

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον
παγκόσμιο ιστό»

«αριθμός άσκησης» 3	Τίτλος άσκησης ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Νικολέτα-Χαρά Βλάχου-Σακελλαρίου - Π19028
Ημερομηνία παράδοσης	8/9/2021



Εκφώνηση της άσκησης

Στην τελική εργασία του μαθήματος θα επεκτείνετε την 2η Άσκηση ώστε να ολοκληρώσετε την εφαρμογή τριών επιπέδων (3-tier), η οποία θα υλοποιεί όλες τις λειτουργίες (μεθόδους) που ορίσατε στην 1η Άσκηση (με τις πιθανές αλλαγές που έγιναν).

id	name	password	Τύπος
1	Mary123	123	Ασθενής
2	Nara90	9	Γιατρός
3	Sally2	2	Διαχειριστής
4	Kana1	1	Ασθενής

Πίνακας αναφοράς για κωδικούς μερικών χρηστών

Επειδή στη βάση είναι αποθηκευμένοι σε hashed+ ~~saleted~~ μορφή

Εικόνα 0 – Βοηθητική



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

0 Εικόνα-Βοηθητική	2
1 Γενική Περιγραφή της λύσης	4
2 Η βάση δεδομένων	4
2.1 Οι πίνακες της βάσης usersdb	4
2.1.1 users	4
2.1.2 patients	4
2.1.3 doctors	4
2.1.4 appointment	5
2.1.5 admins	5
Εικόνα 1 - EER diagram	5
3 Κώδικας προγράμματος	6
3.1 HTML & JSP	6
Εικόνα 2-Αρχική Σελίδα	6
Εικόνα 3- Φόρμα Login	6
3.1.2 patientLogged.jsp	7
Εικόνα 4-Σελίδα Ασθενών	7
3.1.3 DoctorLogged	7
Εικόνα 5-Σελίδα Γιατρών	7
Εικόνα 6 -Φόρμα Ακύρωσης	8
3.2 Τα Πακέτα του προγράμματος	8
3.2.1 mainpackage	8
3.2.1.1 Users	8
3.2.1.2 Patient	9
3.2.1.3 Doctor	9
3.2.1.4 Admin	10
3.2.1.5 Appointment	10
3.2.2 dynamic.com.web.service	10
3.2.2.1 DatabaseConnection	10
3.2.2.2 HashNSalt	10
3.2.2.3 ManagerServlet 1	1
3.2.2.4 ServletForPatient (Από Εργασία 2η,δεν χρησιμοποιείται).....	12
3.2.2.5 AdminServlet (Από Εργασία 2η,δεν χρησιμοποιείται)	12
3.2.2.6 Doctor Servlet(Από Εργασία 2η,δεν χρησιμοποιείται).....	12
3.2.3 dynamic.com.web.service.jsp.services	12
3.2.3.1 CreateAppointmentsServlet	12
3.2.3.2 CancellationServlet	12
3.2.3.3 ShowDoctorAvailibty	12
3.2.3.4 AddAppointment	13



3.2.4 RetrieveDataFromDB	13
3.2.4.1 GetUsers	13
3.2.4.2 <i>GetPatients</i>	13
3.2.4.3 GetDoctors	14
3.2.4.4 GetAppointments	14
3.2.4.4 GetAdmins	14
4 Βιβλιογραφία	14

1 Γενική Περιγραφή της λύσης

Το πρόγραμμα χρησιμοποιεί την βάση usersdb, tomcat server 10, servlets.jsp, και απλό κώδικα java ώστε να δημιουργηθούν μερικές σελίδες που στο σύνολο τους προσομοιώνουν ένα σύστημα υγείας. Για να το δείτε στο browser πάτε στο /DynamicWebProjext2021/src/main/webapp/index.html όπου βρίσκεται η κεντρική σελίδα

2 Η βάση δεδομένων

Στη βάση usersdb υπάρχουν οι πίνακες users, doctors, patients, admins και appointment.

2.1 Οι πίνακες της βάσης usersdb

2.1.1 users

Στον πίνακα users αποθηκεύονται τα στοιχεία όλων των χρηστών του προγράμματος(id, username, name, password, surname, type, salt) και κλειδί του είναι το id του κάθε χρήστη.

2.1.2 patients

Στον πίνακα users αποθηκεύονται τα στοιχεία όλων των ασθενών του συστήματος(user_id, AMKA, dateStart και dateEnd(για κάποιο ραντεβού), username, password) , κλειδί του είναι το AMKA και ξένο κλειδί το user_id.

2.1.3 doctors

Στον πίνακα users αποθηκεύονται τα στοιχεία όλων των γιατρών του συστήματος(user_id, admin_id, specialty και username) κλειδί του είναι το user_id και ξένο κλειδί το admin_id.

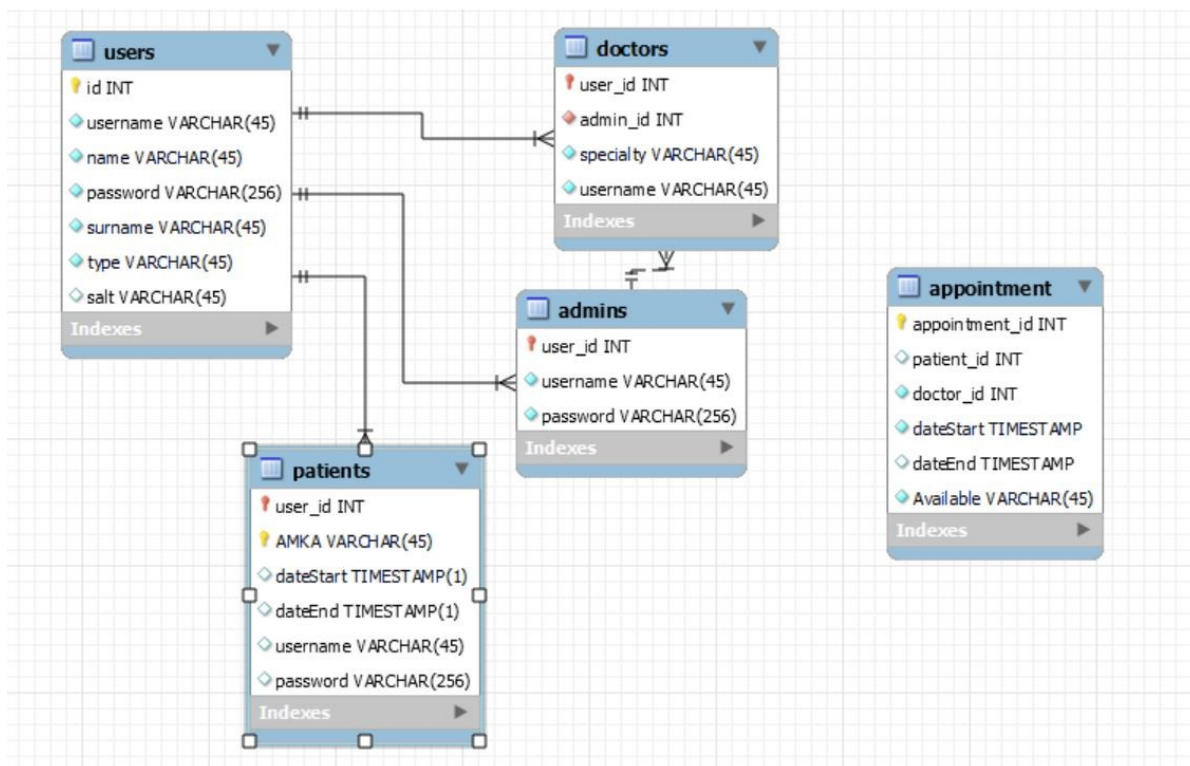


2.1.4 appointment

Στον πίνακα appointment αποθηκεύονται τα στοιχεία όλων των ραντεβού του συστήματος (appointment_id, **patient_id** (το οποίο είναι **null** εάν υπάρχει διαθέσιμο ραντεβού), doctor_id, dateStart και dateEnd, Available), κλειδί του είναι το appointment_id.

2.1.5 admins

Στον πίνακα admins αποθηκεύονται τα στοιχεία όλων των διαχειριστών του συστήματος (user_id, username, password), διαθέτει ξένο κλειδί το user_id.



Εικόνα 1 - EER diagram



3 Κώδικας προγράμματος

Ακολουθεί η αναλυτική περιγραφή του προγράμματος.

3.1 HTML & JSP

Η αρχική σελίδα του προγράμματος είναι η `index.html`. Περιέχει μια φόρμα εισαγωγής στοιχείων, από εκεί τα στοιχεία στέλνονται στο **ManagerServlet**. Όλες οι σελίδες έχουν μορφοποιηθεί από το αρχείο **style.css**.

Εικόνα 2-Αρχική Σελίδα



Login to our system!

User Name

Password

Submit

Στη συνέχεια η κατηγορία του χρήστη αναγνωρίζεται από το **ManagerServlet** και οδηγείται σε μια από τις σελίδες:

- **adminLogged.jsp**
- **patientLogged.jsp**
- **doctorLoggged.jsp**

Εικόνα 3- Φόρμα Login



3.1.2 patientLogged.jsp

Η σελίδα αυτή είναι για τους ασθενείς του συστήματος και μπορούν να δουν το ιστορικό τους, τη διαθεσιμότητα των ιατρών, να κανονίσουν ή και να ακυρώσουν ραντεβού.

Εικόνα 4-Σελίδα Ασθενών

Local Hospital

[Home](#) [Patients](#) [About Covid-19](#)

Mary123 with id 1 welcome back to the hospital system

User's appointments are:

2022-01-01 10:10:10 2022-06-01 14:10:10

Check doctor availability

The Cardiologist is available at: 2022-02-01 14:10:10 The Cardiologist is available at: 2022-03-01 14:10:10 The Dermatologist is available at: 2022-04-01 14:10:10 The Cardiologist is available at: 2022-05-01 14:10:10 The Dermatologist is available at: 2022-06-01 14:10:10 The Cardiologist is available at: 2022-07-01 14:10:10

3.1.3 DoctorLogged

Εικόνα 5-Σελίδα Γιατρών

Local Hospital

[Home](#) [Patients](#) [About Covid-19](#)

Doctor Nara90 welcome back to the hospital system

User's appointments are:

2022-01-01 10:10:10 2022-02-01 14:10:10 2022-03-01 14:10:10 2022-05-01 14:10:10 2022-07-01 14:10:10

Η σελίδα αυτή είναι για τους γιατρούς του συστήματος και μπορούν να κάνουν κατάθεση διαθεσιμότητας ραντεβού, να δουν τα ραντεβού τους και να ακυρώσουν ραντεβού.



Cancel date

Date Time(for example 2022-02-01 14:10:10)

Doctor

id

2

Cancel

Εικόνα 6 -Φόρμα Ακύρωσης

3.2 Τα Πακέτα του προγράμματος

Το πρόγραμμα διαθέτει τα πακέτα:

1. **mainpackage** με χαρακτηριστικά και λειτουργίες των χρηστών από την πρώτη εργασία
2. **dynamic.com.web.service** όπου βρίσκονται τα περισσότερα servlet και μερικές κλάσεις που εξυπηρετούν αυτά
3. **dynamic.com.web.service.jsp.services** που εκτελεί τις λειτουργίες των σελίδων (πχ ακύρωση ραντεβού)
4. **RetrieveDataFromDB** που επιστρέφει σε ArrayList όλους τους χρήστες μιας κατηγορίας που βρίσκονται στη βάση usersdb

3.2.1 mainpackage

3.2.1.1 Users

Η κλάση αυτή έχει τα χαρακτηριστικά κάθε χρήστη της εφαρμογής (id,username,name,password,surname,type) και ένα μετρητή των χρηστών της userCounter που αυξάνεται με την προσθήκη κάθε νέου χρήστη. Διαθέτει έναν public constructor ώστε να μπορεί να χρησιμοποιηθεί παντού στην εφαρμογή για την προσθήκη νέων χρηστών στο σύστημα. Τέλος, επειδή τα χαρακτηριστικά είναι private



υπάρχουν τυπικοί getters και setters για την αλλαγή και πρόσβαση σε αυτά αντιστοίχως.

3.2.1.2 Patient

Γενική Περιγραφή

Η κλάση αυτή έχει τα χαρακτηριστικά κάθε ασθενή της εφαρμογής (id, AMKA) και κληρονομεί όλα τα χαρακτηριστικά της κλάσης **Users**. Διαθέτει έναν public constructor ώστε να μπορεί να χρησιμοποιηθεί παντού στην εφαρμογή για την προσθήκη νέων ασθενών στο σύστημα. Επειδή τα χαρακτηριστικά είναι private υπάρχει getter για το AMKA.

Μέθοδοι

Show()

Διαθέτει την μέθοδο Show με όρισμα το id ενός ασθενή που επιστρέφει τα προγραμματισμένα ραντεβού του. Η μέθοδος αυτή συνδέεται με την βάση usersdb μέσω της κλάσης DatabaseConnection, επιλέγει τις ημερομηνίες που είναι στην ίδια γραμμή με το id του ασθενή, τις προθέτει επαναληπτικά στο StringBuilder sb και το επιστρέφει ως αποτέλεσμα. Τέλος εάν προκύψει exception επιστρέφει ανάλογο μήνυμα και γίνεται χειρισμός με try- catch blocks.

3.2.1.3 Doctor

Γενική Περιγραφή

Η κλάση αυτή έχει τα χαρακτηριστικά κάθε γιατρού της εφαρμογής (id, username, **specialty**, **admin_id**) και κληρονομεί όλα τα χαρακτηριστικά της κλάσης **Users**. Διαθέτει έναν public constructor ώστε να μπορεί να χρησιμοποιηθεί παντού στην εφαρμογή για την προσθήκη νέων γιατρών στο σύστημα. Επειδή τα χαρακτηριστικά είναι private υπάρχει getters για να είναι δυνατή η πρόσβαση σε αυτά.

Μέθοδοι

Show()

Διαθέτει την μέθοδο Show με όρισμα το id ενός γιατρού που επιστρέφει τα προγραμματισμένα ραντεβού του. Η μέθοδος αυτή συνδέεται με την βάση usersdb μέσω της κλάσης DatabaseConnection, επιλέγει τις ημερομηνίες που είναι στην ίδια γραμμή με το id του ασθενή, τις προθέτει επαναληπτικά στο StringBuilder sb και το επιστρέφει ως αποτέλεσμα. Τέλος εάν προκύψει exception επιστρέφει ανάλογο μήνυμα και γίνεται χειρισμός με try- catch blocks.



3.2.1.4 Admin

Γενική Περιγραφή

Η κλάση αυτή έχει τα χαρακτηριστικά κάθε διαχειρηστή της εφαρμογής (admin_id,username) και κληρονομεί όλα τα χαρακτηριστικά της κλάσης **Users** . Διαθέτει έναν public constructor ώστε να μπορεί να χρησιμοποιηθεί παντού στην εφαρμογή για την προσθήκη νέων ασθενών στο σύστημα. Επειδή τα χαρακτηριστικά είναι private υπάρχουν getters για να είναι δυνατή η πρόσβαση σε αυτά.

Μέθοδοι

add_new_doctor()

Δέχεται σαν παραμέτρους, τα id ,specialty,name ενός γιατρού , δημιουργεί ένα νέο αντικείμενο τύπου Doctor και το επιστρέφει ως αποτέλεσμα.

3.2.1.5 Appointment

Γενική Περιγραφή

Η κλάση αυτή έχει τα χαρακτηριστικά κάθε ραντεβού της εφαρμογής (appointment_id,patient_id,doctor_id,dateStart,Availability). Διαθέτει έναν public constructor ώστε να μπορεί να χρησιμοποιηθεί παντού στην εφαρμογή για την προσθήκη νέων ασθενών στο σύστημα. Επειδή τα χαρακτηριστικά είναι private υπάρχουν getters για να είναι δυνατή η πρόσβαση σε αυτά.

3.2.2 dynamic.com.web.service

3.2.2.1 DatabaseConnection

Η κλάση αυτή πραγματοποιεί την σύνδεση με την βάση.

Μέθοδοι

initializeDatabase()

Κλασικός τρόπος σύνδεσης με βάση, διαθέτει το url που χρειάζεται η σύνδεση Connection ως παράμετρο,το όνομα χρήστη ,τον κωδικό καθώς και το όνομα της βάσης. Τέλος επιστρέφει την σύνδεση **con** και κάνει χειρισμό εξαιρέσεων με το throws.

3.2.2.2 HashNSalt

Η κλάση αυτή χρησιμοποιείται για να μετατρέπει κωδικούς σε hashed+salted μορφή



Μέθοδοι

CheckForEquality()

Η μέθοδος δέχεται τις παραμέτρους `userPassword` για τον κωδικό που δίνει ο χρήστης της εφαρμογής, `salt` που είναι το `salt` που βρίσκεται αποθηκευμένο στη βάση για τον συγκεκριμένο χρήστη και `hashedNSaltedPassword` που είναι ο κωδικός σε κωδικοποιημένη μορφή στη βάση. Αρχίζει με την υπόθεση πως ο κωδικός που δόθηκε είναι λάθος. Στη συνέχεια μετατρέπει από `string` σε `byte` το `salt` της βάσης και στη συνέχεια κωδικοποιεί τον κωδικό που έδωσε ο χρήστης με τη μέθοδο **`getSecurePassword`**. Εάν το αποτέλεσμα είναι το ίδιο με τον κωδικό της βάσης η μέθοδος επιστρέφει ότι ταιριάζουν (`true`), αλλιώς επιστρέφει ότι ο παραπάνω ισχυρισμός είναι ψευδής (`false`).

GetSecurePassword()

Στόχος της είναι να επιστρέψει τον κωδικό που της δίνεται σε `hashed+salted` μορφή. Δέχεται τις παραμέτρους `password` και `salt`. Αρχικά χρησιμοποιείται ο αλγόριθμος `sha-256`, στον οποίο μπαίνει και το `salt` με τη μέθοδο `update()`. Στη συνέχεια η μέθοδος `digest()` δίνει τον κωδικό στη ζητούμενη μορφή. Με τη βοήθεια ενός επαναληπτικού βρόγχου εισάγεται σε έναν `StringBuilder` όπου τελικά μετατρέπεται σε `String` και επιστρέφεται ως αποτέλεσμα.

GetSalt()

Δημιουργεί έναν τυχαίο αριθμό καπó `bytes` και τον επιστρέφει .

3.2.2.3 ManagerServlet

Το `servlet` αυτό ελέγχει αν τα στοιχεία του χρήστη είναι σωστά και τον ανακατευθύνει στην σελίδα που ταιριάζει στην κατηγορία του.

Μέθοδοι

doGet()

Αρχικά γίνεται σύνδεση με την βάση με την μέθοδο `initializeDatabase()` της κλάσης `DatabaseConnection` και εμφανίζει κατάλληλο μήνυμα αν γίνει με επιτυχία. Στη συνέχεια , χρησιμοποιώντας την παράμετρο `request` εκχωρεί σε μεταβλητές τα στοιχεία που έδωσε ο χρήστης. Ελέγχει εάν υπάρχει στη βάση και έπειτα εάν ο κωδικός ταιριάζει, που διαπιστώνεται με τη μέθοδο `CheckForEquality` της κλάσης `HashNSalt`, βλέπει την κατηγορία του χρήστη από τη βάση και τον κατευθύνει στην ανάλογη σελίδα. Επίσης ,στην καινούρια σελίδα στέλνονται και τα `username,id` του χρήστη, ώστε να μπορέσουν να χρησιμοποιηθούν αργότερα. Τέλος, γίνεται χειρισμός εξαιρέσεων με `try-catch blocks`.



3.2.2.4 ServletForPatient (Από Εργασία 2η,δεν χρησιμοποιείται)

3.2.2.5 AdminServlet (Από Εργασία 2η,δεν χρησιμοποιείται)

3.2.2.6 Doctor Servlet(Από Εργασία 2η,δεν χρησιμοποιείται)

3.2.3 dynamic.com.web.service.jsp.services

3.2.3.1 CreateAppointmentsServlet

Το servlet αυτό δημιουργεί νέα ραντεβού για τους ασθενείς της εφαρμογής

Μέθοδοι

doGet()

Έχει τα κλασικά ορίσματα request,respond που το πρώτο μεταφέρει τα στοιχεία του ραντεβού που θα δημιουργηθεί από το patientLoggeg.jsp.

Αρχικά συνδέεται με τη βάση usersdb με τη βοήθεια της κλάσης

DatabaseConnection. Στη συνέχεια κάνει update τη στήλη που ταιριάζει με την ημερομηνία που διάλεξε ο χρήστης από τις διαθέσιμες. Επιστρέφει αν το ραντεβού κλείστηκε με επιτυχία σε νέα σελίδα .Τέλος κάνει χειρισμό εξαιρέσεων με try-catch blocks.

3.2.3.2 CancellationServlet

Το servlet αυτό ακυρώνει τα ραντεβού των ασθενών της εφαρμογής

Μέθοδοι

doGet()

Έχει τα κλασικά ορίσματα request,respond που το πρώτο μεταφέρει τα στοιχεία του ραντεβού που θα ακυρωθεί από το patientLoggeg.jsp.

Αρχικά συνδέεται με τη βάση usersdb με τη βοήθεια της κλάσης

DatabaseConnection. Στη συνέχεια κάνει update τη στήλη που ταιριάζει με την ημερομηνία που διάλεξε ο χρήστης για ακύρωση από τα κλεισμένα του ραντεβού.

Επιστρέφει αν το ραντεβού ακυρώθηκε με επιτυχία σε νέα σελίδα .Τέλος κάνει χειρισμό εξαιρέσεων με try-catch blocks.

3.2.3.3 ShowDoctorAvailabilty

Η κλάση αυτή δείχνει τη διαθεσιμότητα κάθε γιατρού της εφαρμογής

Μέθοδοι

Show()

Αρχικά όλοι οι γιατροί της εφαρμογής αποθηκεύονται σε ένα ArrayList τύπου Doctor με την βοήθεια της κλάσης **GetDoctors.** Στη συνέχεια συνδέεται με τη βάση usersdb με τη βοήθεια της κλάσης

DatabaseConnection. Επιλέγει όλα τα ραντεβού από τον πίνακα



appointment της βάσης usersdb, και βάζει επαναληπτικά σε ένα StringBuilder sb όσα είναι διαθέσιμα μαζί με το όνομα του αντίστοιχου γιατρού από το ArrayList<Doctor> . Στο τέλος επιστρέφει το sb, ή ανάλογο μήνυμα εάν υπάρξει κάποιο σφάλμα το οποίο χειρίζεται με try-catch blocks.

3.2.3.4 AddAppointment

Η κλάση αυτή προσθέτει νέα ραντεβού στο σύστημα (Δεν λειτουργεί, λόγω σφάλματος)

Μέθοδοι

doGet()

Έχει τα κλασικά ορίσματα request, respond που το πρώτο μεταφέρει τα στοιχεία του ραντεβού που θα προσθέτει από το doctorLoggeg.jsp.

Αρχικά συνδέεται με τη βάση usersdb με τη βοήθεια της κλάσης

DatabaseConnection. Στη συνέχεια κάνει insert νέα στήλη με τα στοιχεία που έδωσε ο γιατρός. Τέλος κάνει χειρισμό εξαιρέσεων με try-catch blocks.

3.2.4 RetrieveDataFromDB

3.2.4.1 GetUsers

Η κλάση αυτή επιστρέφει τους χρήστες (users) της βάσης usersdb σε λίστα.

Μέθοδοι

ArrayList()

Αρχικοποιεί μια ArrayList τύπου Users. Στη συνέχεια, μέσα σε ένα try-catch block συνδέεται με τη βάση, από την οποία διαλέγει όλους τους χρήστες και όλα τους τα χαρακτηριστικά. Μετά δημιουργεί νέα αντικείμενα τύπου Users επαναληπτικά και τους προσθέτει στη λίστα. Τέλος επιστρέφει την λίστα αυτή και κάνει χειρισμό εξαιρέσεων.

3.2.4.2 GetPatients

Η κλάση αυτή επιστρέφει τους ασθενείς (Patients) της βάσης usersdb σε λίστα.

Μέθοδοι

ArrayList()

Αρχικοποιεί μια ArrayList τύπου Patients. Στη συνέχεια, μέσα σε ένα try-catch block συνδέεται με τη βάση, από την οποία διαλέγει όλους τους χρήστες και όλα τους τα χαρακτηριστικά. Μετά δημιουργεί νέα



αντικείμενα τύπου Patient επαναληπτικά και τους προσθέτει στη λίστα. Τέλος επιστρέφει την λίστα αυτή και κάνει χειρισμό εξαιρέσεων.

3.2.4.3 GetDoctors

Η κλάση αυτή επιστρέφει τους γιατρούς (Doctors) της βάσης usersdb σε λίστα.

Μέθοδοι

ArrayList()

Αρχικοποιεί μια ArrayList τύπου Doctor. Στη συνέχεια, μέσα σε ένα try-catch block συνδέεται με τη βάση, από την οποία διαλέγει όλους τους χρήστες και όλα τους τα χαρακτηριστικά. Μετά δημιουργεί νέα αντικείμενα τύπου Doctor επαναληπτικά και τους προσθέτει στη λίστα. Τέλος επιστρέφει την λίστα αυτή και κάνει χειρισμό εξαιρέσεων.

3.2.4.4 GetAppointments

Η κλάση αυτή επιστρέφει τα ραντεβού (Appointment) της βάσης usersdb σε λίστα.

Μέθοδοι

ArrayList()

Αρχικοποιεί μια ArrayList τύπου Appointment. Στη συνέχεια, μέσα σε ένα try-catch block συνδέεται με τη βάση, από την οποία διαλέγει όλους τα ραντεβού και όλα τους τα χαρακτηριστικά. Μετά δημιουργεί νέα αντικείμενα τύπου Appointment επαναληπτικά και τα προσθέτει στη λίστα. Τέλος επιστρέφει την λίστα αυτή και κάνει χειρισμό εξαιρέσεων.

3.2.4.4 GetAdmins

Η κλάση αυτή επιστρέφει τους διαχειριστές (Admins) της βάσης usersdb σε λίστα.

Μέθοδοι

ArrayList()

Αρχικοποιεί μια ArrayList τύπου Admin. Στη συνέχεια, μέσα σε ένα try-catch block συνδέεται με τη βάση, από την οποία διαλέγει όλους τους διαχειριστές και όλα τους τα χαρακτηριστικά. Μετά δημιουργεί νέα αντικείμενα τύπου Admin επαναληπτικά και τα προσθέτει στη λίστα. Τέλος επιστρέφει την λίστα αυτή και κάνει χειρισμό εξαιρέσεων.

4 Βιβλιογραφία

- [hash + Salt](#)
- [gunet](#)