

Project HY359

Χρηστος Παπασταμος
csd4569

Δημητρης Μπισιας
csd4273

Κλασεις & Μεθοδοι

Αρχικα στις κλασεις της βασης δεδομενων που μας δοθηκαν, σε καποιες προσθεσαμε βοηθητικες μεθοδους για αναζητηση βασισμενες σε αυτες που ειχαν δωθει. Δημιουργησαμε επισης κλασεις για τα Servlets τα οποια θα αναφερουμε παρακατω. Μια πολυ χρησιμοπονημενη μεθοδος μας ειναι η `createResponse(HttpServletResponse response, int statuscode, String data)` η οποια δημιουργει ευκολα responses με μια εντολη.

Servlets

Το project μας περιεχει τα εξης servlets:

- **BloodTestsAPI:**
 - doGet: Συμφωνα με το πεδιο amka που ερχεται απο το query part, το servlet γυρναι τα Bloodtests του συγκεκριμενου χρηστη (σε json object)
 - doPost: Δεχεται ενα json object με τις τιμες για την δημιουργια ενος καινουριου bloodtest (και υπολογιζει τις τιμες)
- **CertifiedDoctorBoy:**
 - doGet: Συμφωνα με το πεδιο certified που παιρνει απο το query part, το servlet γυρναι τους γιατρους (certified ή ολους)
 - doPost: Δεχεται ενα json object με τις τιμες για την δημιουργια ενος καινουριου Doctor
 - doPut: Δεχεται ενα json object με τις τιμες για την ενημερωση ενος ηδη υπαρχων Doctor
- **GetInfo:**
 - doGet: Ελενχει αν υπαρχει στο session αποθηκευμενη τιμη logged_in και password. Αν αυτες υπαρχουν και αντιστοιχουν σε doctor, user ή admin, το servlet επιστρεφει ενα json object με ολα τα στοιχεια του χρηστη.
- **Login:**
 - doPost: Δεχεται ενα json object με τιμες logged_in και password. Αν υπαρχουν αυτες επιστρεφει ενα json object που επιβεβαιωνει την επιτυχημενη συνδεση του χρηστη

- **MessagesAPI:**
 - doGet: Συμφωνα με το πεδίο bloodtype επιστρεφει τα μηνυματα για αιμοδοσια. Αν δεν υπαρχει, τοτε συμφωνα με τα πεδια doctor_id και user_id επιστρεφει μια συνομιλια μεταξυ ενος γιατρου και ενος χρηστη
 - doPost: Δεχεται ενα json object με τις τιμες για την δημιουργια ενος καινουριου Message
- **Register:**
 - doPost: Δεχεται ενα json object με τις τιμες για την δημιουργια ενος καινουριου Χρηστη (διακρινει users και doctors απο το πεδίο usertype)
- **RendezvousAPI:**
 - doGet: Αν δεχτει το πεδίο doctor_id στο query part, επιστρεφει τα ραντεβου του αντιστοιχου γιατρου (το πεδίο mode αναφερεi αν θα επιστραφουν τα ελευθερα η τα επιλεγμενα). Αν δεχτει το πεδίο user_id στο query part, επιστρεφει τα ραντεβου του χρηστη
 - doPost: Δεχεται ενα json object με τις τιμες για την ενημερωση ενος ραντεβου απο εναν γιατρο
 - doGet: Δεχεται ενα json object με τις τιμες για την δημιουργια ενος καινουριου ραντεβου απο εναν γιατρο
 - doDelete: Δεχεται ενα json object το οποιο αναφερεi ενα ραντεβου το οποιο θα ακυρωθει
- **TestForDuplicates:**
 - doPost: Δεχεται ενα json object. Αν αυτο περιεχει τα πεδια username, email η/και amka το servlet επιστρεφει 1 αν υπαρχει και 0 αν δεν υπαρχει ηδη στην βαση δεδομενων
- **TreatmentsAPI:**
 - doGet: Αν το query part περιεχει το πεδίο user_id, επιστρεφει τις θεραπειες του χρηστη. Αν περιεχει το πεδίο bloodtest_id, επιστρεφει ενα συγκεκριμενο bloodtest
 - doPost: Δεχεται ενα Json object με τις τιμες για την δημιουργια ενος καινουριου Bloodtest
- **UserAPI:**
 - doGet: Αν το query part περιεχει το πεδίο type με τιμη patient, επιστρεφει τους ασθενεις με done ραντεβου με εναν γιατρο (που αναφερεται απο το doctor_id στο το query part). Αν το πεδίο type εχει την τιμη user, επιστρεφει ολους τους χρηστες της βασης εκτως του admin. Αν το πεδίο type δεν εχει οριστεί, γυρναι ολους τους χρηστες
 - doPost: Δεχεται ενα json object με τις τιμες για την ενημερωση των πεδιων ενος χρηστη
 - doDelete: Δεχεται ενα json object το οποιο αναφερεi εναν χρηστη προς διαγραφη

Front-End

(Υπάρχουν screenshots στο παραδοτέο για όσα αναφερούμε παρακάτω)

Η αρχική σελίδα έχει στο nav bar δυο κουμπια για Register και Login. Απο κατω εμφανιζονται ολοι οι ιατροι οι οποιοι ειναι certified. Κατω στο footer βρισκονται κουμπια για redirect στα φαρμακεια που βρισκονται στο ηρακλειο καθως και για τα νοσοκομεια

➤ /admin

Στην σελιδα του admin υπαρχουν 3 κουμπια. Το πρωτο εμφανιζει ολους τους ιατροους certified και μη και ο admin απο εκει μπορει να κανει οποιους θελει certified. Το δευτερο εμφανιζει ολους τους απλους χρηστες και ο admin απο εκει μπορει να διαγραφει τους λογαριασμους τους. Το τριτο ειναι ενα απλο logout κουμπι.

➤ /doctor

Με το που κανει login ενας γιατρος εμφανιζονται τα στοιχεια του και μπορει να κανει ο,τι αλλαγες θελει (εκτος απο username, amka και τον τυπο του χρηστη). Επισης στο nav bar υπαρχουν 5 νεα κουμπια το My Info, το Rendezvous, το Patients, το Messages και τελος το Notify Blood Donors. Το πρωτο εμφανιζει τα στοιχεια του ιατρου και επιτρεπει την επεξεργασια τους. Το δευτερο εμφανιζει τα rendezvous του ιατρου και μπορει ο ιατρος να τα δει και να κανει cancel ή να ορισει καποιο ως done. Επισης μπορει να δημιουργησει και ενα νεο rendezvous. Το τριτο εμφανιζει ολους τους ασθενεις που εχουν τελειωμενο (done) rendezvous με τον ιατρο που ειναι logged in και επιτρεπει στον ιατρο να δει τα προηγουμενα τεστ του (αν εχει), να προσθεσει καινουριο τεστ και επισης να δει τις θεραπειες για καποιο τεστ. Το τεταρτο επιτρεπει στον ιατρο να δει τα chat με τους ασθενεις του και να στειλει και να λαβει νεα μηνυματα. Το πεμπτο επιτρεπει στον ιατρο να στειλει ενα μηνυμα σε ολα τα ατομα μιας συγκεκριμενης ομαδας αιματος (πχ για να πανε να δωσουν αιμα).

➤ /user

Με το που κανει login ενας απλος χρηστης εμφανιζονται (οπως και στον ιατρο) τα στοιχεια του και μπορει να κανει ο,τι αλλαγες θελει (με τους ιδιους περιορισμους που εχει και ο ιατρος). Στο nav bar υπαρχουν 5 κουμπια παλι. Το πρωτο ειναι παλι το My Info που εμφανιζει οπως και πριν τα στοιχεια του χρηστη. Το δευτερο ειναι το Test and Treatments οπου ενας χρηστης μπορει να δει τα τεστ του και να προσθεσει νεο και επισης μπορει να δει τις θεραπειες για καποιο συγκεκριμενο τεστ. Επισης υπαρχει και οπτικοποιηση σε ενα γραφο. Το τριτο ειναι το Rendezvous που επιτρεπει σε ενα χρηστη να δει τα rendezvous του και επισης να κλεισει νεο απο μια λιστα ιατρων που εμφανιζονται τα free rendezvous τους. Το τεταρτο ειναι το Doctors το οποιο εμφανιζει τους διαθεσιμους (certified) γιατροους, ταξινομημενους βαση αποστασης ή διαρκειας

αποστασης. Το πεμπτο είναι το Messages που είναι το ίδιο με τους ιατρούς και μπορεί να δει τα μηνύματα του με κάποιον ιατρό

JavaScript

Τα περισσότερα αρχεία html του frontend μας συνοδεύονται με τον αντίστοιχο κώδικα της σελίδας ο οποίος υπάρχει στην αρχή του εκαστοτε αρχείου. Μεσα σε κάθε directory frontend υπάρχει και το αντίστοιχο script.js αρχείο το οποίο περιεχει βασικές συναρτήσεις και λειτουργίες για την φόρτωση της σελίδας. Στον φακέλο scripts υπάρχει το αρχείο usercontrols.js το οποίο περιεχει βασικές συναρτήσεις που χρησιμοποιούνται στην λειτουργία του χρήστη. Χρησιμοποιήσαμε επίσης τα εξής javascript libraries: OpenLayers, JQuery, xml2json, momentjs, gstatic και bootstrap.

CSS

Στο project μας χρησιμοποιήσαμε δυο stylesheets, ένα για το admin page και ένα για τα υπολοίπα pages.

Ajax

Όλα τα functions του project μας (από το frontend) χρησιμοποιουν ajax requests με σκοπο να στείλουν requests στον server χωρίς να χρειάζεται να γίνει ανανέωση της σελίδας.

REST

Δεν καταφέραμε ποτε να setάρουμε (από την άσκηση 3) το REST στο IntelliJ, οπότε μετά από την πρόταση του Κυρίου Μουνταντωνάκη, τα υλοποιήσαμε όλα σε servlet (εξου και τα ονόματα των servlet)

Database

Δεν χρειάστηκε να κάνουμε καμία αλλαγή στην βάση δεδομένων παρόλο που σκεφτήκαμε πολλές τροποποιήσεις που θα την έκαναν πιο ευχρήστη και κατανοησιμη