ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον παγκόσμιο ιστό»

«τελική εργασία»	***ΤΕΛΙΚΟ ΠΑΡΑΔΟΤΕΟ ΜΑΘΗΜΑΤΟΣ***
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Παπαϊωάννου Ερατώ
Ημερομηνία παράδοσης	4/7/2017

Εκφώνηση της άσκησης

Στόχοι εργασίας: Ολοκλήρωση λειτουργικότητας *3-tier εφαρμογής, ολοκλήρωση server-side τεχνολογιών (servlets και προαιρετικά jsp), επικοινωνία με βάση δεδομένων, ολοκλήρωση λειτουργιών.*
Στην τελική εργασία του μαθήματος θα επεκτείνετε τις προηγούμενες ασκήσεις ώστε να δημιουργήσετε μία εφαρμογή τριών επιπέδων (3-tier), η οποία θα υλοποιεί όλες τις λειτουργίες (μεθόδους) που ορίσατε στις προηγούμενες ασκήσεις.

Αναλυτικά Βήματα:

1. Επέκταση web project προηγούμενης άσκησης

1.1. Στην τελική εργασία θα επεκτείνετε τη λειτουργικότητα του web project που δημιουργήσατε στην προηγούμενη άσκηση και θα υλοποιήσετε όλη την ζητούμενη λειτουργικότητα για κάθε κατηγορία χρηστών.

2. Δημιουργία διαδικτυακής διεπαφής

2.1. Για την είσοδο των χρηστών στο σύστημα θα υλοποιείστε **μηχανισμό login** με username και password. Το password θα αποθηκεύεται σε κρυπτογραφημένη (hashed+salted) μορφή. Από την αρχική σελίδα οι διάφοροι χρήστες θα μπορούν να έχουν πρόσβαση στις λειτουργίες τους.

2.2. Σε αυτό το βήμα, θα υλοποιήσετε τις διαδικτυακές διεπαφές (html σελίδες) που θα χρησιμοποιούν οι χρήστες όλων των κατηγοριών (Ασθενείς, Ιατροί, Διαχειριστές) για να αλληλεπιδρούν με την εφαρμογή και να χρησιμοποιούν τις αντίστοιχες μεθόδους που απαιτούνται.

2.2.1. Θα υπάρχει ένα κεντρικό μενού σε μία index.html σελίδα, η οποία θα είναι η αρχική σελίδα για όλους τους χρήστες. Μετά το login θα προβάλλεται το μενού λειτουργιών κάθε χρήστη ανάλογα με την κατηγορία στην οποία ανήκει.

2.2.2. Λειτουργίες Ασθενών (Patient): Οι Ασθενείς θα μπορούν να εκτελούν κατ ελάχιστο τις λειτουργίες: προβολή ιστορικού προηγούμενων ραντεβού, προβολή διαθέσιμων κενών για κλείσιμο ραντεβού με έναν γιατρό κάποιας ειδικότητας, κλείσιμο ραντεβού ακύρωση ραντεβού (σε περίπτωση που είναι τουλάχιστον 3 ημέρες μετά).

2.2.3. Λειτουργίες Ιατρών (Doctor): Οι Ιατροί θα μπορούν να εκτελούν κατ ελάχιστο τις λειτουργίες: καταχώρηση διαθεσιμότητας για ραντεβού (ανά μήνα), προβολή πίνακα ραντεβού, ακύρωση ραντεβού (σε περίπτωση που είναι τουλάχιστον 3 ημέρες μετά).

2.2.4. Λειτουργίες Διαχειριστή (Administrator). Οι Διαχειριστές θα μπορούν να εκτελούν κατ ελάχιστο τις λειτουργίες: εισαγωγή νέου Ιατρού και χρήστη, διαγραφή Ιατρού.

2.3. Η εφαρμογή θα υποστηρίζει **διαχείριση συνόδου (session management)** από τη στιγμή που ο χρήστης συνδέεται, μέχρι την αποσύνδεσή του από την εφαρμογή. Κατά την αποσύνδεση του χρήστη θα πρέπει να διαγράφεται το session.

3. Υλοποίηση επιπέδου Δεδομένων και σύνδεση εφαρμογής με τη βάση

3.1. Όλα τα δεδομένα θα αποθηκεύονται σε βάση δεδομένων, την οποία έχετε σχεδιάσει από την 2η Άσκηση (στο παράδειγμα χρησιμοποιήσαμε mysql και mysql workbench). Μπορείτε να προβείτε σε όποιες τροποποιήσεις θεωρείτε απαραίτητες. Προσθέσετε δοκιμαστικά δεδομένα στη βάση.

3.2. Διαμορφώστε κατάλληλα το project σας ώστε να συνδέσετε τη Βάση Δεδομένων που έχετε δημιουργήσει με τον application server σας, ως μία 3-tier εφαρμογή (μπορείτε να βρείτε αντίστοιχο παράδειγμα στα παραδείγματα κώδικα που περιλαμβάνονται στη σελίδα του μαθήματος).

4. Υλοποίηση επιπέδου επεξεργασίας (servlet)

4.1. Διαμορφώστε κατάλληλα το project σας ώστε να επικοινωνεί με τον application server της επιλογής σας (στα java παραδείγματα έχουμε χρησιμοποιήσει apache tomcat).

4.2. Υλοποιήσετε όλες τις λειτουργίες που προσφέρει η εφαρμογή σας χρησιμοποιώντας τεχνολογία servlet. Δημιουργήστε ένα ή περισσότερα servlet τα οποία θα δέχονται είσοδο από το επίπεδο διεπαφής (html σελίδες και φόρμες), θα αναζητούν στη βάση δεδομένων τα στοιχεία που απαιτούνται ότι απαιτείται και θα επιστρέφουν το αποτέλεσμα στον εκάστοτε χρήστη ως δυναμική html σελίδα.

4.3. Προαιρετικά, μπορείτε να χρησιμοποιήσετε τεχνολογία jsp για τη δημιουργία και την διαμόρφωση των ιστοσελίδων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1 Γενική Περιγραφή της λύσης

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε ο server Tomcat 8.5, για τη βάση η MySql Workbench και το Dynamic Web Project αναπτύχθηκε σε Java στο περιβάλλον Eclipse. Η σύνδεση του server με τη βάση πραγματοποιήθηκε με τον database connector, mysql-connector-java-5.1.42.

Το dynamic web project αποτελείται από 2 packages, classes και servlets που περιέχουν τις κλάσεις και τα servlets αντίστοιχα, και από html και jsp σελίδες και από ένα css αρχείο:

- Τρεις κλάσεις που χρησιμοποιούνται, Admin, Doctor και Patient
- Έξι servlets, LoginServlet, DoctorServlet, PatientServlet, AdminServlet, RegisterDoctorServlet, RegisterPatientServlet
- Τρεις HTML σελίδες, index.html, insert_doctor.html, insert_patient.html
- Τρεις JSP σελίδες, admin.jsp, doctor.jsp, patient.jsp
- Το αρχείο CSS, css.css

Το project συνδέεται με τη βάση doctors_online μέσω του παρακάτω κώδικα που βρίσκεται στο αρχείο context.xml του server και σε κάθε servlet γίνεται αρχικοποίηση του datasource. Η σύνδεση στη βάση μέσα στο πρόγραμμα γίνεται με PreparedStatements και queries τα περισσότερα από τα οποία εισάγονται από τις κλάσεις ως Strings.

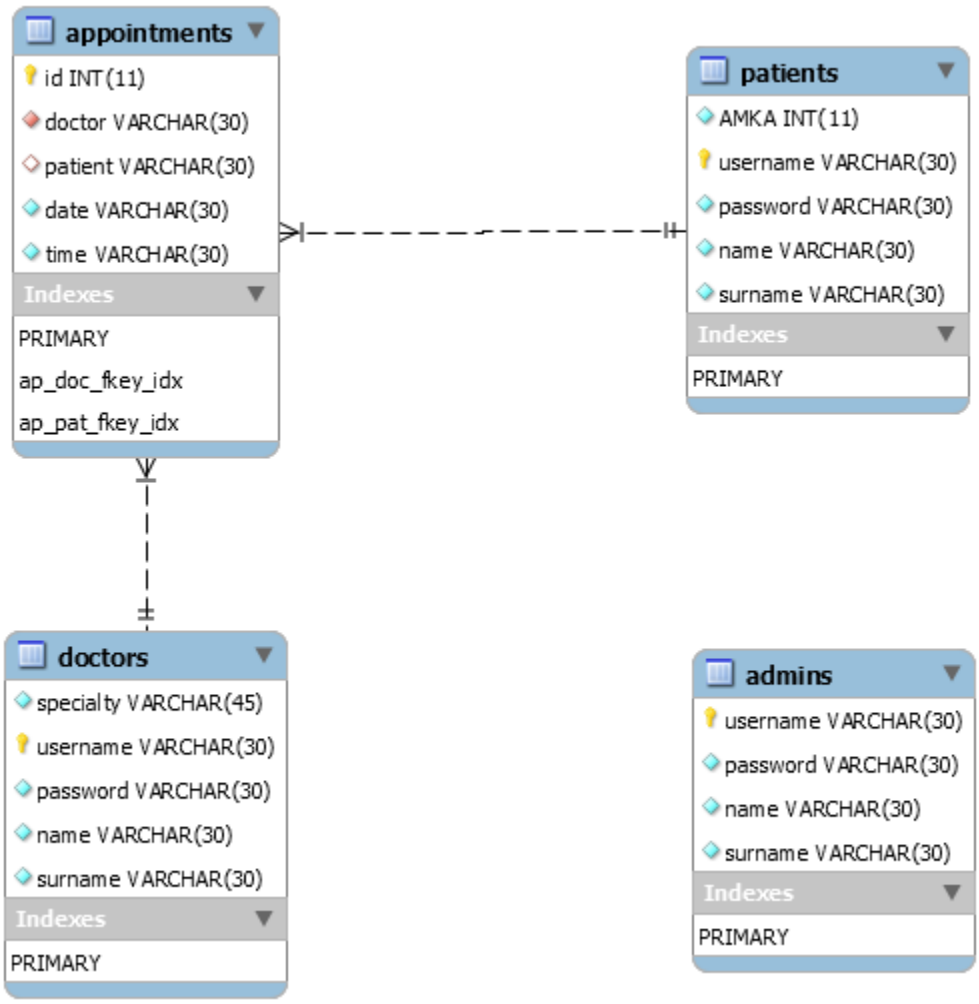
```
<Resource name="jdbc/LiveDataSource" auth="Container"
 driverClassName="com.mysql.jdbc.Driver"
 type="javax.sql.DataSource"
 username="root"
 password="Erato1"
 url="jdbc:mysql://localhost:3306/doctors_online?useSSL=false"
 maxActive="8" >
</Resource>
```

1.1 Βάση δεδομένων

Δημιουργήθηκε το schema την βάσης (doctors_online) και οι πίνακες και οι συσχετίσεις τους, ανάλογα με το παρακάτω μοντέλο με το εργαλείο Mysql Workbench. Η βάση αποτελείται από 4 πίνακες: admins (**username**,password,name,surname), doctors(specialty,**username**,password,name,surname), patients(AMKA,**username**,password,name,surname) και appointments(**id**,doctor,patient,date,time), με δύο εξωτερικά κλειδιά στους πίνακες patients και doctors.

Στον πίνακα patients υπάρχει πάντα ο patient με όνομα χρήστη “null” που χρησιμοποιείται όταν εισάγει ο γιατρός κάποια διαθέσιμη ώρα για να μην υπάρξει πρόβλημα με το εξωτερικό κλειδί. Τότε αυτή η ώρα εισάγεται στον πίνακα appointments σαν ραντεβού με κενό ασθενή.

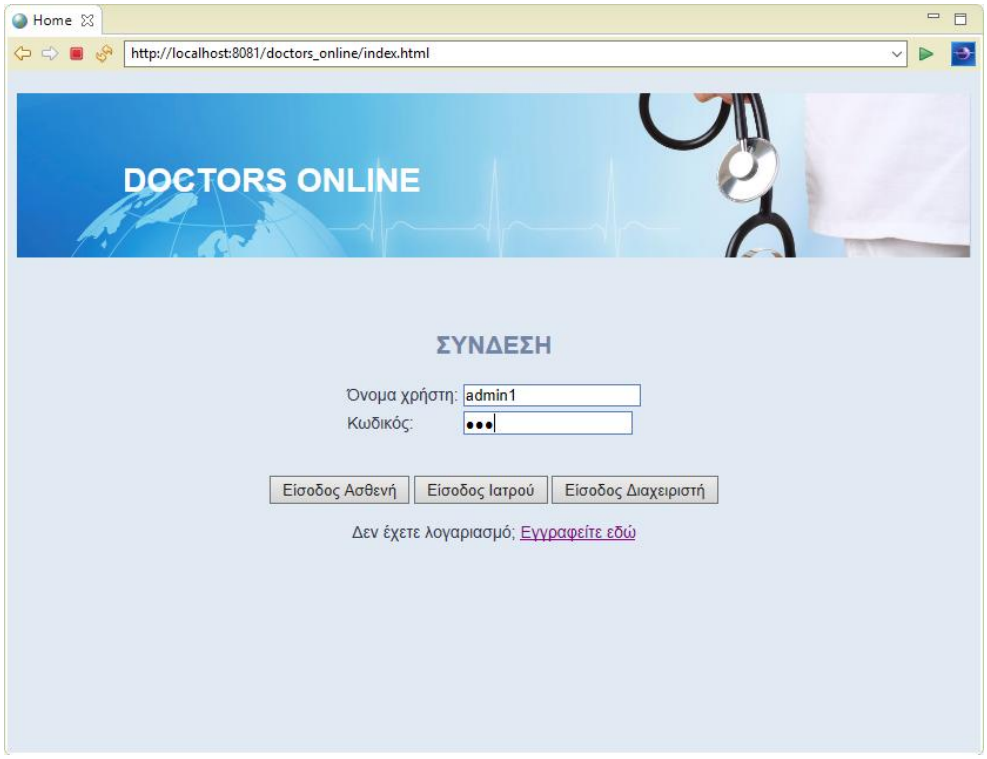
Στην διαγραφή κάποιου γιατρού διαγράφονται αυτόματα τα ραντεβού του, με το εξωτερικό κλειδί.



1.2 HTML

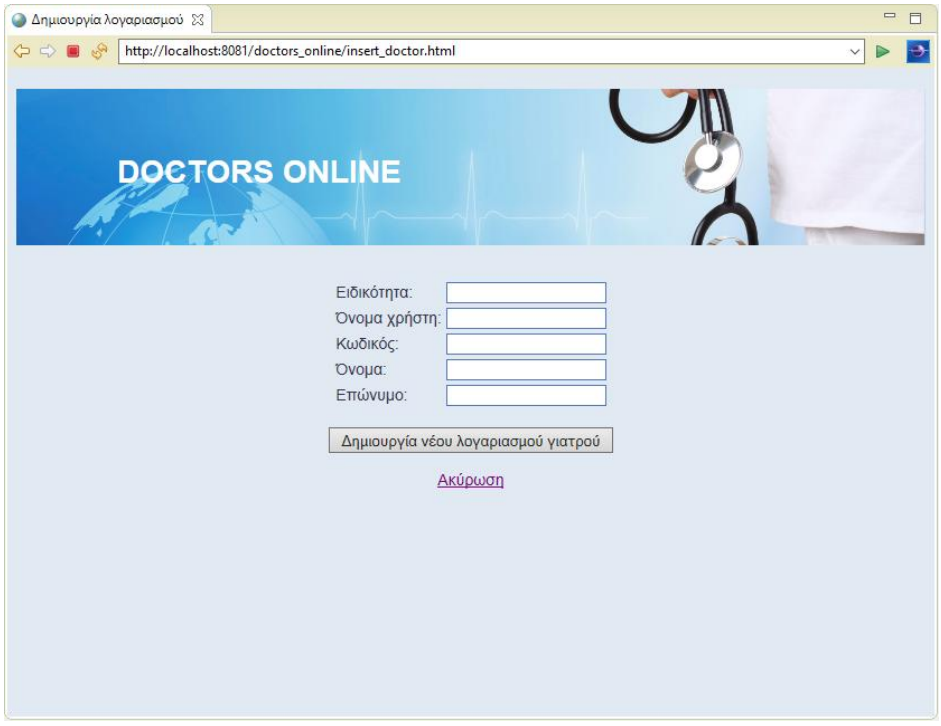
1.2.1 index.html

Είναι η αρχική σελίδα της εφαρμογής. Αποτελείται από μία φόρμα στην οποία ο χρήστης που θέλει να συνδεθεί εισάγει το όνομα χρήστη του και τον κωδικό του και υποβάλλει τα στοιχεία με ένα από τα τρία κουμπιά, είσοδος ως ασθενής, ως γιατρός ή ως διαχειριστής. Η φόρμα χρησιμοποιεί τη μέθοδο post και οδηγεί στο LoginServlet. Επίσης υπάρχει μία υπερσύνδεση για εγγραφή νέου χρήστη (μόνο για ασθενείς) η οποία οδηγεί στην σελίδα insert_patient.html



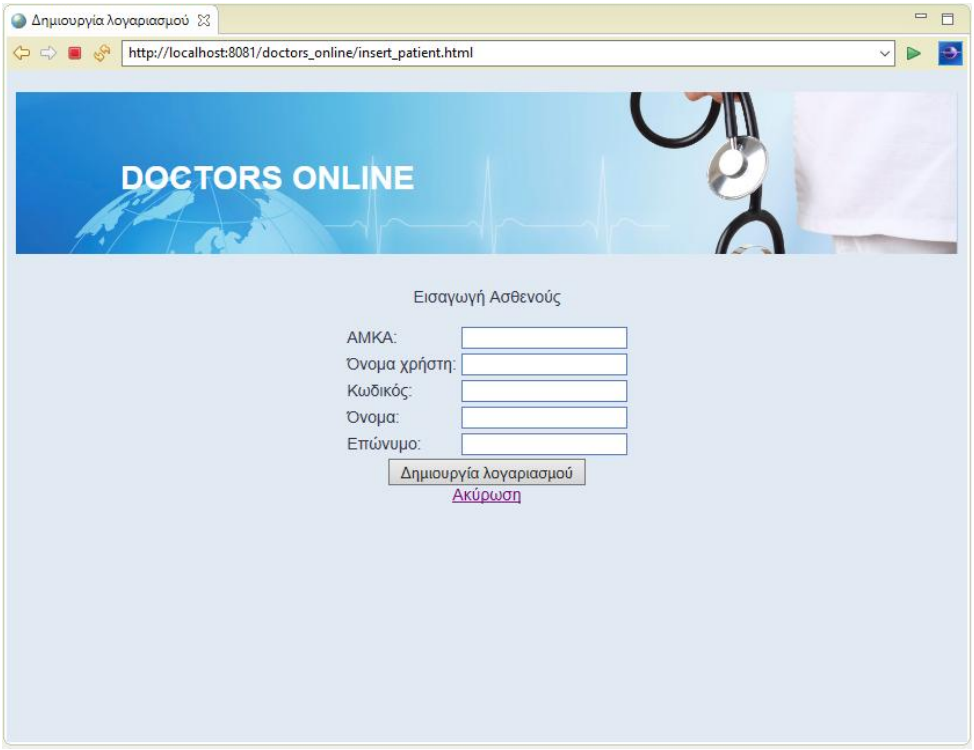
1.2.2 insert_doctor.html

Η σελίδα αυτή είναι προσβάσιμη μόνο από την αρχική σελίδα κάποιου διαχειριστή ο οποίος εισάγει τα στοιχεία ενός γιατρού για να τον προσθέσει στη βάση. Αυτό γίνεται μέσω μίας φόρμας που κάνει χρήση της μεθόδου post και πατώντας το κουμπί submit οδηγεί στο RegisterDoctorServlet. Επίσης υπάρχει μία υπερσύνδεση για ακύρωση και επιστροφή στην αρχική του διαχειριστή.



1.2.3 insert_patient.thml

Η σελίδα αυτή χρησιμοποιείται για την εισαγωγή νέου ασθενή στη βάση. Αποτελείται από μία φόρμα και μία υπερσύνδεση που οδηγεί στο index.html. Στη φόρμα ο ασθενής εισάγει τα στοιχεία του και τα υποβάλλει πατώντας το κουμπί. Η φόρμα χρησιμοποιεί τη μέθοδο post και οδηγεί στο RegisterPatientServlet.

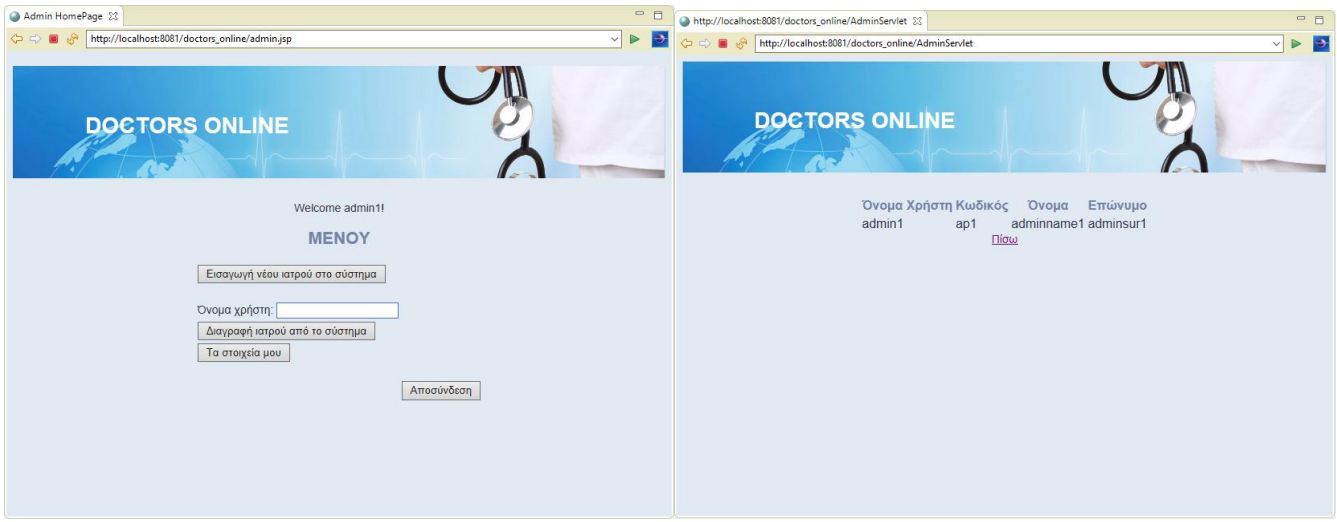


1.3 JSP

1.3.1 admin.jsp

Αυτή είναι η αρχική σελίδα του διαχειριστή που περιέχει το κύριο μενού. Αρχικά η σελίδα ανακτά το όνομα χρήστη του διαχειριστή, έπειτα αποτρέπει τη χρήση του cache memory και σε περίπτωση που ο χρήστης δεν είναι συνδεδεμένος γίνεται redirect στην αρχική σελίδα (index.html) και εμφανίζεται μήνυμα ότι η σελίδα αυτή έχει λήξει.

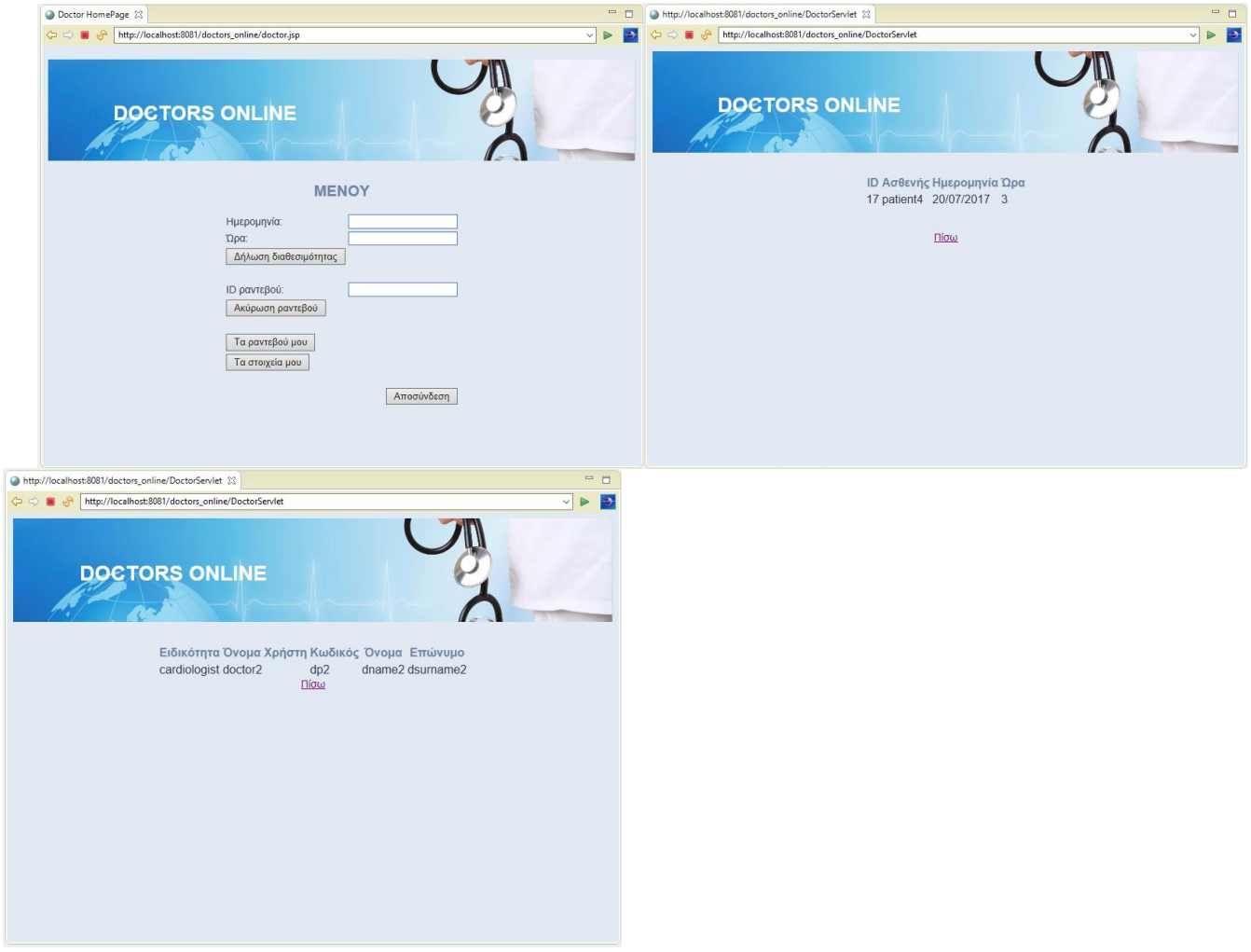
Ακολουθεί η φόρμα html που κάνει χρήση της μεθόδου post και οδηγεί στο AdminServlet. Οι επιλογές είναι η εισαγωγή νέου γιατρού, η διαγραφή λογαριασμού γιατρού, η προβολή των στοιχείων του χρήστη ή η αποσύνδεση, πατώντας ένα από τα τέσσερα κουμπιά τύπου submit. Για τη διαγραφή γιατρού ο διαχειριστής εισάγει το όνομα χρήστη του γιατρού στο πεδίο κειμένου.



1.3.2 doctor.jsp

Αυτή είναι η αρχική σελίδα του γιατρού που περιέχει το κύριο μενού. Αρχικά η σελίδα ανακτά το όνομα χρήστη του γιατρού, έπειτα αποτρέπει τη χρήση του cache memory και σε περίπτωση που ο χρήστης δεν είναι συνδεδεμένος γίνεται redirect στην αρχική σελίδα (index.html) και εμφανίζεται μήνυμα ότι η σελίδα αυτή έχει λήξει.

Ακολουθεί η φόρμα html που κάνει χρήση της μεθόδου post και οδηγεί στο DoctorServlet. Οι επιλογές είναι η εισαγωγή νέας διαθέσιμης ώρας, η ακύρωση ραντεβού, η προβολή των ραντεβού και των στοιχείων του γιατρού και η αποσύνδεση, πατώντας ένα από τα πέντε κουμπιά τύπου submit.



1.3.3 patient.jsp

Αυτή είναι η αρχική σελίδα του ασθενή και περιέχει το κύριο μενού. Η συγκεκριμένη σελίδα πραγματοποιεί σύνδεση με τη βάση και γι αυτό γίνεται αρχικοποίηση του datasource της βάσης. Έπειτα η σελίδα ανακτά το όνομα χρήστη του ασθενή, αποτρέπει τη χρήση του cache memory και σε περίπτωση που ο χρήστης δεν είναι συνδεδεμένος γίνεται redirect στην αρχική σελίδα (index.html) και εμφανίζεται μήνυμα ότι η σελίδα αυτή έχει λήξει.

Η σελίδα αποτελείται από τέσσερις φόρμες. Η πρώτη και η δεύτερη φόρμα χρησιμοποιούνται για το ραντεβού με οποιονδήποτε γιατρό μιας συγκεκριμένης ειδικότητας.

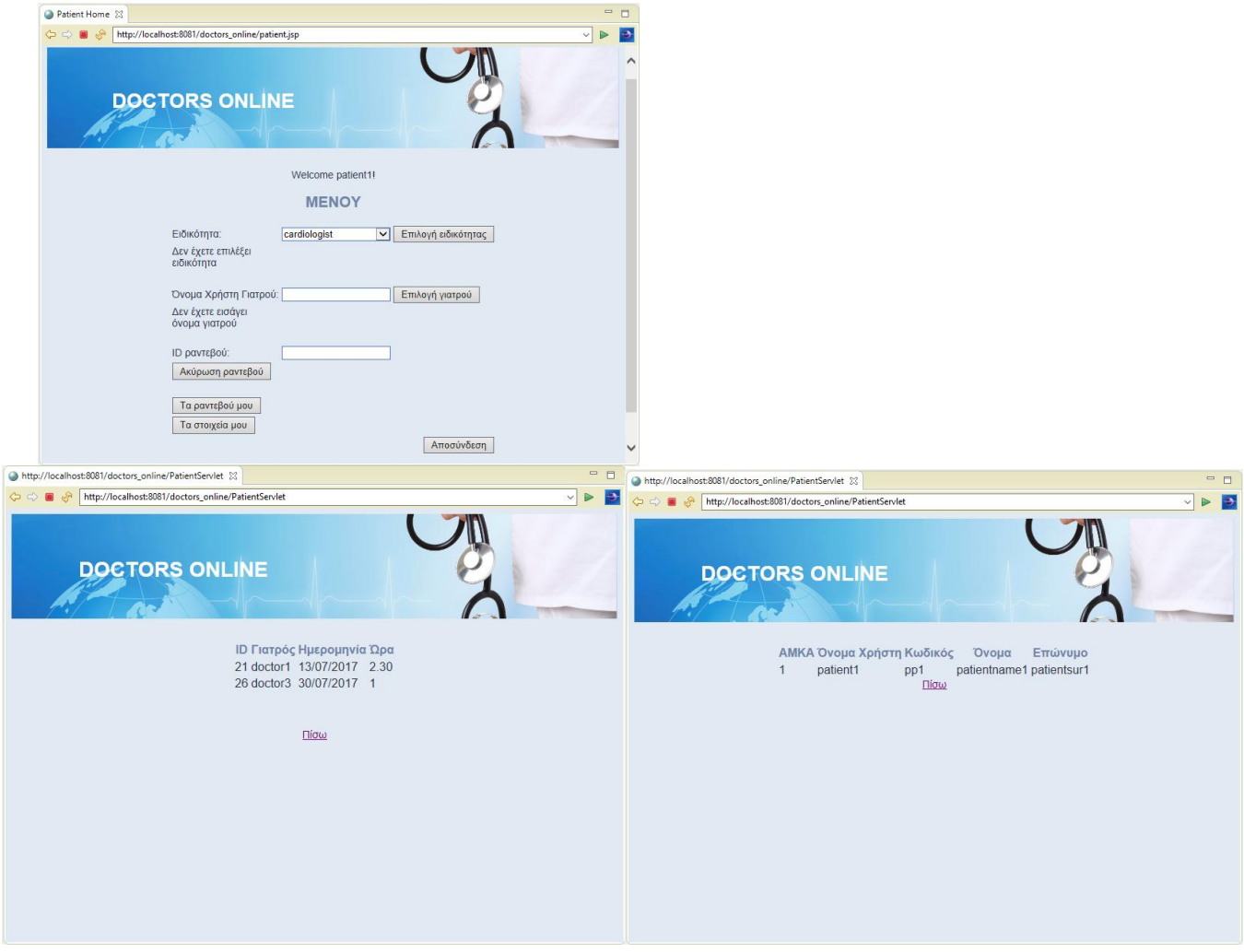
Η πρώτη αποτελείται από μία drop down λίστα τύπου select η οποία, με σύνδεση στη βάση, εμφανίζει όλες τις ειδικότητες γιατρών που υπάρχουν για να διαλέξει ο ασθενής, και ένα κουμπί τύπου submit που αποθηκεύει αυτή την επιλογή.

Η δεύτερη φόρμα χρησιμοποιεί τη μέθοδο post και εμφανίζεται μόνο στην περίπτωση που ο ασθενής έχει επιλέξει κάποια ειδικότητα αλλιώς εμφανίζεται στη θέση της μήνυμα. Αποτελείται από μία δεύτερη drop down λίστα τύπου select η οποία, με σύνδεση στη βάση, εμφανίζει όλα τα διαθέσιμα ραντεβού των γιατρών ανάλογα με την ειδικότητα που έχει διαλέξει παραπάνω ο ασθενής (με την getParameter), και ένα κουμπί τύπου submit που κλείνει τελικά το επιλεγμένο ραντεβού από τη λίστα μέσω του PatientServlet. Αν δεν υπάρχουν ραντεβού διαθέσιμα, εμφανίζεται μήνυμα στη θέση της φόρμας.

Η τρίτη φόρμα αποτελείται από ένα text input στο οποίο ο ασθενής εισάγει το όνομα χρήστη κάποιου συγκεκριμένου γιατρού για να κλείσει ραντεβού μαζί του, και από ένα κουμπί τύπου input που αποθηκεύει την επιλογή του ασθενή.

Η τέταρτη φόρμα χρησιμοποιεί τη μέθοδο post και οδηγεί στο PatientServlet. Αποτελείται αρχικά από ένα dropdown list με τα διαθέσιμα ραντεβού του γιατρού που εισήχθη προηγουμένως και από ένα κουμπί τύπου submit που υποβάλλει τα στοιχεία της φόρμας. Για τη λίστα με τα ραντεβού γίνεται σύνδεση στη βάση και ανάλογα με το value του submit της προηγούμενης φόρμας εμφανίζονται τα σωστά αποτελέσματα. Αν δεν υπάρχουν διαθέσιμα ραντεβού ή αν ο χρήστης δεν έχει εισάγει όνομα γιατρού αντί για τα παραπάνω εμφανίζεται μήνυμα.

Επίσης αυτή η φόρμα αποτελείται από άλλα 4 κουμπιά τύπου input και ένα πεδίο κειμένου. Στο πεδίο ο χρήστης εισάγει το id ενός από τα ραντεβού του για να το ακυρώσει και υποβάλλει την επιλογή του με το κουμπί Ακύρωση ραντεβού. Με τα άλλα 3 κουμπιά ο χρήστης εμφανίζει τα ραντεβού του, τα στοιχεία του ή κάνει αποσύνδεση.



1.4 Κλάσεις

1.4.1 Admin.java

Η κλάση αποτελείται από τον constructor και τις μεθόδους newDoctor(),removeDoctor(),showAdminInfo() οι οποίες είναι σε μορφή String και επιστρέφουν το κατάλληλο query το οποίο θα χρησιμοποιηθεί στο AdminServlet.

1.4.2 Doctor.java

Η κλάση αποτελείται από τον constructor, τα getters/setters και τις μεθόδους setAvailability()(δήλωση διαθεσιμότητας), showAgenda()(προβολή ραντεβού), cancelAppointment(), showDoctorInfo() οι οποίες είναι σε μορφή String και επιστρέφουν το κατάλληλο query το οποίο θα χρησιμοποιηθεί στο DoctorServlet.

1.4.3 Patient.java

Η κλάση αποτελείται από τον constructor, τα getters/setters και τις μεθόδους register(), appointment(),cancelAppointment(), showAppointments(), showTreatmentHistory(), showPatientInfo() οι οποίες είναι σε μορφή String και επιστρέφουν το κατάλληλο query το οποίο θα χρησιμοποιηθεί στο DoctorServlet.

1.5 Servlets

1.5.1 LoginServlet.java

Είναι το servlet στο οποίο μας οδηγεί η index.html και η λειτουργία του είναι η σύνδεση του χρήστη στο λογαριασμό του και η οδήγησή του στην αρχική του σελίδα. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : η συνάρτηση παίρνει το value των πεδίων κειμένου που εισήγαγε ο χρήστης (username και password) και ανάλογα με το κουμπί που έχει πατηθεί ορίζει το usertype (γαιτρός, ασθενής,διαχειριστής) και το link που θα οδηγήσει στην αρχική σελίδα της κάθε κατηγορίας. Αν η Boolean συνάρτηση checkUser(…) είναι true άρα ο χρήστης υπάρχει στη βάση δημιουργείται νέο Http Session και αποθηκεύεται το όνομα χρήστη σαν παράμετρος user. Έπειται καλείται η createDynamicPage(…) για να οδηγήσει το χρήστη στην αρχική του. Αν η checkUser είναι false ο χρήστης οδηγείται στο index.html
- Void createDynamicPage(String message,String link, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Το link αλλάζει ανάλογα με το κουμπί που έχει πατήσει ο χρήστης και οδηγεί πχ στην αρχική σελίδα του γιατρού αν ο χρήστης έχει κάνει είσοδο ως γιατρός ή στο index.html αν τα στοιχεία είναι λάθος.
- boolean checkUser(String username,String password,String usertype,HttpServletRequest request) : ελέγχει αν ο χρήστης με το username και το password στις παραμέτρους της συνάρτησης υπάρχει στη βάση. Το query της αναζήτησης στη βάση διαμορφώνεται ανάλογα με το usertype αν πχ είναι γιατρός η αναζήτηση γίνεται στον πίνακα με τους γιατρούς.

- Void createNullUser() : χρησιμοποιείται για να εισάγει στον πίνακα patients έναν «patient» με στοιχεία "null" για να μην δημιουργηθεί πρόβλημα με το εξωτερικό κλειδί στον πίνακα appointments όταν ο γιατρός εισάγει κάποια διαθέσιμη ώρα που ακόμη δεν έχει patient.

1.5.2 RegisterDoctorServlet.java

Είναι το servlet στο οποίο μας οδηγεί η admin.jsp και η λειτουργία του είναι η δημιουργία νέου λογαριασμού γιατρού. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : η συνάρτηση παίρνει το value των πεδίων κειμένου που εισήγαγε ο χρήστης και αν είναι ολοκληρωμένα τα εισάγει στη βάση με τη βοήθεια της συνάρτησης newDoctor() της κλάσης Admin.
- Void createDynamicPage(String message, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Υπάρχει link για επιστροφή στην αρχική του χρήστη

1.5.3 RegisterPatientServlet.java

Είναι το servlet στο οποίο μας οδηγεί η index.html και η λειτουργία του είναι η δημιουργία νέου λογαριασμού ασθενούς. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : η συνάρτηση παίρνει το value των πεδίων κειμένου που εισήγαγε ο χρήστης και αν είναι ολοκληρωμένα τα εισάγει στη βάση με τη βοήθεια της συνάρτησης register() της κλάσης Patient. Αν τα στοιχεία είναι λάθος καλείται η createDynamicPage(…).
- Void createDynamicPage(String message, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Υπάρχει link για επιστροφή στην αρχική.

1.5.4 AdminServlet.java

Είναι το servlet στο οποίο μας οδηγεί η admin.jsp και η λειτουργία του είναι να κάνει κάποια ενέργεια ανάλογα με την επιλογή που έχει κάνει ο χρήστης από το μενού στην αρχική του σελίδα. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση και γίνεται ανάκτηση του ονόματος χρήστη από το Http Session. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : ανάλογα με το κουμπί που έχει πατήσει ο διαχειριστής εκτελείται διαφορετικό κομμάτι κώδικα.

  «Εισαγωγή νέου ιατρού στο σύστημα» γίνεται redirect στη σελίδα insert_doctor.html.

  «Διαγραφή ιατρού από το σύστημα» γίνεται ανάκτηση του ονόματος του γιατρού από το πεδίο κειμένου, πραγματοποιείται σύνδεση με τη βάση και ο γιατρός διαγράφεται ή εμφανίζεται σελίδα με μήνυμα λάθους με την createDynamicPage(…).

  «Τα στοιχεία μου» καλείται η συνάρτηση showInfo(…)

Αλλιώς έχει επιλεγεί η αποσύνδεση οπότε γίνεται διαγραφή του Http Session και γίνεται redirect στην αρχική σελίδα.

- Void createDynamicPage(String message, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Υπάρχει link για επιστροφή στην αρχική.
- void showInfo(String username, HttpServletResponse response) : δημιουργεί μια dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη σε πίνακα. Γίνεται σύνδεση με τη βάση και με τη χρήση Prepared Statement εμφανίζονται τα στοιχεία του χρήστη username με την βοήθεια της συνάρτησης showAdminInfo() της κλάσης Admin.

1.5.5 DoctorServlet.java

Είναι το servlet στο οποίο μας οδηγεί η doctor.jsp και η λειτουργία του είναι να κάνει κάποια ενέργεια ανάλογα με την επιλογή που έχει κάνει ο χρήστης από το μενού στην αρχική του σελίδα. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση και γίνεται ανάκτηση του ονόματος χρήστη από το Http Session. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : ανάλογα με το κουμπί που έχει πατήσει ο διαχειριστής εκτελείται μία διαφορετική συνάρτηση. Αν έχει επιλεγεί η αποσύνδεση οπότε γίνεται διαγραφή του Http Session και γίνεται redirect στην αρχική σελίδα.
- Void createDynamicPage(String message, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Υπάρχει link για επιστροφή στην αρχική.
- void showInfo(String username, HttpServletResponse response) : δημιουργεί μια dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη σε πίνακα. Γίνεται σύνδεση με τη βάση και με τη χρήση Prepared Statement εμφανίζονται τα στοιχεία του χρήστη username με την βοήθεια της συνάρτησης showDoctorInfo() της κλάσης Doctor.
- void availability (String user, HttpServletRequest request, HttpServletResponse response) : εισάγει την ώρα και ημερομηνία που δήλωσε ο χρήστης ως διαθέσιμη στη βάση με τη χρήση Prepared Statement και την βοήθεια της συνάρτησης setAvailability () της κλάσης Doctor. Ανάλογα με το αποτέλεσμα της σύνδεσης καλείται η createDynamicPage() εμφανίζοντας το κατάλληλο μήνυμα.
- void cancel (String user, HttpServletRequest request, HttpServletResponse response) : ακυρώνεται το ραντεβού που επέλεξε ο χρήστης με σύνδεση στη βάση με τη χρήση Prepared Statement και την βοήθεια της συνάρτησης cancelAppointment() της κλάσης Doctor. Ανάλογα με το αποτέλεσμα της σύνδεσης καλείται η createDynamicPage() εμφανίζοντας το κατάλληλο μήνυμα.
- void showAppointments(String username, HttpServletResponse response) : δημιουργεί μια dynamic web page η οποία εμφανίζει τα ραντεβού του συνδεδεμένου χρήστη σε πίνακα. Γίνεται σύνδεση με τη βάση και με τη χρήση Prepared Statement εμφανίζονται τα στοιχεία του χρήστη username με την βοήθεια της συνάρτησης showAgenda() της κλάσης Doctor. Ανάλογα με το αποτέλεσμα καλείται η createDynamicPage(…) εμφανίζοντας το κατάλληλο μήνυμα.

1.5.6 PatientServlet.java

Είναι το servlet στο οποίο μας οδηγεί η patient.jsp και η λειτουργία του είναι να κάνει κάποια ενέργεια ανάλογα με την επιλογή που έχει κάνει ο χρήστης από το μενού στην αρχική του σελίδα. Στην αρχή γίνεται αρχικοποίηση του datasource για τη βάση και γίνεται ανάκτηση του ονόματος χρήστη από το Http Session. Οι συναρτήσεις είναι οι εξής:

- Void doPost(HttpServletRequest request, HttpServletResponse response) : ανάλογα με το κουμπί που έχει πατήσει ο ασθενής εκτελείται μία διαφορετική συνάρτηση. Αν έχει επιλεγεί η αποσύνδεση οπότε γίνεται διαγραφή του Http Session και γίνεται redirect στην αρχική σελίδα.
- void createDynamicPage(String message, HttpServletResponse response) : δημιουργεί δυναμική σελίδα εμφανίζοντας το μήνυμα message. Υπάρχει link για επιστροφή στην αρχική.
- void showAppointments(String username, HttpServletRequest request, HttpServletResponse response) :

  δημιουργεί μια dynamic web page η οποία εμφανίζει τα ραντεβού του συνδεδεμένου χρήστη σε πίνακα. Γίνεται σύνδεση με τη βάση και με τη χρήση Prepared Statement εμφανίζονται τα στοιχεία του χρήστη username με την βοήθεια της συνάρτησης showAppointments () της κλάσης Patient. Ανάλογα με το αποτέλεσμα καλείται η createDynamicPage(…) εμφανίζοντας το κατάλληλο μήνυμα.
- void showInfo(String username, HttpServletResponse response) : δημιουργεί μια dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη σε πίνακα. Γίνεται σύνδεση με τη βάση και με τη χρήση Prepared Statement εμφανίζονται τα στοιχεία του χρήστη username με την βοήθεια της συνάρτησης showPatientInfo () της κλάσης Patient.
- void appointment(String username, String action, HttpServletRequest request, HttpServletResponse response) : Χρησιμοποιείται για το κλείσιμο νέων ραντεβού (αν το action = “specialty” ή “name”) και τη διαγραφή ραντεβού (αν το action =” cancel”). Ανάλογα με το action παίρνουμε το value από το πεδίο κειμένου ή την drop down λίστα και γίνεται σύνδεση με τη βάση και είτε εισάγεται είτε διαγράφεται το ραντεβού. Ανάλογα με την εκτέλεση καλείται η createDynamicPage(…) και εμφανίζει το αντίστοιχο μήνυμα.

2 Κώδικας Προγράμματος

2.1 Κώδικας HTML σελίδων

2.1.1 index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" href="css.css" type="text/css">
<title>Home</title>
```

```
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>

<br><h2>ΣΥΝΔΕΣΗ</h2>
<form method="post" action="/doctors_online/LoginServlet" >
<table>
<tr><td>Όνομα χρήστη: </td><td><INPUT TYPE="text" NAME="NameField"></td></tr>
<tr><td>Κωδικός: </td><td><INPUT TYPE="password" NAME="PasswordField"></td></tr>
</table><br><br>
<INPUT TYPE = "submit" name="submit" value="Είσοδος Ασθενή">
<INPUT TYPE = "submit" name= "submit" value="Είσοδος Ιατρού">
<INPUT TYPE = "submit" name= "submit" value="Είσοδος Διαχειριστή">
</form>

<br />
Δεν έχετε λογαριασμό; <a href="/doctors_online/insert_patient.html">Εγγραφείτε εδώ</a>
<br />

</body>
</html>
```

2.1.2 insert_doctor.html

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="css.css" type="text/css">
<title>Δημιουργία λογαριασμού</title>
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>
	<form method="post" action="/doctors_online/RegisterDoctorServlet">
		<table>
			<tr>
				<td>Ειδικότητα:</td>
				<td><input type="text" name="specialty" /></td>

			</tr>
			<tr>
				<td>Όνομα χρήστη:</td>
				<td><input type="text" name="username" /></td>

			</tr>
			<tr>
				<td>Κωδικός:</td>
				<td><input type="text" name="password" /></td>

			</tr>
			<tr>
				<td>Όνομα:</td>
```

```
                        <td><input type="text" name="name" /></td>

                    </tr>
                    <tr>
                        <td>Επώνυμο:</td>
                        <td><input type="text" name="surname" /></td>

                    </tr>
                </table><br>
                <input type="submit" value="Δημιουργία νέου λογαριασμού γιατρού"><br><br>
        </form>
        <a href="/doctors_online/admin.html">Ακύρωση</a>
</body>
</html>
```

2.1.3 insert_patient.thml

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset="UTF-8">
<link rel="stylesheet" href="css.css" type="text/css">
<title>Δημιουργία λογαριασμού</title>
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>
<p>Εισαγωγή Ασθενούς </p>
        <form method="post" action="/doctors_online/RegisterPatientServlet">
                <input type="hidden" name="requestType" value="register" />
                <table>
                    <tr>
                        <td>AMKA:</td>
                        <td><input type="text" name="amka" /></td>

                    </tr>
                    <tr>
                        <td>Όνομα χρήστη:</td>
                        <td><input type="text" name="username" /></td>

                    </tr>
                    <tr>
                        <td>Κωδικός:</td>
                        <td><input type="text" name="password" /></td>

                    </tr>
                    <tr>
                        <td>Όνομα:</td>
                        <td><input type="text" name="name" /></td>

                    </tr>
                    <tr>
                        <td>Επώνυμο:</td>
                        <td><input type="text" name="surname" /></td>
```

```
                </tr>
            </table>
            <input type="submit" value="Δημιουργία λογαριασμού">
        </form>
        <a href="/doctors_online/index.html">Ακύρωση</a>
</body>
</html>
```

2.2 Κώδικας JSP σελίδων

2.2.1 admin.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
 <%
 String keep = (String)request.getSession().getAttribute("user");//ανάκτηση ονόματος χρήστη από
το session
response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); // HTTP 1.1.
response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
if (keep == null) {response.sendRedirect("index.html");}
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Admin HomePage</title>
</head><!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css.css" type="text/css">
<title>Home</title>
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>
<a style="text-align:left;">Welcome <%=keep %>!</a>
<table>
<h2>MENOY</h2>
<form action="/doctors_online/AdminServlet" method="Post">
<input type="hidden" name="requestType" value="type" />
<tr><td>
<input type="submit" name="admin" value="Εισαγωγή νέου ιατρού στο σύστημα"></td></tr>
<tr>
    <td> </td>

</tr>
<tr><td>Όνομα χρήστη: <input type="text" name="doctor"/></td></tr>
<tr><td><input type="submit" name="admin" value="Διαγραφή ιατρού από το σύστημα">
</td></tr>
```

```
<tr><td>
<input type="submit" name="admin" value="Τα στοιχεία μου"></td></tr>


<tr>
    <td> </td>

</tr>
<tr><td></td><td>
<input type = "submit" name = "admin" value="Αποσύνδεση" style="float:right;"></td></tr>
</form>

</table>
</body>
</html>
```

2.2.2 doctor.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <%
    String keep = (String)request.getSession().getAttribute("user");//ανάκτηση ονόματος χρήστη
από το session

response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); // HTTP 1.1.
response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
if (keep == null) {response.sendRedirect("index.html");}

%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css.css" type="text/css">
<title>Doctor HomePage</title>
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>
<h2>MENOY</h2>
<table>

<form action="/doctors_online/DoctorServlet" method="Post">

<tr><td>Ημερομηνία:</td><td><input type="text" name="Ημερομηνία"/></td></tr>
<tr><td>Ώρα:</td><td><input type="text" name="Ώρα"/></td></tr>
<tr><td><input type="submit" name="doctor" value="Δήλωση διαθεσιμότητας"></td>
<tr>
    <td> </td>

</tr>
<tr><td>ID ραντεβού:</td><td><input type="text" name="IDραντεβού"/></td></tr>
<tr><td>
<input type="submit" name="doctor" value="Ακύρωση ραντεβού"></td></tr>
```

```
<tr>
    <td> </td>

</tr>
<tr><td>
<input type="submit" name="doctor" value="Τα ραντεβού μου"></td></tr>
<tr><td>
<input type="submit" name="doctor" value="Τα στοιχεία μου"></td></tr>

<tr>
    <td> </td>

</tr>
<tr><td></td><td><input type = "submit" name = "doctor" value="Αποσύνδεση"
style="float:right;"></td>
</form>
</table>

</body>
</html>
```

2.2.3 patient.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<%@ page import=" javax.naming.InitialContext" %>
<%@ page  import=" javax.sql.DataSource" %>

<%
 DataSource datasource = null;
try
{
      InitialContext ctx = new InitialContext();
      datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
}
catch(Exception e)
{
      throw new ServletException(e.toString());
}
String keep = (String)request.getSession().getAttribute("user");



response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); // HTTP 1.1.
response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
if (keep == null) {response.sendRedirect("index.html");}
%>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<link rel="stylesheet" href="css.css" type="text/css">
<title>Patient Home</title>
</head>
<body>
<div>
<h1>DOCTORS ONLINE</h1>
</div>

<a>Welcome <%=keep %>!</a>
<h2>MENOY</h2>
<table>
<%
try
{
Connection con = datasource.getConnection();
String sql = "SELECT MIN(specialty) as specialty FROM doctors  group by specialty";
PreparedStatement ps = con.prepareStatement(sql);
ResultSet rs = ps.executeQuery();
%>
<form method="get" action = "#"> <!-- ----------------------FIRST FORM-------------------------
----- -->
<tr>
<td>Ειδικότητα:</td><td>
<select name="specialtyselect">
<%
while(rs.next())
{
String fname = rs.getString("specialty");
%>
<option value="<%=fname %>"><%=fname %></option>
<%
}
%>
</select>
</td>

<%
rs.close();
con.close();
}
catch(SQLException sqe)
{
out.println(sqe);
}
%>
<td>
<input type="submit" name="submitspecialty" value="Επιλογή ειδικότητας"></td></tr>
</form>
<form action="/doctors_online/PatientServlet" method="Post"><!-- ----------SECOND FORM---------
---- -->

<%
String s=request.getParameter("specialtyselect");
try
{
Connection con1 = datasource.getConnection();
```

```
String sql1 = "SELECT a.id,a.doctor,a.date,a.time FROM appointments a inner join doctors d on
a.doctor=d.username where d.specialty = ? and a.patient='null' ";
PreparedStatement ps1 = con1.prepareStatement(sql1);

ps1.setString(1, s);
ResultSet rs1 = ps1.executeQuery();
if(rs1.next()){
      rs1.previous();
%><tr><td>
Διαλέξτε ώρα:</td><td>
<select name="appointselect">
<%

while(rs1.next())
{
String id = rs1.getString("id");
String doctor = rs1.getString("doctor");
String date = rs1.getString("date");
String time = rs1.getString("time");
%>
<option value="<%=id %>"> Νο <%=id %>: <%=date %> και ώρα <%=time %>, με τον/την <%=doctor %>
</option>
<%
}
%>
</select></td></tr><tr><td>
<input type="submit" name="chose" value="Ραντεβού με γιατρό ειδικότητας"></td>
<%
}
else
{
      if(s==null)
      {

%>
<td>Δεν έχετε επιλέξει<br> ειδικότητα</td>
<%}else{ %>
<td>Δεν υπάρχουν διαθέσιμα<br> ραντεβού με <%=s %></td>
<%}
}
rs1.close();
con1.close();
}
catch(SQLException sqe)
{
out.println(sqe);
}
%>
</tr>

<tr>
    <td> </td>

</tr>
</form>
```

```
<form method="get" action = "#"><!-- --------------------THIRD FORM-----------------------------
-- -->
<tr><td>Όνομα Χρήστη Γιατρού:</td><td> <input type="text" name="byname"/></td>
<td>
<input type="submit" name="submitdoctor" value="Επιλογή γιατρού"></td></tr>
</form>
<form  action="/doctors_online/PatientServlet" method="Post"><!-- ------------FOURTH FORM------
---- -->

<%
String n=request.getParameter("byname");
try
{
Connection con2 = datasource.getConnection();
String sql2 = "SELECT id,date,time FROM appointments where doctor = ? and patient='null' ";
PreparedStatement ps2 = con2.prepareStatement(sql2);

ps2.setString(1, n);
ResultSet rs2 = ps2.executeQuery();
if(rs2.next()){
      rs2.previous();
%>
<tr><td>
Διαλέξτε ώρα:</td><td>
<select name="appointselect1">
<%

while(rs2.next())
{
String id = rs2.getString("id");
String date = rs2.getString("date");
String time = rs2.getString("time");
%>
<option value="<%=id %>"> No <%=id %>: <%=date %> και ώρα <%=time %></option>
<%
}
%>
</select></td></tr><tr><td>
<input type="submit" name="chose" value="Ραντεβού με γιατρό"></td>
<%
}
else
{
      if(n==""|| n==null)
      {
%>
<td>Δεν έχετε εισάγει<br> όνομα γιατρού</td>
<%}else{ %>
<td>Δεν υπάρχουν διαθέσιμα<br> ραντεβού με τον/την <%=n %></td>
<%}
}
rs2.close();
con2.close();
}
catch(SQLException sqe)
{
```

```
out.println(sqe);
}
%>
<tr>
    <td> </td>

</tr>

<tr><td>ID ραντεβού: </td><td><input type="text" name="apid"/> </td></tr>
<tr>
<td><input type="submit" name="chose" value="Ακύρωση ραντεβού"></td></tr>

<tr>
    <td> </td>

</tr>
<tr><td><input type="submit" name="chose" value="Τα ραντεβού μου"></td></tr>
<tr><td><input type="submit" name="chose" value="Τα στοιχεία μου"></td></tr>
<tr><td></td><td></td><td><input type = "submit" name = "chose" value="Αποσύνδεση"
style="float:right;"></td></tr>
</form>
</table>
</body>
</html>
```

2.3 Κώδικας κλάσεων

2.3.1 Admin

```
package classes;

public class Admin extends Users
{
	//________CONSTRUCTOR________//
	public Admin(String username, String password, String name, String surname)
	{
		super(username, password, name, surname);
	}
	//________METHODS________//
	public String newDoctor() // eisagwgi neu giatru sto systima
	{

		String insertDoctorStatement = "INSERT INTO doctors (specialty, username,
password, name, surname) VALUES (?,?,?,?,?)";
		return(insertDoctorStatement);
	}
	public String removeDoctor() // diagrafi giatru apo to systima
	{
		String deleteDoctorStatement ="delete from doctors where username = ?";
		return(deleteDoctorStatement);
```

```
	}
	public String showAdminInfo() //emfanisi stoixeiwn admin
	{
		String showAdminStatement = "SELECT * FROM admins WHERE username = ?";
		return(showAdminStatement);
	}
}
```

2.3.2 Doctor

```
package classes;

public class Doctor extends Users
{
	private String specialty;

	//________CONSTRUCTOR________//
	public Doctor(String specialty, String username, String password, String name, String
surname)
	{
		super(username, password, name, surname);
		setSpecialty(specialty);
	}
	//________GETTERS&SETTERS________//
	public String getSpecialty()
	{
		return specialty;
	}
	public void setSpecialty(String specialty)
	{
		this.specialty = specialty;
	}
	//________METHODS________//
	public String setAvailability() // kataxwrisi diathesimotitas iatru ana mina
	{
		String availStm = "INSERT INTO appointments (doctor,patient,date,time) VALUES
(?,'null',?,?)";
		return(availStm);
	}
	public String showAgenda() //provoli programmatos radevu
	{
		String showApStatement = "SELECT * FROM appointments WHERE doctor = ? and patient
!= 'null'";
		return(showApStatement);
	}
	public String cancelAppointment() //akyrwsi radevu
	{
		String cancelStm;

			cancelStm ="delete from appointments where doctor= ? and id = ?";

		return(cancelStm);
	}
```

```
	public String showDoctorInfo() //emfanisi stoixeiwn giatrou
	{
		String showStatement = "SELECT * FROM doctors WHERE username = ?";
		return(showStatement);
	}
}
```

2.3.3 Patient

```
package classes;

public class Patient extends Users
{
	private int AMKA; //didetai 1 fora !!// ...

	//________CONSTRUCTOR________//
	public Patient(int amka, String username, String password, String name, String surname)
	{
		super(username, password, name, surname);
		setAMKA(amka);
	}

	//________GETTERS&SETTERS________//
	public int getAMKA()
	{
		return AMKA;
	}
	public void setAMKA(int AMKA)
	{
		this.AMKA = AMKA;
	}
	//________METHODS________//
	public String register() //eggrafi asthenus
	{

		String insertPatientStatement = "INSERT INTO patients (amka, username, password,
name, surname) VALUES (?,?,?,?,?)";
		return(insertPatientStatement);


	}
	public String appointment() //neo radevu
	{
		String appStm = "update appointments set patient =  ? where id = ?";
		return(appStm);
	}
	public String cancelAppointment() //akyrwsi radevu
	{
		String docAppStm = "update appointments set patient= 'null' where patient= ? and
id = ?";
		return(docAppStm);
	}
	public String showAppointments() //provoli radevu
	{
```

```
		String showAoointmentsStatement = "SELECT * FROM appointments WHERE patient = ?";
		return(showAoointmentsStatement);
	}
	public void showTreatmentHistory()//provoli istoriku radevu
	{
		System.out.println("you have no appointment history yet!\n");
	}
	public String showPatientInfo()//provoli stoixeiwn asthenus
	{
		String showPatientStatement = "SELECT * FROM patients WHERE username = ?";
		return(showPatientStatement);
	}

}
```

2.4 Κώδικας των Servlets

2.4.1 LoginServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.sql.DataSource;

@WebServlet({ "/LoginServlet", "/doctors_online/WebContent/patient.jsp" })
public class LoginServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private DataSource datasource = null;
	public void init() throws ServletException
	{
		try //καθορισμός των στοιχείων για τη βάση
		{
			InitialContext ctx = new InitialContext();
			datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
		}
		catch(Exception e)
		{
			throw new ServletException(e.toString());
		}
	}
    public LoginServlet() {
```

```
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    }
      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
      {
            //η συνάρτηση που εκτελείται με την κλήση του servlet
            response.setContentType("text/html; charset=UTF-8");
            request.setCharacterEncoding("UTF-8");
            response.setCharacterEncoding("UTF-8");

            //μετατροπή σε String των στοιχείων που εισήγαγε ο χρήστης για είσοδο
            String username = request.getParameter("NameField");
            String password = request.getParameter("PasswordField");

            //ανάλογα με ποιο κουμπί έχει πατήσει ο χρήστης κάνει :
            String submit = request.getParameter("submit");
       String usertype,link;
       if ("Είσοδος Ασθενή".equals(submit)){ //είσοδο ως ασθενής
                  usertype = "p";
                  link = "<a href='/doctors_online/patient.jsp'>Πήγαινε στο Home
σου!</a><br>";
            }
       else if ("Είσοδος Ιατρού".equals(submit)){ //είσοδο ως γιατρός
            usertype = "d";
            link = "<a href='/doctors_online/doctor.jsp'>Πήγαινε στο Home σου!</a><br>";
            }
       else { //είσοδο ως διαχειριστής
            usertype = "a";
            link = "<a href='/doctors_online/admin.jsp'>Πήγαινε στο Home σου!</a><br>";
       }
            if(checkUser(username,password,usertype,request)) //αν ο χρήστης υπάρχει στη βάση
            {
                  //δημιουργία http session
                  HttpSession session = request.getSession();
                  String name = request.getParameter("NameField");
                  session.setAttribute("user", name); //δημιουργία attribute που συγκρατεί
                  //το όνομα χρήστη που συνδέθηκε

                  String keep = (String) session.getAttribute("user");//attribute to String

                  createDynamicPage("Καλωσήρθες " +keep,link, response);//εμφάνιση χαιρετισμού

            }//αν ο χρήστης πληκτρολόγησε λάθος στοιχεία για σύνδεση
            else { createDynamicPage("Τα στοιχεία δεν είναι σωστά!","<a
href=\"/doctors_online/index.html\">Πήγαινε στο Home!</a><br>", response); }
      }
      private void createDynamicPage(String message,String link, HttpServletResponse response)
throws IOException
      {// δημιουργία dynamic web page η οποία εμφανίζει κάποιο μήνυμα

            response.setContentType("text/html; charset=UTF-8");
            response.setCharacterEncoding("UTF-8");
            response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
```

```
            response.setHeader("Pragma","no-cache"); //HTTP 1.0
            response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

            PrintWriter out = response.getWriter();
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");

            //εμφάνιση μηνύματος
            out.println("<html>");
            out.println("<head><title>Message</title></head>");
            out.println("<body>");
            out.println("<p>" + message + "</p>");

            //hyperlink
            out.println(link);
            out.println("</body></html>");
      }
      private boolean checkUser(String username,String password,String
usertype,HttpServletRequest request)
    {//συνάρτηση τύπου String που ελέγχει αν τα στοιχεία που εισήγαγε ο χρήστης είναι σωστά
            //και υπάρχει στη βάση

             boolean st =false;
          try{//σύνδεση με τη βάση και εκτέλεση statement
            Connection con= datasource.getConnection();
            String query=null;
            //ανάλογα με τον τύπο χρήστη το query αλλάζει
            if (usertype == "p"){
                        query ="select * from patients where username=? and password=?";
                  }
            else if (usertype == "d"){
                  query ="select * from doctors where username=? and password=?";
                  }
            else {
                  query ="select * from admins where username=? and password=?";
            }
            PreparedStatement ps =con.prepareStatement(query);
            ps.setString(1, username);
            ps.setString(2, password);
            ResultSet rs =ps.executeQuery();//εκτέλεση query
            st = rs.next();
          }catch(Exception e)
          {
              e.printStackTrace();
          }
            return st;  //επιστροφή αν υπάρχει ή οχι ο χρήστης (true , false)
void createNullUser()
      {
            try
            {//σύνδεση με τη βάση

                  Connection c = datasource.getConnection();

                   String query = "INSERT INTO patients (amka,username,password,name,surname)"
                               +" VALUES (0, 'null', 'null', 'null', 'null')"
```

```
                                    + "WHERE NOT EXISTS (SELECT username FROM patients WHERE
username='null');";
                        PreparedStatement s = c.prepareStatement(query);

                    int exe = s.executeUpdate();

                    s.close();
                        c.close();
                }
                catch(SQLException sqle)
                {
                        sqle.printStackTrace();
                }
        }
    }
}
```

2.4.2 RegisterDoctorServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

import classes.Admin;

@WebServlet("/RegisterDoctorServlet")
public class RegisterDoctorServlet extends HttpServlet {
       private static final long serialVersionUID = 1L;


       private DataSource datasource = null;

       public void init() throws ServletException
       {

              try //καθορισμός στοιχείων για τη βάση
              {
                     InitialContext ctx = new InitialContext();
                     datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
              }
              catch(Exception e)
              {
```

```
                throw new ServletException(e.toString());
            }

       }

    public RegisterDoctorServlet() {
        super();
    }

       protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            response.setContentType("text/html; charset=UTF-8");
            request.setCharacterEncoding("UTF-8");
            response.setCharacterEncoding("UTF-8");

            //μετατροπή σε String των στοιχείων που εισήγαγε ο χρήστης
            String specialty = request.getParameter("specialty");
            String username = request.getParameter("username");
            String password = request.getParameter("password");
            String name = request.getParameter("name");
            String surname = request.getParameter("surname");

            if(specialty!="" && username!="" && password!="" && name!="" && surname !="")
            {//αν όλα τα πεδία είναι συμπληρωμένα
                  try
                  {//σύνδεση με τη βάση

                        Connection con = datasource.getConnection();

                        Admin a = new Admin("","","","");

                        //εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
                        //Admin, newDoctor(), για εισαγωγή του χρήστη στη βάση
                        PreparedStatement insertDoctor = con.prepareStatement(a.newDoctor());
                        insertDoctor.setString(1,specialty);
                        insertDoctor.setString(2, username);
                        insertDoctor.setString(3, password);
                        insertDoctor.setString(4, name);
                        insertDoctor.setString(5, surname);


                     int exe = insertDoctor.executeUpdate();
                     if(exe>0)
                     {//αν το statement εκτελέστηκε σωστά
                        createDynamicPage("Συγχαρητήρια, ο γιατρός εισήχθη επιτυχώς στο
σύστημα !",response);
                     }
                     else
                     {//αν κάτι πήγε στραβά
                        createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά!",response);
                     }
                     insertDoctor.close();
                        con.close();
                  }
                  catch(SQLException sqle)
                  {
```

```
                        sqle.printStackTrace();
                    }

                }
                else
                {//αν δεν έχουν συμπληρωθεί όλα τα πεδία
                    createDynamicPage("Δεν συμπληρώσατε όλα τα πεδία",response);
                }

        }

        private void createDynamicPage(String message, HttpServletResponse response) throws
IOException
        {//συνάρτηση δημιουργίας dynamic web page η οποία καλείται σε διάφορα σημεία του κώδικα
            //για την εμφάνιση του κατάλληλου μηνύματος κάθε φορά
            response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
            response.setHeader("Pragma","no-cache"); //HTTP 1.0
            response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

            response.setContentType("text/html; charset=UTF-8");
            response.setCharacterEncoding("UTF-8");
            PrintWriter out = response.getWriter();
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");

            out.println("<html>");
            out.println("<head><title>Register</title></head>");
            out.println("<body>");
            out.println("<p>" + message + "</p>");//εμφάνιση μηνύματος
            out.println("<a href=\"/doctors_online/admin.jsp\">Πίσω</a><br>");
            out.println("</body></html>");
        }

}
```

2.4.3 RegisterPatientServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;
import classes.Patient;
```

```
@WebServlet({"/RegisterPatientServlet", "/doctors_online/WebContent/insert_patient.html" })
public class RegisterPatientServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private DataSource datasource = null;
	public void init() throws ServletException
	{
		try //καθορισμός στοιχείων για τη βάση
		{
			InitialContext ctx = new InitialContext();
			datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
		}
		catch(Exception e)
		{
			throw new ServletException(e.toString());
		}
	}
    public RegisterPatientServlet()
    {
       super();
    }
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
	{
		response.setContentType("text/html; charset=UTF-8");
		request.setCharacterEncoding("UTF-8");
		response.setCharacterEncoding("UTF-8");

		//μετατροπή σε String τω στοιχείων που εισήγαγε ο χρήστης
		String amka = request.getParameter("amka");
		String username = request.getParameter("username");
		String password = request.getParameter("password");
		String name = request.getParameter("name");
		String surname = request.getParameter("surname");

		if(amka!="" && username!="" && password!="" && name!="" && surname !="")
		{//αν όλα τα πεδία είναι συμπληρωμένα

			int iex1 = 0;
			try //έλεγχος για integer στο πεδίο amka
			{
				iex1 = Integer.parseInt(amka);
			}
			catch(Exception e)
			{
				e.printStackTrace();
			}
			try
			{//σύνδεση με τη βάση

				Connection con = datasource.getConnection();

				Patient p = new Patient(iex1,username,password,name,surname);

				//εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
				//patient, register(), για εισαγωγή του χρήστη στη βάση
```

```
                        PreparedStatement insertPatient = con.prepareStatement(p.register());
                        insertPatient.setInt(1,Integer.parseInt(amka));
                        insertPatient.setString(2, username);
                        insertPatient.setString(3, password);
                        insertPatient.setString(4, name);
                    insertPatient.setString(5, surname);


                    int exe = insertPatient.executeUpdate();
                    if(exe>0)
                    {//αν το statement εκτελέστηκε σωστά
                        createDynamicPage("Συγχαρητήρια, συνδεθήκατε επιτυχώς!",response);
                    }
                    else
                    {//αν κάτι πήγε στραβά
                        createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά με τη σύνδεσή
σας!",response);
                    }
                    insertPatient.close();
                        con.close();
                }
                catch(SQLException sqle)
                {
                        sqle.printStackTrace();
                }
            }
            else
            {//αν δεν έχουν συμπληρωθεί όλα τα πεδία
                createDynamicPage("Δεν συμπληρώσατε όλα τα πεδία",response);
            }
        }
        private void createDynamicPage(String message, HttpServletResponse response) throws
IOException
        {//συνάρτηση δημιουργίας dynamic web page η οποία καλείται σε διάφορα σημεία του κώδικα
            //για την εμφάνιση του κατάλληλου μηνύματος κάθε φορά
            response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
            response.setHeader("Pragma","no-cache"); //HTTP 1.0
            response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

            response.setContentType("text/html; charset=UTF-8");
            response.setCharacterEncoding("UTF-8");

            PrintWriter out = response.getWriter();
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");

            out.println("<html>");
            out.println("<head><title>Register</title></head>");
            out.println("<body>");
            out.println("<p>" + message + "</p>");
            out.println("<a href=\"/doctors_online/index.html\">Επιστροφή στην αρχική
σελίδα</a>");
            out.println("</body></html>");
        }
```

```
}
```

2.4.4 AdminServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;
import classes.Admin;

@WebServlet("/AdminServlet")
public class AdminServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private DataSource datasource = null;
	public void init() throws ServletException
	{
		try //καθορισμός των στοιχείων για τη βάση
		{
			InitialContext ctx = new InitialContext();
			datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
		}
		catch(Exception e)
		{
			throw new ServletException(e.toString());
		}
	}
    public AdminServlet() {
        super();
    }
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		response.setContentType("text/html; charset=UTF-8");
		request.setCharacterEncoding("UTF-8");
		response.setCharacterEncoding("UTF-8");

		String keep = (String)request.getSession().getAttribute("user");//ανάκτηση
ονόματος χρήστη από το session
		String menu = request.getParameter("admin"); //η επιλογή του διαχειριστή σε String
	//ανάλογα με την επιλογή εκτελείται και διαφορετικό κομμάτι κώδικα
	if ("Εισαγωγή νέου ιατρού στο σύστημα".equals(menu)){
			response.sendRedirect("insert_doctor.html"); //οδήγηση στη σελίδα για
register γιατρού
		}
```

```
        else if ("Διαγραφή ιατρού από το σύστημα".equals(menu))
        {
              String doc = request.getParameter("doctor");
                    if(doc!= "") //αν το πεδίο δεν είναι κενό διαγραφή του γιατρού που εισήγαγε
ο διαχειριστής
                    {
                    try
                    {//σύνδεση με τη βάση
                          Connection con = datasource.getConnection();

                          Admin a = new Admin("","","","");
                          //εκτέλεση statement με τη βοήθεια της συνάρτησης removeDoctor()
                          //της κλάσης Admin
                          PreparedStatement DeleteDoctor =
con.prepareStatement(a.removeDoctor());
                        DeleteDoctor.setString(1, doc);
                        int exe = DeleteDoctor.executeUpdate();
                        if(exe>0)
                        {//αν το statement εκτελέστηκε σωστά τύπωση μηνύματος
                          createDynamicPage("Ο ιατρός διαγράφηκε επιτυχώς!",response);
                          response.sendRedirect("admin.jsp");
                        }
                        else
                        {//αν κάτι πήγε στραβά τύπωση μηνύματος
                          createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά!",response);
                        }
                          con.close();
                    }
                    catch(SQLException sqle)
                    {
                          sqle.printStackTrace();
                    }
                    }
                    else
                    {
                          createDynamicPage("Δεν υπάρχει ιατρός με αυτό το όνομα
χρήστη",response);
                    }
              }
        else if ("Τα στοιχεία μου".equals(menu))//αν έχει επιλεγεί η εμφάνιση των στοιχείων του
διαχειριστή
        {
              showInfo(keep, response);
        }
        else
        {
              request.getSession().removeAttribute("user");
                    request.getSession().invalidate();
                    response.sendRedirect("index.html");
        }
      }
      private void showInfo(String username, HttpServletResponse response) throws IOException
      {// δημιουργία dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη
              response.setContentType("text/html; charset=UTF-8");
              response.setCharacterEncoding("UTF-8");
```

```
            response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
            response.setHeader("Pragma", "no-cache"); // HTTP 1.0.

            PrintWriter out = response.getWriter();
            if (username == null) {response.sendRedirect("index.html");}
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");//css
            try
            {//σύνδεση με τη βάση

                  Connection con1 = datasource.getConnection();
                  //δημιουργία instance της κλάσης patient
                  Admin a1 = new Admin("","","","");

                  //εκτέλεση του statement που βρίσκεται στην συνάρτηση showPatientInfo()
                  //της κλάσης patient
                  PreparedStatement ps1 = con1.prepareStatement(a1.showAdminInfo());
                  ps1.setString(1, username);
                  ResultSet rs1 =ps1.executeQuery();

                  if (rs1.next()) {//εκτυπώνονται σε πίνακα τα στοιχεία του διαχειριστή
                      out.print("<table><tr><th>Όνομα
Χρήστη</th><th>Κωδικός</th><th>Όνομα</th><th>Επώνυμο</th></tr>");
                      do {
                          out.print("<tr>");
                          out.print(new
StringBuilder("<td>").append(rs1.getObject("username")).append("</td>").toString());
                          out.print(new
StringBuilder("<td>").append(rs1.getObject("password")).append("</td>").toString());
                          out.print(new
StringBuilder("<td>").append(rs1.getObject("name")).append("</td>").toString());
                          out.print(new
StringBuilder("<td>").append(rs1.getObject("surname")).append("</td>").toString());

                          out.print("</tr>");
                      } while (rs1.next());
                      out.print("</table>");
                      out.println("<a href=\"/doctors_online/admin.jsp\">Πίσω</a>");//link
                      ps1.close();
                         con1.close();
                  }
            }
            catch(SQLException sqle) //exception
            {
                  sqle.printStackTrace();
            }
      }
      private void createDynamicPage(String message, HttpServletResponse response) throws
IOException
      {// δημιουργία dynamic web page η οποία εμφανίζει κάποιο μήνυμα
            response.setContentType("text/html; charset=UTF-8");
            response.setCharacterEncoding("UTF-8");
            response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
            response.setHeader("Pragma","no-cache"); //HTTP 1.0
```

```
            response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

            PrintWriter out = response.getWriter();
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");

            //εμφάνιση μηνύματος
            out.println("<html>");
            out.println("<head><title>Message</title></head>");
            out.println("<body>");
            out.println("<p>" + message + "</p>");

            //hyperlink
            out.println("<a href=\"/doctors_online/admin.jsp\">Πίσω</a><br>");
            out.println("</body></html>");
      }
}
```

2.4.5 DoctorServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

import classes.Doctor;

@WebServlet("/DoctorServlet")
public class DoctorServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;
      private DataSource datasource = null;
      public void init() throws ServletException
      {
            try //καθορισμός των στοιχείων για τη βάση
            {
                  InitialContext ctx = new InitialContext();
                  datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
            }
            catch(Exception e)
            {
                  throw new ServletException(e.toString());
            }
```

```
    }
    public DoctorServlet() {
        super();
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            response.setContentType("text/html; charset=UTF-8");
            request.setCharacterEncoding("UTF-8");
            response.setCharacterEncoding("UTF-8");

            String keep = (String)request.getSession().getAttribute("user");//ανάκτηση
ονόματος χρήστη του session

            String menu = request.getParameter("doctor"); //επιλογή του χρήστη σε String
        //ανάλογα με την επιλογή εκτελείται διαφορετικό κομμάτι κώδικα
        if ("Δήλωση διαθεσιμότητας".equals(menu)){
                availability(keep, request,response);
            }
        else if ("Ακύρωση ραντεβού".equals(menu))
        {
            cancel(keep,request,response);
            }
        else if ("Τα ραντεβού μου".equals(menu)){
            showAppointments(keep,response);
            }
        else if ("Τα στοιχεία μου".equals(menu))
        {
            showInfo(keep,response);
        }
        else
        {
                request.getSession().removeAttribute("user");
                request.getSession().invalidate();
                response.sendRedirect("index.html");
        }
      }
      void availability (String user, HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
      { //συνάρτηση που εισάγει τη διαθέσιμη ημερομηνία και ώρα που εισήγαγε ο γιατρός
            //στον πίνακα appointments με  ασθενή = "null" ( αργότερα η εγγραφή θα γίνει
ραντεβού και
            //η τιμή "null" θα αντικατασταθεί από το όνομα χρήστη του ασθενή)

            String date = request.getParameter("Ημερομηνία");//ημ/νία και ώρα που εισήγαγε ο
γιατρός
            String time = request.getParameter("Ώρα");
            try
            {//σύνδεση με τη βάση

                Connection con = datasource.getConnection();
                Doctor d = new Doctor("","","","","");

                //εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
                //doctor, setAvailability(), για εισαγωγή της διαθέσιμης ώρας στη βάση

                PreparedStatement norcStm = con.prepareStatement(d.setAvailability());
```

```
                    norcStm.setString(1,user);
                    norcStm.setString(2, date);
                    norcStm.setString(3, time);
                int exe = norcStm.executeUpdate();
                if(exe>0)
                {//αν το statement εκτελέστηκε σωστά
                    createDynamicPage("Η διαθέσιμη ώρα εισήχθη επιτυχώς",response);
                }
                else
                {//αν κάτι πήγε στραβά
                    createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά με τη σύνδεσή
σας!",response);
                }
                norcStm.close();
                    con.close();
            }
            catch(SQLException sqle)
            {
                    sqle.printStackTrace();
            }
      }
      void cancel (String user, HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
      { //συνάρτηση η οποία χρησιμοποιείται για ακύρωση ραντεβού
            response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
            response.setHeader("Pragma","no-cache"); //HTTP 1.0
            response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

            String idr = request.getParameter("IDραντεβού"); //ανάκτηση τιμής που εισήγαγε ο
χρήστης
            try
            {//σύνδεση με τη βάση
                    Connection con = datasource.getConnection();
                    Doctor d = new Doctor("","","","","");

                    //εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
                    //doctor, cancelAppointment(), για διαγραφή του ραντεβού
                    PreparedStatement norcStm = con.prepareStatement(d.cancelAppointment());
                    norcStm.setString(1,user);
                    norcStm.setString(2, idr);
                int exe = norcStm.executeUpdate();
                if(exe>0)
                {//αν το statement εκτελέστηκε σωστά
                    createDynamicPage("Το ραντεβού ακυρώθηκε επιτυχώς",response);
                }
                else
                {//αν κάτι πήγε στραβά
                    createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά με τη σύνδεσή
σας!",response);
                }
                norcStm.close();
                    con.close();
            }
            catch(SQLException sqle)
            {
                    sqle.printStackTrace();
```

```
			}
		}
		private void createDynamicPage(String message, HttpServletResponse response) throws
IOException
		{//συνάρτηση δημιουργίας dynamic web page η οποία καλείται σε διάφορα σημεία του κώδικα
			//για την εμφάνιση του κατάλληλου μηνύματος κάθε φορά
			response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
			response.setHeader("Pragma","no-cache"); //HTTP 1.0
			response.setDateHeader ("Expires", 0); //prevents caching at the proxy server

			response.setContentType("text/html; charset=UTF-8");
			response.setCharacterEncoding("UTF-8");

			PrintWriter out = response.getWriter();
			out.print("<div><h1>DOCTORS ONLINE</h1></div>");
			out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");
			out.println("<html>");
			out.println("<head><title>Register</title></head>");
			out.println("<body>");
			out.println("<p>" + message + "</p>"); //εμφάνιση μηνύματος
			out.println("<a href=\"/doctors_online/doctor.jsp\">Πίσω</a>");
			out.println("</body></html>");
		}
		private void showAppointments(String username, HttpServletResponse response) throws
IOException
		{// δημιουργία dynamic web page η οποία εμφανίζει τα ραντεβού του συνδεδεμένου χρήστη

			response.setContentType("text/html; charset=UTF-8");
			response.setCharacterEncoding("UTF-8");
			response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
			response.setHeader("Pragma", "no-cache"); // HTTP 1.0.

			if (username == null) {response.sendRedirect("index.html");}
			PrintWriter out = response.getWriter();
			out.print("<div><h1>DOCTORS ONLINE</h1></div>");
			out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");
			try
			{//σύνδεση με τη βάση
				Connection con = datasource.getConnection();
				Doctor d = new Doctor("","","","","");
				//sql statement που χρησιμοποιεί το String της κλάσης Doctor της συνάρτησης
showAgenda()
				//για εμφάνιση των ραντεβού
				PreparedStatement ps = con.prepareStatement(d.showAgenda());
				ps.setString(1, username);
				ResultSet rs =ps.executeQuery();
				if (rs.next()) {
					//αν υπάρχουν ραντεβού τυπώνονται σε ένα πίνακα

out.print("<table><tr><th>ID</th><th>Ασθενής</th><th>Ημερομηνία</th><th>Ώρα</th></tr>");
					do {
					    out.print("<tr>");
```

```
                        out.print(new
StringBuilder("<td>").append(rs.getObject("id")).append("</td>").toString());
                        out.print(new
StringBuilder("<td>").append(rs.getObject("patient")).append("</td>").toString());
                        out.print(new
StringBuilder("<td>").append(rs.getObject("date")).append("</td>").toString());
                        out.print(new
StringBuilder("<td>").append(rs.getObject("time")).append("</td>").toString());
                        out.print("</tr>");
                    } while (rs.next());
                    out.print("</table>");
                }
                else
                {//αν δεν υπάρχουν ραντεβού τυπώνεται μήνυμα
                      out.print("Δεν έχετε προγραμματισμένα ραντεβού");
                }
               //hyperlinks που επιστρέφουν στις προηγούμενες σελίδες
               out.println("<br><br><a href=\"/doctors_online/doctor.jsp\">Πίσω</a>");
               ps.close();
                  con.close();//τερματισμός σύνδεσης με τη βάση
            }
            catch(SQLException sqle) //αν δεν γίνει σωστά η σύνδεση με τη βάση
            {
                  sqle.printStackTrace();
            }
      }
      private void showInfo(String username, HttpServletResponse response) throws IOException
      {// δημιουργία dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη

            response.setContentType("text/html; charset=UTF-8");
            response.setCharacterEncoding("UTF-8");
            response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
            response.setHeader("Pragma", "no-cache"); // HTTP 1.0.

            if (username == null) {response.sendRedirect("index.html");} //αν το username
είναι null
            //ο χρήστης δεν είναι συνδεδεμένος άρα οδηγείται στο index.html
            PrintWriter out = response.getWriter();
            out.print("<div><h1>DOCTORS ONLINE</h1></div>");
            out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");//css
            try
            {//σύνδεση με τη βάση
                  Connection con1 = datasource.getConnection();
                  //δημιουργία instance της κλάσης doctor
                  Doctor d1 = new Doctor("","","","","");

                  //εκτέλεση του statement που βρίσκεται στην συνάρτηση showDoctorInfo()
                  //της κλάσης Doctor
                  PreparedStatement ps1 = con1.prepareStatement(d1.showDoctorInfo());
                  ps1.setString(1, username);
                  ResultSet rs1 =ps1.executeQuery();
                  if (rs1.next()) {//εκτυπώνονται σε πίνακα τα στοιχεία του γιατρού
                      out.print("<table><tr><th>Ειδικότητα</th><th>Όνομα
Χρήστη</th><th>Κωδικός</th><th>Όνομα</th><th>Επώνυμο</th></tr>");
```

```
                        do {
                            out.print("<tr>");
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("specialty")).append("</td>").toString());
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("username")).append("</td>").toString());
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("password")).append("</td>").toString());
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("name")).append("</td>").toString());
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("surname")).append("</td>").toString());

                            out.print("</tr>");
                        } while (rs1.next());
                        out.print("</table>");
                        out.println("<a href=\"/doctors_online/doctor.jsp\">Πίσω</a>");//link
                        ps1.close();
                           con1.close();
                    }
                }
                catch(SQLException sqle) //exception
                {
                      sqle.printStackTrace();
                }
        }
}
```

2.4.6 PatientServlet.java

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;
import classes.Patient;

@WebServlet({ "/PatientServlet"})
public class PatientServlet extends HttpServlet {
       private static final long serialVersionUID = 1L;

       private DataSource datasource = null;
```

```
	public void init() throws ServletException
	{
		try //καθορισμός των στοιχείων για τη βάση
		{
			InitialContext ctx = new InitialContext();
			datasource = (DataSource)ctx.lookup("java:comp/env/jdbc/LiveDataSource");
		}
		catch(Exception e)
		{
			throw new ServletException(e.toString());
		}
	}
    public PatientServlet() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    }
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
	// συνάρτηση για τη λειτουργία post του servlet

		response.setContentType("text/html; charset=UTF-8");
		request.setCharacterEncoding("UTF-8");
		response.setCharacterEncoding("UTF-8");

		String keep = (String)request.getSession().getAttribute("user");//ανάκτηση
ονόματος χρήστη από το session

		String menu = request.getParameter("chose");
		//menu: επιλογή του χρήστη (ένα από τα κουμπιά τύπου submit με όνομα chose)
		//ανάλογα με το κουμπί που πάτησε ο ασθενής καλείται κάποια συνάρτηση για να
		//ολοκληρωθεί η ενέργεια
	  if ("Ραντεβού με γιατρό ειδικότητας".equals(menu))
	  {
			appointment(keep,"specialty",request,response);
		}
	  else if ("Ραντεβού με γιατρό".equals(menu))
	  {
		appointment(keep,"name",request,response);
		}

	  else if  ("Ακύρωση ραντεβού".equals(menu))
	  {
		appointment(keep,"delete",request,response);
		}
	  else if ("Τα ραντεβού μου".equals(menu)){
		showAppointments(keep,request,response);
		}
	  else if ("Τα στοιχεία μου".equals(menu))
	  {
		showInfo(keep,response);
	  }
	  else //αν έχει πατηθεί το κουμπί για logout
	  {
```

```
                    request.getSession().removeAttribute("user");
                    request.getSession().invalidate();
                    response.sendRedirect("index.html");
              }
        }
        private void createDynamicPage(String message, HttpServletResponse response) throws
IOException
        {// δημιουργία dynamic web page η οποία εμφανίζει κάποιο μήνυμα

              response.setContentType("text/html; charset=UTF-8");
              response.setCharacterEncoding("UTF-8");
              PrintWriter out = response.getWriter();
              out.print("<div><h1>DOCTORS ONLINE</h1></div>");
              out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");

              //εμφάνιση μηνύματος
              out.println("<html>");
              out.println("<head><title>Message</title></head>");
              out.println("<body>");
              out.println("<p>" + message + "</p>");

              response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
              response.setHeader("Pragma","no-cache"); //HTTP 1.0
              response.setDateHeader ("Expires", 0); //prevents caching at the proxy server
              out.println("<br><br><br><a href=\"/doctors_online/patient.jsp\">Πίσω</a>");
              out.println("</body></html>");
        }
        private void showAppointments(String username, HttpServletRequest request,
HttpServletResponse response) throws IOException
        {// δημιουργία dynamic web page η οποία εμφανίζει τα ραντεβού του συνδεδεμένου χρήστη

              response.setContentType("text/html; charset=UTF-8");
              response.setCharacterEncoding("UTF-8");
              response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
              response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
              if (username == null) {response.sendRedirect("index.html");}

              PrintWriter out = response.getWriter();
              out.print("<div><h1>DOCTORS ONLINE</h1></div>");
              out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");//css

              try
              {//σύνδεση με τη βάση

                    Connection con = datasource.getConnection();
                    Patient p = new Patient(0,"","","","");
                    PreparedStatement ps = con.prepareStatement(p.showAppointments());
                    ps.setString(1, username);
                    ResultSet rs =ps.executeQuery();

                    if (rs.next())
                    {
                          //τυπώνεται ένας πίνακας με τα ραντεβού του ασθενή
```

```
out.print("<table><tr><th>ID</th><th>Γιατρός</th><th>Ημερομηνία</th><th>Ώρα</th></tr>");
                        do {

                            out.print("<tr>");
                            out.print(new
StringBuilder("<td>").append(rs.getObject("id")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs.getObject("doctor")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs.getObject("date")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs.getObject("time")).append("</td>").toString());

                            out.print("</tr>");
                        } while (rs.next());

                        out.print("</table>");
                    }
                       else
                       {
                              out.print("Δεν έχετε κανένα ραντεβού");
                       }
                    //hyperlinks που επιστρέφουν στις προηγούμενες σελίδες
                    out.println("<br><br><br><a href=\"/doctors_online/patient.jsp\">Πίσω</a>");
                    ps.close();
                       con.close();//τερματισμός σύνδεσης με τη βάση
                }
                catch(SQLException sqle) //αν δεν γίνει σωστά η σύνδεση με τη βάση
                {
                       sqle.printStackTrace();
                }

        }
        private void showInfo(String username, HttpServletResponse response) throws IOException
        {// δημιουργία dynamic web page η οποία εμφανίζει τα στοιχεία του συνδεδεμένου χρήστη

                response.setContentType("text/html; charset=UTF-8");
                response.setCharacterEncoding("UTF-8");
                response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
                response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
                if (username == null) {response.sendRedirect("index.html");}

                PrintWriter out = response.getWriter();
                out.print("<div><h1>DOCTORS ONLINE</h1></div>");
                out.println("<link rel='stylesheet' type='text/css' href='/doctors_online/css.css'
/>");//css
                try
                {//σύνδεση με τη βάση

                       Connection con1 = datasource.getConnection();
```

```
                    //δημιουργία instance της κλάσης patient
                    Patient p = new Patient(0,"","","","");

                    //εκτέλεση του statement που βρίσκεται στην συνάρτηση showPatientInfo()
                    //της κλάσης patient
                    PreparedStatement ps1 = con1.prepareStatement(p.showPatientInfo());
                    ps1.setString(1, username);
                    ResultSet rs1 =ps1.executeQuery();

                    if (rs1.next())
                    {//εκτυπώνονται σε πίνακα τα στοιχεία του ασθενή
                        out.print("<table><tr><th>AMKA</th><th>Όνομα
Χρήστη</th><th>Κωδικός</th><th>Όνομα</th><th>Επώνυμο</th></tr>");

                        do {
                            out.print("<tr>");
                            out.print(new
StringBuilder("<td>").append(rs1.getObject("amka")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs1.getObject("username")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs1.getObject("password")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs1.getObject("name")).append("</td>").toString());

                            out.print(new
StringBuilder("<td>").append(rs1.getObject("surname")).append("</td>").toString());

                            out.print("</tr>");
                        } while (rs1.next());

                        out.print("</table>");
                        out.println("<a href=\"/doctors_online/patient.jsp\">Πίσω</a>");//link
                        ps1.close();
                           con1.close();
                    }
                }
                catch(SQLException sqle) //exception
                {
                        sqle.printStackTrace();
                }
        }
        private void appointment(String username, String action,HttpServletRequest request,
HttpServletResponse response) throws IOException
        {//συνάρτηση που εισάγει ή διαγράφει ραντεβού ανάλογα με την τιμή του action
(specialty,name,delete)

                if(action=="specialty"||action=="name")
                {
                        String apsel1="";
                        if (action == "specialty" )//νέο ραντεβού με ειδικότητα - επιλέχθηκε στο
select
                        {
```

```
                apsel1 = request.getParameter("appointselect");

            }
            else if (action == "name")//νέο ραντεβού με γιατρό-επιλέχθηκε στο select
            {
                apsel1 = request.getParameter("appointselect1");
            }
            try
            {//σύνδεση με τη βάση

                Connection con = datasource.getConnection();

                Patient p1 = new Patient(0,"","","","");

                //εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
                //patient, appointment(), για το κλείσιμο του ραντεβού με το γιατρό

                PreparedStatement s = con.prepareStatement(p1.appointment());

                s.setString(1,username);
                s.setInt(2,Integer.parseInt(apsel1));

               int exe = s.executeUpdate();
               if(exe>0)
               {//αν το statement εκτελέστηκε σωστά
                  createDynamicPage("Κλείσατε επιτυχώς το ραντεβού",response);
               }
               else
               {//αν κάτι πήγε στραβά
                  createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά!",response);
               }
               s.close();
                  con.close();
            }
            catch(SQLException sqle)
            {
                sqle.printStackTrace();
            }
        }
        else//action=delete/διαγραφή ραντεβού με id που εισήγαγε ο ασθενής στο txt input
        {
            String apid = request.getParameter("apid");
            int iex1 = 0;
            try //έλεγχος για integer στο πεδίο amka
            {
                iex1 = Integer.parseInt(apid);
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
            try
            {//σύνδεση με τη βάση

                Connection con = datasource.getConnection();
```

```
                        Patient p1 = new Patient(0,"","","","");

                        //εκτέλεση statement που υπάρχει στη συνάρτηση της κλάσης
                        //patient, cancelAppointment(), για διαγραφή του ραντεβού
                        PreparedStatement s = con.prepareStatement(p1.cancelAppointment());

                        s.setString(1,username);
                        s.setInt(2,iex1);

                    int exe = s.executeUpdate();
                    if(exe>0)
                    {//αν το statement εκτελέστηκε σωστά
                        createDynamicPage("Το ραντεβού ακυρώθηκε επιτυχώς",response);
                    }
                    else
                    {//αν κάτι πήγε στραβά
                        createDynamicPage("Λυπούμαστε, κάτι πήγε στραβά!",response);
                    }
                    s.close();
                        con.close();
                }
                catch(SQLException sqle)
                {
                        sqle.printStackTrace();
                }
            }
        }
}
```

2.5 CSS

```
@CHARSET "UTF-8";
body {
        background-color: #E1EAF3;
        text-align:center;
        font-size:14px;
        color:#33344C;
        font-family: sans-serif;
        padding-bottom:50px;
}
a{
        text-align:left;
}
a:hover { color: white }
h1{
```

```
color:white;
text-align:left;
margin-left:90px;
}
h2{
	color:#7384A9;
}
th{
	color:#7384A9;
}
table{
	margin: 0px auto;
}
td
{
	text-align:left;
}
select {
    width: 100%;
    border: 1px solid #5476BE;
    background-color: white;
}
div{
	background-image:url("image.jpg");
	background-size:	cover;
    background-repeat:   no-repeat;
    background-position: center center;
    vertical-align: middle;
line-height: 150px;
	height:140px;
margin-bottom:30px;
}

input[type=text],[type=password]{
    border: 1px solid #5476BE;
}
```

3 Βιβλιογραφικές Πηγές

1. https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/201102161747563cexy8tj
2. https://stackoverflow.com/questions/14152621/preventing-user-to-go-back-after-logout-in-jsp
3. http://www.easywayserver.com/jsp/JSP-form.htm