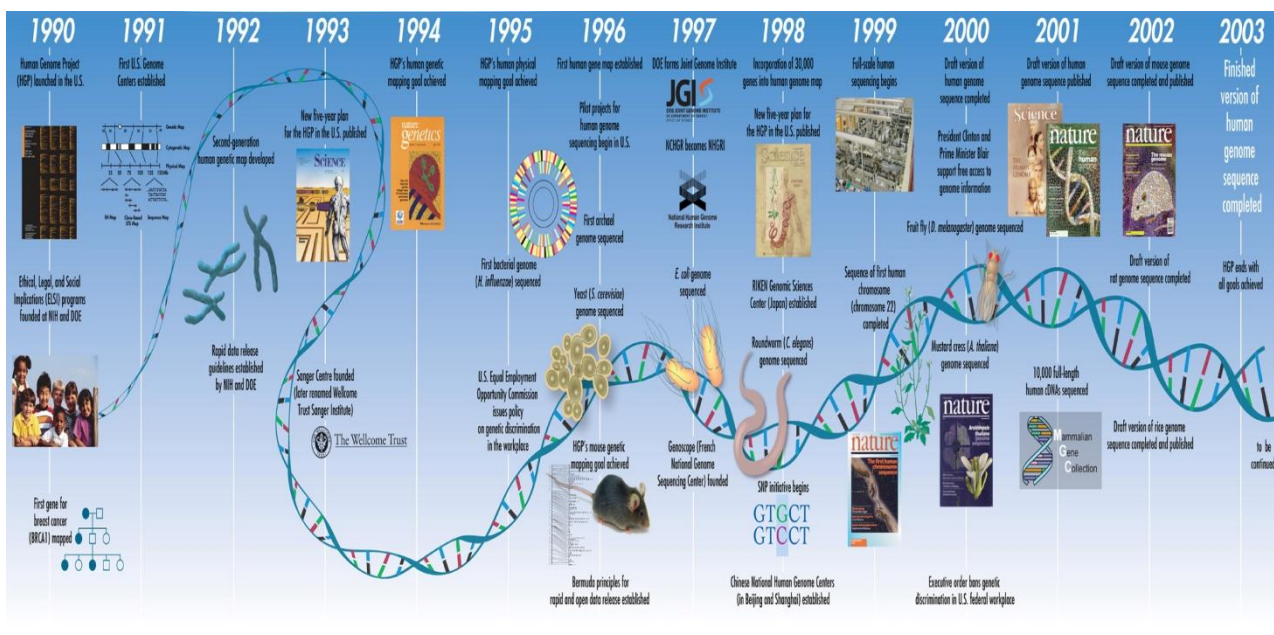


Figure 5 Το χρονοδιάγραμμα του HGP

Η έρευνα του HGP στην Βιοπληροφορική

Ένα από τα βασικά ερευνητικά πεδία του HGP είναι η Βιοπληροφορική χωρίς την καθοριστική συμβολή της οι πληροφορίες που συλλέχθηκαν δεν θα μπορούσαν να αξιοποιηθούν στον βαθμό που έχουν αξιοποιηθεί σήμερα.

Το HGP συνεχίζει να επενδύει σε βάσεις δεδομένων και εργαλεία για την ανάλυση των πληροφοριών. Οι βάσεις δεδομένων πρέπει να προσαρμόζονται στις όλο και μεγαλύτερες ανάγκες της επιστημονικής κοινότητας. Η ιδέα που υπάρχει για τις βάσεις δεδομένων είναι να αντιγράψουν τον τρόπο λειτουργίας των βάσεων δεδομένων των οργανισμών που συνδέονται με τις φαινοτυπικές πληροφορίες. Ως φαινότυπος ορίζεται τα φυσικά σε συνδυασμό με τα βιοχημικά χαρακτηριστικά που προκύπτουν από την αλληλεπίδραση του γονότυπου και του

περιβάλλοντος. Έτσι δύο οργανισμοί που διαθέτουν ίδιους γονότυπους δεν σημαίνει ότι θα μοιάζουν ή ότι θα δρουν με τον ίδιο τρόπο επειδή ο χαρακτήρας και η εμφάνιση καθορίζεται από περιβαλλοντικές συνθήκες.

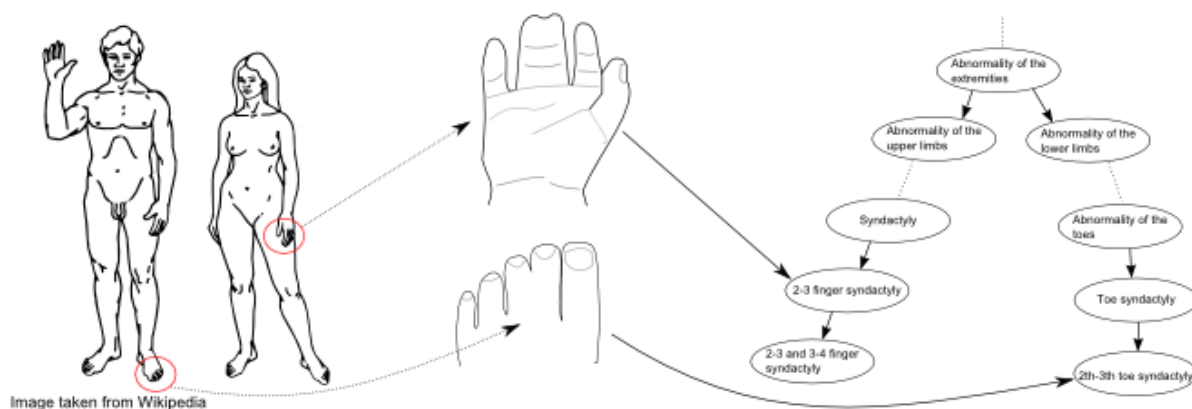


Figure 6 Φαινοτυπικά χαρακτηριστικά

Επίσης ένα άλλο ερευνητικό πεδίο είναι η ανάπτυξη βάσεων δεδομένων και αναλυτικών εργαλείων για την μοντελοποίηση σύνθετων βιολογικών δικτύων.

Πιο συγκεκριμένα οι στόχοι του HGP ήταν και παραμένει

- Η Βελτίωση του περιεχομένου και της λειτουργίας των βάσεων δεδομένων
- Ανάπτυξη καλύτερων εργαλείων για συλλογή και επεξεργασία δεδομένων
- Ανάπτυξη καλύτερων εργαλείων για αναπαράσταση και ανάλυση ακολουθιών
- Ανάπτυξη πρωτοπόρων λογισμικών

Στην ιστοσελίδα του National Center for Biotechnology Information υπάρχει το Human Genome Resources στο οποίο συγκεντρώνονται και διαχειρίζονται στοιχεία από μια σειρά πληροφοριών.

Τα αποτελέσματα από την έρευνα του HGP αν και είναι ακόμα υπό έρευνα αναμένεται να δώσουν πληροφορίες που θα φανούν χρήσιμες στην φαρμακευτική και βιοτεχνολογία. Ήδη σημαντικές πληροφορίες έχουν συλλεχθεί σχετικά με τον καρκίνο του μαστού, τις αιμοστατικές διαταραχές, την κυστική ίνωση. Η βάση δεδομένων που έχει δημιουργηθεί προσφέρει στους επιστήμονες ένα παράθυρο στον ερευνητικό κόσμο αφού μπορεί να διαβάσει τα συμπεράσματα από την έρευνα άλλων επιστημόνων συμπεριλαμβανομένου και του τρισδιάστατου μοντέλου του γονιδίου, της λειτουργίας του, της αλληλεπίδρασης του με άλλα γονίδια. Επίσης η ανάλυση των ομοιοτήτων και διαφορών μεταξύ της ακολουθίας του DNA διαφορετικών οργανισμών μπορεί να βοηθήσει στην εξελικτική έρευνα. Πολλές ερωτήσεις μπορούν να απαντηθούν σχετικά με της ομοιότητες και τις διαφορές για την εξέλιξη του ανθρώπου και των «συγγενών» του.

Τέλος το HGP ενέπνευσε την έρευνα και σε άλλα πεδία όπως την γεωργία και την γεωργική παραγωγή.

Αλλά και στον τομέα της βιοηθικής το HGP αποτέλεσε πρωτοπόρο ερευνητικό εγχείρημα αφού τελικά προέκυψε το Ethical, Legal, and Social Implications πρόγραμμα το οποίο ασχολείται με τα ηθικά και νόμιμα όρια των πληροφοριών από την ανάλυση του ανθρώπινου γονιδιόματος και την χρήση τους.

Τρέχοντα προγράμματα Βιοπληροφορικής

Επιγραμματικά γίνεται μια αναφορά σε τρέχοντα προγράμματα βιοπληροφορικής

- 1000 Genomes Project (<http://www.1000genomes.org/>)
- Chimpanzee genome project (<https://www.hgsc.bcm.edu/non-human-primates/chimpanzee-genome-project>)
- ENCODE (<https://www.encodeproject.org/>)
- Human Brain Project (<http://www.braininitiative.nih.gov/index.htm>)

- Human Connectome Project (<http://www.neuroscienceblueprint.nih.gov/connectome/>)
- Human Proteome Project (<http://www.thehpp.org/>)

Βιβλιογραφία

- Εισαγωγή στη JAVA , Λιακέας Γιώργος
- Java, προχωρημένες τεχνικές, Κερκίρη Τάνια Α
- Βάσεις και αποθήκες δεδομένων, σημειώσεις Γιάννης Θεοδωρίδης
- www.genome.gov
- www.barcodeoflife.org
- SQL Tutorial - W3Schools
- Βιοπληροφορική, Κοσσιδά Σοφία
- Ιατρική Πληροφορική – Medical Informatics, Google Scholar

ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα παρατίθεται το σύνολο του κώδικα JAVA του ανωτέρω λογισμικού

Extra.java

```
package med_app;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.lang.Character;

import javax.swing.JList;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

public class Extra implements ActionListener {

    JFrame frame, frame1;

    JTextField
    textbox, textbox2, textbox3, textbox4, textbox5, textbox6, textbox7, textbox8, textbox9, textbox10, text
    box11, textbox12;

    JLabel label, label2, label3, label4, label5, label6, label7, label8, label9, label10, label11, label12;

    JButton button;

    JPanel panel;

    static JTable table;

    static String
    ac_in, ac_in2, ac_in3, ac_in4, ac_in5, ac_in6, ac_in7, ac_in8, ac_in9, ac_in10, ac_in11, ac_in12, text;
```

```
        public static void connection(){
            try {
                Class.forName("com.mysql.jdbc.Driver");
                System.out.println("Driver loaded");
            } catch (ClassNotFoundException e) {
                System.out.println("Connection failed");
                e.printStackTrace();
            }
        }

    }

    public void searchgui(){
        frame = new JFrame("INSERT RECORDS IN DATABASE");
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        frame.setLayout(null);
        textbox = new JTextField();
        textbox.setBounds(85,100,150,20);
        label = new JLabel("PLEASE PROVIDE CODE FOR THE ACTIVE INGREDIENT");
        label.setBounds(85, 22, 350, 120);
        button = new JButton("Insert");
        button.setBounds(385,660,150,20);
        button.addActionListener(this);
        label2 = new JLabel("PLEASE PROVIDE CATEGORY FOR THE ACTIVE INGREDIENT");
        label2.setBounds(85, 155,350, 120);
        textbox2 = new JTextField();
        textbox2.setBounds(85,230,150,20);
        label3 = new JLabel("PLEASE PROVIDE NAME FOR THE ACTIVE INGREDIENT");
        label3.setBounds(85, 255, 350, 120);
        textbox3 = new JTextField();
        textbox3.setBounds(85,330,150,20);
        label4 = new JLabel("PLEASE PROVIDE DISEASE CODE");
        label4.setBounds(525,67,220,10);
    }
}
```

```
textbox4 = new JTextField();
textbox4.setBounds(525,85,150,20);
label5 = new JLabel("PLEASE PROVIDE THE NAME OF THE DISEASE");
label5.setBounds(525, 107, 350, 120);
textbox5 = new JTextField();
textbox5.setBounds(525,200,150,20);
label8 = new JLabel("PLEASE PROVIDE THE MEDICINE DOSAGE");
label8.setBounds(85, 357, 350, 120);
textbox8 = new JTextField();
textbox8.setBounds(85,440,150,20);
label9 = new JLabel("PLEASE PROVIDE THE METHOD OF ADMINISTRATOR");
label9.setBounds(525, 276, 290, 20);
textbox9 = new JTextField();
textbox9.setBounds(525,300,150,20);
label7 = new JLabel("PLEASE PROVIDE THE TRADENAME");
label7.setBounds(525, 396, 290, 20);
textbox7 = new JTextField();
textbox7.setBounds(525,420,150,20);
label6 = new JLabel("PLEASE PROVIDE THE CODE OF MEDICINE THAT CREATES
COMPLICATION");
label6.setBounds(525, 496, 350, 20);
textbox6 = new JTextField();
textbox6.setBounds(525,520,150,20);
label10 = new JLabel("PLEASE PROVIDE THE COMPLICATION");
label10.setBounds(85, 496, 290, 20);
textbox10 = new JTextField();
textbox10.setBounds(85,520,150,20);
label11 = new JLabel("PLEASE PROVIDE THE CONTRAINDICATIONS");
label11.setBounds(525, 596, 290, 20);
textbox11 = new JTextField();
textbox11.setBounds(525,620,150,20);
label12 = new JLabel("PLEASE PROVIDE THE CODE OF TRADENAME");
label12.setBounds(85, 576, 290, 20);
```

```
textbox12 = new JTextField();  
textbox12.setBounds(85,620,150,20);
```

```
frame.add(textbox);  
frame.add(textbox2);  
frame.add(textbox3);  
frame.add(textbox4);  
frame.add(textbox5);  
frame.add(textbox7);  
frame.add(textbox8);  
frame.add(textbox9);  
frame.add(textbox6);  
frame.add(textbox10);  
frame.add(textbox11);  
frame.add(textbox12);  
frame.add(label);  
frame.add(label2);  
frame.add(label3);  
frame.add(label4);  
frame.add(label5);  
frame.add(label5);  
frame.add(label7);  
frame.add(label8);  
frame.add(label6);  
frame.add(label9);  
frame.add(label10);  
frame.add(label11);  
frame.add(label12);  
frame.add(button);  
frame.setVisible(true);  
frame.setSize(1000, 1000);
```

```
}  
public void actionPerformed(ActionEvent ae)  
{  
    button = (JButton)ae.getSource();  
    System.out.println("Showing Table Data.....");  
    ac_in = textbox.getText();  
    ac_in = ac_in.replaceAll("[^a-zA-Z0-9]", "");  
    if (textbox.getText().trim().length() == 0){  
        JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 1", "ERROR", 0,  
        null);  
    }  
    ac_in2 = textbox2.getText();  
    ac_in2 = ac_in2.replaceAll("[^a-zA-Z]+", "");  
    if (textbox2.getText().trim().length() == 0){  
        JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 2", "ERROR", 0,  
        null);  
    }  
    ac_in3 = textbox3.getText();  
    ac_in3 = ac_in3.replaceAll("[^a-zA-Z0-9]", "");  
    if (textbox3.getText().trim().length() == 0){  
        JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 3", "ERROR", 0,  
        null);  
    }  
    ac_in4 = textbox4.getText();  
    ac_in4 = ac_in4.replaceAll("[^a-zA-Z0-9]", "");  
    if (textbox4.getText().trim().length() == 0){  
        JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 4", "ERROR", 0,  
        null);  
    }  
    ac_in5 = textbox5.getText();  
    ac_in5 = ac_in5.replaceAll("[^a-zA-Z0-9]", "");  
    if (textbox5.getText().trim().length() == 0){  
        JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 5", "ERROR", 0,  
        null);  
    }  
}
```

```
}  
ac_in6 = textbox6.getText();  
ac_in6 = ac_in6.replaceAll("[^a-zA-Z0-9]", "");  
if (textbox6.getText().trim().length() == 0){  
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 6", "ERROR", 0,  
null);  
}  
ac_in7 = textbox7.getText();  
ac_in7 = ac_in7.replaceAll("[^a-zA-Z0-9]", "");  
if (textbox7.getText().trim().length() == 0){  
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 7", "ERROR", 0,  
null);  
}  
ac_in8 = textbox8.getText();  
ac_in8 = ac_in8.replaceAll("[^a-zA-Z0-9]", "");  
if (textbox8.getText().trim().length() == 0){  
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 8", "ERROR", 0,  
null);  
}  
ac_in9 = textbox9.getText();  
ac_in9 = ac_in9.replaceAll("[^a-zA-Z]+", "");  
if (textbox9.getText().trim().length() == 0){  
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 9", "ERROR", 0,  
null);  
}  
ac_in10 = textbox10.getText();  
ac_in10 = ac_in10.replaceAll("[^a-zA-Z]+", "");  
if (textbox10.getText().trim().length() == 0){  
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 10", "ERROR", 0,  
null);  
}  
ac_in11 = textbox11.getText();  
ac_in11 = ac_in11.replaceAll("[^a-zA-Z]+", "");  
if (textbox11.getText().trim().length() == 0){
```

```
JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 11", "ERROR", 0,
null);
}
ac_in12 = textbox12.getText();
ac_in12 = ac_in12.replaceAll("[^a-zA-Z0-9]", "");
if (textbox12.getText().trim().length()== 0){
    JOptionPane.showMessageDialog( null, "PROVIDE DATA IN TEXTBOX 12", "ERROR", 0,
null);
}
//Integer.parseInt(ac_in);

ConnectwithMysql();
}
public static void ConnectwithMysql(){
    connection();
    String host = "jdbc:mysql://localhost/medicine?useOldAliasMetadataBehavior=true";
    String username = "root";
    String password = "";

    try {
        Connection connect = DriverManager.getConnection(host, username,
password) ;
        Statement statement = connect.createStatement();
        System.out.println("Connection established");

        //String ac_in = JOptionPane.showInputDialog("Provide the active ingredient
name",JOptionPane.CANCEL_OPTION);
        //if (ac_in == null)
            //System.exit(0);
        //else
        {
            try{
```



```
String sql = "INSERT INTO `ACTIVE INGREDIENT` (CODE,CATEGORY,NAME)
VALUES (" +ac_in+"," +ac_in2+"," +ac_in3+");"
```

```
String sql2 = "INSERT INTO `DISEASES` (DCODE,NAMES) VALUES
(" +ac_in4+"," +ac_in5+");"
```

```
String sql3 = "INSERT INTO `TRADENAMES` (TRCODE,Medicine_Name,`active
ingredient_CODE`) VALUES (" +ac_in12+"," +ac_in7+"," +ac_in+");"
```

```
String sql4 = "INSERT INTO `active ingredient_indicated_for_diseases` (`active
ingredient_CODE`,`DISEASES_DCODE`,`DOSAGE,PROVIDING) VALUES
(" +ac_in+"," +ac_in4+"," +ac_in8+"," +ac_in9+");"
```

```
String sql5 = "INSERT INTO `active ingredient_interacts_active ingredient` (`active
ingredient_CODE`,`active ingredient_CODE1`,`Complications) VALUES
(" +ac_in+"," +ac_in6+"," +ac_in10+");"
```

```
String sql6 = "INSERT INTO `active ingredient_not_indicated_for_diseases` (`active
ingredient_CODE`,`Complication) VALUES (" +ac_in+"," +ac_in11+");"
```

```
int result = statement.executeUpdate(sql);
```

```
int result2 = statement.executeUpdate(sql2);
```

```
int result3 = statement.executeUpdate(sql3);
```

```
int result4 = statement.executeUpdate(sql4);
```

```
int result5 = statement.executeUpdate(sql5);
```

```
int result6 = statement.executeUpdate(sql6);
```

```
JOptionPane.showMessageDialog(null,null,"DATABASE
UPDATED",JOptionPane.INFORMATION_MESSAGE);
```

```
}
```

```
catch (NullPointerException e){
```

```
JOptionPane.showMessageDialog( null, "PROVIDE VALID
ACTIVE INGRIDIENT", "ERROR", 0, null);
```

```
e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

```
catch (SQLException e) {
```

```
System.out.println("Connection to mysql Db failed");
```

```
        e.printStackTrace();
    }}

public static void main(String[] args)
{
    Extra m= new Extra();
    m.searchgui();

}
}
```

INTRO.JAVA

```
package med_app;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTable;
import javax.swing.JTextField;
import med_app.Main;
import med_app.Main2;
import med_app.Extra;

/**
 *
 * @author Kiriakos
```

```
*/  
  
public class Intro implements ActionListener {  
  
    JFrame frame, frame1;  
    JTextField textbox;  
    JLabel label,label2,label3;  
    JButton button,button2,button3;  
    JPanel panel;  
    static JTable table;  
    static String ac_in;  
    private String[] args;  
  
    public void searchgui(){  
        frame = new JFrame("CHOOSE ACTION");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setLayout(null);  
        label = new JLabel("PRESS FOR ACTIVE INGREDIENT SEARCH");  
        label2 = new JLabel("PRESS FOR ACTIVE INGREDIENT CONFLICTS");  
        label3 = new JLabel("PRESS FOR INSERTING DATA");  
        label.setBounds(50, 100, 280, 20);  
        label2.setBounds(50, 150, 290, 80);  
        label3.setBounds(50,45,270,10);  
        button = new JButton("Search");  
        button2 = new JButton("Conflicts");  
        button3 = new JButton("Insert");  
        button.setBounds(120,120,150,20);  
        button2.setBounds(120,200,150,20);  
        button3.setBounds(120,60,150,20);  
        button.addActionListener(this);  
        button2.addActionListener(this);  
        button3.addActionListener(this);
```

```
frame.add(label);
frame.add(button);
frame.add(label2);
frame.add(button2);
frame.add(label3);
frame.add(button3);
frame.setVisible(true);
frame.setSize(600,500);

}
public void actionPerformed(ActionEvent ae)
{
    Object src = ae.getSource();
    if(src == button){
        med_app.Main.main(args);
    }
    if(src == button2){
        med_app.Main2.main(args);
    }
    if(src == button3){
        med_app.Extra.main(args);
    }
}

public static void main(String[] args)
{

    Intro m= new Intro();
    m.searchgui();
```

```
    }  
}
```

MAIN.JAVA

```
import java.awt.Dimension;  
import java.awt.Font;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.ResultSetMetaData;  
import java.sql.SQLException;  
import java.sql.Statement;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
  
import javax.swing.JList;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JScrollPane;  
import javax.swing.JTable;  
import javax.swing.JTextField;  
  
public class Main implements ActionListener {  
    JFrame frame, frame1;  
    JTextField textbox;  
    JLabel label;  
    JButton button;  
    JPanel panel;  
    static JTable table;
```

```
static String ac_in;

        public static void connection(){
            try {
                Class.forName("com.mysql.jdbc.Driver");
                System.out.println("Driver loaded");
            } catch (ClassNotFoundException e) {
                System.out.println("Connection failed");
                e.printStackTrace();
            }
        }

public void searchgui(){
    frame = new JFrame("DATABASE SEARCH");
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    frame.setLayout(null);
    textbox = new JTextField();
    textbox.setBounds(120,100,150,20);
    label = new JLabel("PLEASE PROVIDE ACTIVE INGREDIENT");
    label.setBounds(85, 2, 250, 120);
    button = new JButton("Search");
    button.setBounds(120,130,150,20);
    button.addActionListener(this);

    frame.add(textbox);
    frame.add(label);
    frame.add(button);
    frame.setVisible(true);
    frame.setSize(500, 400);

}
```

```

public void actionPerformed(ActionEvent ae)
{
    button = (JButton)ae.getSource();
    System.out.println("Showing Table Data.....");
    ac_in = textbox.getText();
    ConnectwithMysql();
}

public static void ConnectwithMysql(){
    connection();
    String host = "jdbc:mysql://localhost/medicine?useOldAliasMetadataBehavior=true";
    String username = "root";
    String password = "";

    try {
        Connection connect = DriverManager.getConnection(host, username,
password) ;
        Statement statement = connect.createStatement();
        System.out.println("Connection established");

        //String ac_in = JOptionPane.showInputDialog("Provide the active ingredient
name",JOptionPane.CANCEL_OPTION);
        //if (ac_in == null)
            //System.exit(0);
        //else
        {
            try{
                String sql = "SELECT CATEGORY AS 'MEDICAL CATEGORY' ,diseases.NAMES
AS 'INDICATED FOR',complication AS 'COMPLICATIONS',GROUP_CONCAT(DISTINCT
Medicine_Name) AS `TRADENAMES`,DOSAGE AS 'TYPICAL DOSAGE',PROVIDING,CODE
FROM `active ingredient` JOIN `active ingredient_indicated_for_diseases` ON `active
ingredient`.CODE = `active ingredient_indicated_for_diseases`. `active ingredient_CODE` JOIN
DISEASES ON `active ingredient_indicated_for_diseases`.DISEASES_DCODE=
diseases.DCODE JOIN `active ingredient_not_indicated_for_diseases` ON `active
ingredient_not_indicated_for_diseases`. `active ingredient_CODE` = `active ingredient`.CODE
JOIN tradenames ON tradenames. `active ingredient_CODE` = `active ingredient`.CODE
WHERE `active ingredient`.NAME = ("+"ac_in+"");

                ResultSet result = statement.executeQuery(sql);
            }
        }
    }
}

```

```

        displayResults(result);
    }
    catch (NullPointerException e){
        JOptionPane.showMessageDialog( null, "PROVIDE VALID
ACTIVE INGRIDIENT", "ERROR", 0, null);
        e.printStackTrace();
    }
}
}
catch (SQLException e) {
    System.out.println("Connection to mysql Db failed");
    e.printStackTrace();
}}

```

```

private static void displayResults(ResultSet result) throws SQLException {
    ResultSetMetaData rmeta = result.getMetaData();
    int numColumns=result.getMetaData().getColumnCount();
    String[] a = new String [numColumns];
    String[] b = new String [numColumns];
    String[] c = new String [numColumns];
    while (result.next()){

        if (result.getString(1) != null) {
            for (int i=1;i<numColumns;i++)

                {

                    b[i]= rmeta.getColumnName(i);
                    a[i] = result.getString(i).trim();
                    c[i] = b[i]+ "-->" +a[i];
                }
        }
    }
}

```



```

        }

        JList list = new JList(c);

        JScrollPane scrollpane = new JScrollPane(list);

        JPanel panel = new JPanel();

        panel.add(scrollpane);

        scrollpane.getViewport().add(list);

        scrollpane.setPreferredSize(new Dimension(850,300));

        Font myFont = new Font("Arial", Font.BOLD,24);

        list.setFont(myFont);

        JOptionPane.showMessageDialog(null,scrollpane,"MEDICAL
INFORMATION", JOptionPane.INFORMATION_MESSAGE);

        System.out.println(numColumns);

    }else JOptionPane.showMessageDialog( null, "PROVIDE VALID ACTIVE
INGRIDIENT", "ERROR", 0, null);

    }

}

public static void main(String[] args)

    {

        Main m= new Main();

        m.searchgui();

    }

}

MAIN2.JAVA
import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JDesktopPane;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;

import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;

public class Main2 implements ActionListener {
    JFrame frame, frame1;
    JTextField textbox, textbox2;
    JLabel label, labe2;
    JButton button;
    JPanel panel;
    static JTable table;
    static String ac_in, ac_in2;
    static String sf;

    public static void connection(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
```

```
        System.out.println("Driver loaded");
    } catch (ClassNotFoundException e) {
        System.out.println("Connection failed");
        e.printStackTrace();
    }
}

}

public void searchgui(){
    frame = new JFrame("DATABASE SEARCH FOR CONFLICTS");
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    frame.setLayout(null);
    frame.setState(frame.NORMAL);
    frame.setVisible(true);
    textbox = new JTextField();
    textbox2 = new JTextField();
    textbox2.setBounds(100,190,150,20);
    textbox.setBounds(100,90,150,20);
    label = new JLabel("PLEASE PROVIDE THE FIRST MEDICINE");
    labe2 = new JLabel("PLEASE PROVIDE THE SECOND MEDICINE");
    label.setBounds(50, 50, 280, 20);
    labe2.setBounds(50, 110, 290, 80);
    button = new JButton("search");
    button.setBounds(120,280,150,20);
    button.addActionListener(this);

    frame.add(textbox);
    frame.add(textbox2);
    frame.add(label);
    frame.add(labe2);
    frame.add(button);
    frame.setVisible(true);
    frame.setSize(500, 400);
```

```

}

public void actionPerformed(ActionEvent ae)
{
    button = (JButton)ae.getSource();
    ac_in = textbox.getText();
    ac_in2 = textbox2.getText();
    System.out.println(ac_in);
    if ( ac_in == null ? "" == null : ac_in.equals("")&&ac_in2 == null ? "" == null : ac_in2.equals(""))
    {
        JOptionPane.showMessageDialog( null, "PROVIDE VALID ACTIVE INGRIDIENT",
        "ERROR", 0, null);

    }

    ConnectwithMysql();
}

public static void ConnectwithMysql(){
    connection();

    String host = "jdbc:mysql://localhost/medicine?useOldAliasMetadataBehavior=true";
    String username = "root";
    String password = "";

    try {
        Connection connect = DriverManager.getConnection(host, username,
password) ;

        Statement statement = connect.createStatement();
        System.out.println("Connection established");

        //String ac_in = JOptionPane.showInputDialog("Provide the active ingredient
name",JOptionPane.CANCEL_OPTION);

        //if (ac_in == null)
            //System.exit(0);

        //else
        {

```

```

        try{

            String sql = "SELECT DISTINCT Complications as COMPLICATIONS FROM` active
ingredient_interacts_active ingredient` INNER JOIN TRADENAMES AS TR1 ON` active
ingredient_interacts_active ingredient`.` active ingredient_CODE`=TR1.` active
ingredient_CODE` INNER JOIN TRADENAMES AS TR2 ON` active ingredient_interacts_active
ingredient`.` active ingredient_CODE1`=TR2.` active ingredient_CODE` WHERE
TR1.Medicine_Name=(``+ac_in+)``) AND TR2.Medicine_Name=(``+ac_in2+)``";

            ResultSet result = statement.executeQuery(sql);

            displayResults(result);

        }

        catch (NullPointerException e){

            JOptionPane.showMessageDialog( null, "PROVIDE VALID
ACTIVE INGRIDIENT", "ERROR", 0, null);

            e.printStackTrace();

        }

    }

}

catch (SQLException e) {

    System.out.println("Connection to mysql Db failed");

    e.printStackTrace();

}}

```

```

private static void displayResults(ResultSet result) throws SQLException {

    ResultSetMetaData rmeta = result.getMetaData();

    int numColumns=result.getMetaData().getColumnCount();

    String[] a = new String [1];

    String[] b = new String [1];

    String[] c = new String [1];

    String s1 = new String();

    String s2 = new String();

```

```
if (result.next()==true)
{
do

{
//if (result.next()==false) {/
//for(int i =1;i < numColumns; i++)
//{

s1 = rmeta.getColumnName(1);
s2 = result.getString(1);
sf = s1+"-->" +s2;

JFrame.setDefaultLookAndFeelDecorated(true);
JFrame frame = new JFrame();
frame.setTitle("CAUTION");
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
JLabel label = new JLabel(sf);
frame.add(label);
frame.pack();
frame.setSize(new Dimension(300, 250));
frame.getContentPane().setBackground(Color.RED);
frame.setVisible(true);
// }

} while (result.next()) ;

} else JOptionPane.showMessageDialog( null, "PROVIDE ANOTHER INGRIDIENT NO
ASSOCIATION FOUND", "WARNING", JOptionPane.WARNING_MESSAGE);
}
```

```
public static void main(String[] args)
    {
        Main2 m= new Main2();
        m.searchgui();
    }
}
```