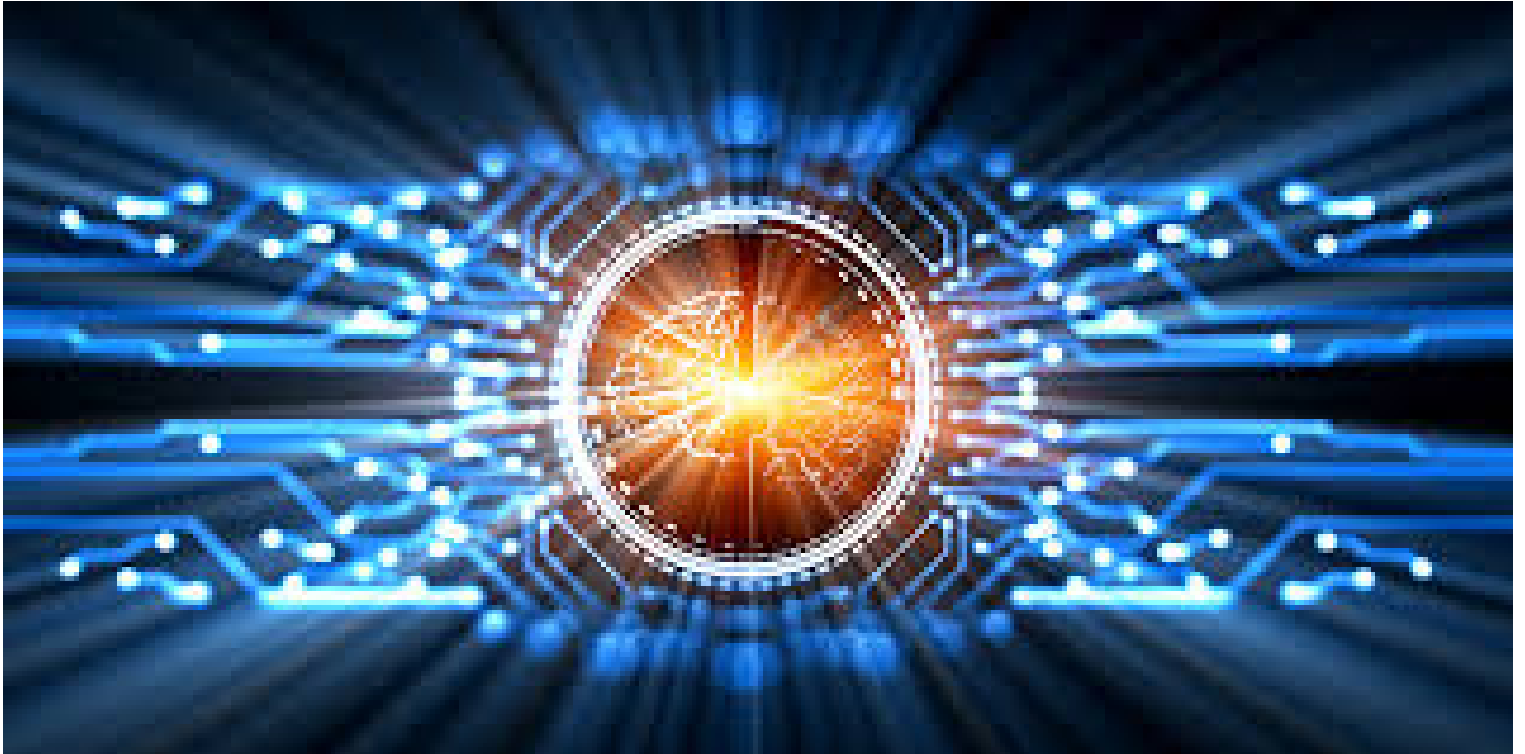


Δικτυακός Προγραμματισμός

Τελική εργασία

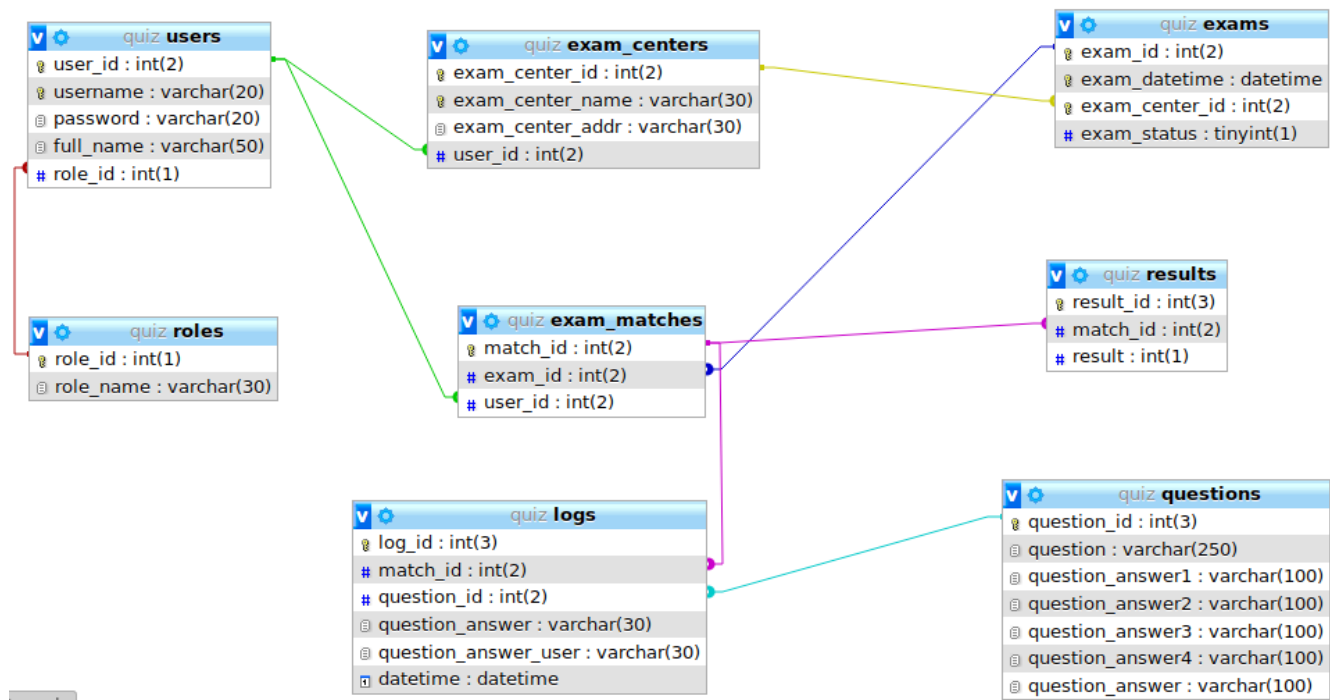


Στοιχεία Φοιτητή: **Ράκας Δημήτρης**
ΑΜ: **131052**

Περίληψη εργασίας

Η εργασία που εκπονήθηκε υλοποιεί τεχνολογίες όπως τα web services σε RESTful μορφή και ένα συνδυασμό από web κομμάτι και desktop. Η πρόσεγγιση που πραγματοποιήθηκε είναι ένα σύστημα εξέτασης γνώσεων για τα σήματα κυκλοφορίας ΚΟΚ στον δρόμο. Υπάρχουν τρεις διακριτοί ρόλοι ο διαχειριστής του συστήματος, ο υπεύθυνος εξεταστικών κέντρων και τέλος οι εξεταζόμενοι που λαμβάνουν μέρος στην εξέταση για τα σήματα. Επιλέγονται 5 τυχαίες ερωτήσεις από την αποθήκη ερωτήσεων που υπάρχει στη βάση δεδομένων και διαγωνίζονται σε αυτές. Μπορούμε να εισαγούμε χρήστες, ερωτήσεις, εξεταστικά κέντρα και εξετάσεις καθώς και να αντιστοιχίσουμε υπεύθυνους με κέντρα και εξεταζόμενους με εξετάσεις. Παρακάτω θα παρουσιαστούν αναλυτικότερα τα επιμέρους κομμάτια που απαρτίζουν την εργασία και screenshots που δείχνουν την υλοποίηση της.

Βάση Δεδομένων



- **users:** Ο πίνακας αυτός περιλαμβάνει όλους τους χρήστες του συστήματος με τα προσωπικά τους στοιχεία και το ρόλο τους στην εφαρμογή.
- **Roles:** Ο πίνακας αυτός περιλαμβάνει τους τρεις διακριτούς ρόλους της εφαρμογής (Διαχειριστής, υπεύθυνος και εξεταζόμενος).
- **exam_centers:** Ο πίνακας αυτός περιλαμβάνει τα εξεταστικά κέντρα που συμμετέχουν στην εξέταση, με τα στοιχεία της κάθε εξέτασης καθώς και τον υπεύθυνο αυτής. Ένας χρήστης τύπου υπεύθυνος μπορεί να είναι υπεύθυνος σε περισσότερα από ένα εξεταστικά κέντρα.
- **exam:** Ο πίνακας αυτός περιλαμβάνει τις διαθέσιμες εξετάσεις που υπάρχουν, στοιχεία για αυτές (ημερομηνία και ώρα), αντιστοιχίση με τα εξεταστικά κέντρα που πραγματοποιείται καθώς και την κατάσταση της (ενεργή/ανενεργή).
- **exam_matches:** Ο πίνακας αυτός περιλαμβάνει αντιστοιχήσεις χρηστών τύπου εξεταζόμενος με την εξέταση που του έχει αποδοθεί. Κάθε εξεταζόμενος μπορεί να συμμετέχει μόνο σε μία εξέταση.
- **results:** Ο πίνακας αυτός περιλαμβάνει τις απαντήσεις των εξεταζομένων στις ερωτήσεις που απάντησαν κατά την διάρκεια της εξέτασης τους.
- **logs:** Ο πίνακας αυτός περιλαμβάνει τις αναφορές του συστήματος με βάση κάποιο στοιχείο αναζήτησης για τις εξετάσεις των εξεταζομένων.
- **questions:** Ο πίνακας αυτός περιλαμβάνει την αποθήκη ερωτήσεων που διαθέτει η εφαρμογή, η δομή της είναι ερώτηση - απάντηση 1 – απάντηση 2 – απάντηση 3 – απάντηση 4 και σωστή απάντηση (1 έως 4).

Web Services:

Όλα τα web services είναι υλοποιημένα σε ένα αρχείο DBManager για βέλτιστη διαχείριση των services, καλώντας τα services από τα αντίστοιχα αρχεία τους.

- 1) **login:** Το service αυτό είναι υπεύθυνο για την σύνδεση του κάθε χρήστη στο σύστημα, δέχεται username και password και επιστρέφει ένα code. Το code αυτό δηλώνει την ιδιότητα του εκάστοτε χρήστη, δηλαδή 1 για τον διαχειριστή, 2 για τον υπεύθυνο, 3 για τον εξεταζόμενο και 0 σε περίπτωση λάνθασμένων στοιχείων.
- 2) **examq:** Το service αυτό είναι υπεύθυνο για την επιλογή 5 τυχαίων ερωτήσεων από την αποθήκη ερωτήσεων στη βάση για κάθε εξεταζόμενο που πραγματοποιεί μια εξέταση. Επιστρέφει όλα τα πεδία του πίνακα questions από τη βάση σε μορφή JSON.
- 3) **exam_con:** Το service αυτό είναι υπεύθυνο για τον έλεγχο διαθέσιμης εξέτασης για έναν χρήστη εφόσον αυτός την διακινούται (εξεταζόμενος). Δέχεται το username του χρήστη και επιστρέφει μηδέν όταν ο εξεταζόμενος έχει πραγματοποιήσει την εξέταση ή δεν έχει δικαίωμα συμμετοχής σε αυτή. Σε αντίθετη περίπτωση επιστρέφει το match_id από την αντιστοίχιση με την εξέταση που έχει ο χρήστης και θα δούμε παρακάτω.
- 4) **exam_enable:** Το service αυτό είναι υπεύθυνο για την ενεργοποίηση ή απενεργοποίηση μίας εξέτασης. Δέχεται το id μίας εξέτασης και αλλάζει στον πίνακα exams το πεδίο exam_status από 0 σε 1 και το αντίστροφο.
- 5) **add_question:** Το service αυτό είναι υπεύθυνο για την εισαγωγή ερωτήσεων στη βάση μέσω φόρμας από τον διαχειριστή. Δέχεται την ερώτηση, τις 4 πιθανές απαντήσεις και τέλος την σωστή απάντηση από τις παραπάνω. Επιστρέφει μήνυμα για την σωστή ή μη εισαγωγή της ερώτησης.
- 6) **add_user:** Το service αυτό είναι υπεύθυνο για την εισαγωγή χρηστών στη βάση μέσω φόρμας από τον διαχειριστή. Δέχεται τα στοιχεία του χρήστη, username ,password, το ονοματεπώνυμο του και επιστρέφει μήνυμα για την σωστή ή μη εισαγωγή του χρήστη.
- 7) **add_exam_center:** Το service αυτό είναι υπεύθυνο για την εισαγωγή εξεταστικού κέντρου στη βάση μέσω φόρμας από τον διαχειριστή. Δέχεται τα στοιχεία του εξεταστικού κέντρου , δηλαδή όνομα, διεύθυνση και επιστρέφει μήνυμα για την σωστή ή μη εισαγωγή του κέντρου.
- 8) **add_exam:** Το service αυτό είναι υπεύθυνο για την εισαγωγή εξέτασης στη βάση μέσω φόρμας από τον διαχειριστή. Δέχεται τα στοιχεία της εξέτασης, δηλαδή ημερομηνία και ώρα διεξαγωγής και εξεταστικό κέντρο (που είδαμε παραπάνω) και επιστρέφει μήνυμα για την σωστή ή μη εισαγωγή της εξέτασης.
- 9) **add_result:** Το service αυτό είναι υπεύθυνο για την καταχώρηση του αποτελέσματος της εξέτασης ενός χρήστη στη βάση . Δέχεται το match_id της

εξέτασης που συμμετείχε ο χρήστης και το συνολικό σκορ που πέτυχε στην εξέταση και επιστρέφει κατάλληλο μήνυμα.

- 10) **answer_question:** Το service αυτό είναι υπεύθυνο για την καταχώρηση του αποτελέσματος κάθε απάντησης που δίνει ο εξεταζόμενος σε μία ερώτηση. Δέχεται το match_id της εξέτασης, το id της ερώτησης, την σωστή απάντηση καθώς και την απάντηση που έδωσε ο εξεταζόμενος. Σε κάθε καταχώρηση γίνεται ενημέρωση της ημερομηνίας και ώρας που δώθηκε η απάντηση από time-server και επιστρέφεται κατάλληλο μήνυμα.
- 11) **centers:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των εξεταστικών κέντρων. Αυτή η ενέργεια γίνεται μόνο από τον διαχειριστή και επιστρέφει τα στοιχεία των κέντρων σε μορφή JSON.
- 12) **centers_ava:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των διαθέσιμων εξεταστικών κέντρων που ακόμη δεν έχουν υπεύθυνο. Αυτή η ενέργεια γίνεται μόνο από τον διαχειριστή και επιστρέφει τα στοιχεία των διαθέσιμων κέντρων σε μορφή JSON.
- 13) **all_exams:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των εξετάσεων. Αυτή η ενέργεια γίνεται μόνο από τον διαχειριστή και επιστρέφει τα στοιχεία των εξετάσεων (id,ημερομηνία,ώρα, εξεταστικό κέντρο) σε μορφή JSON.
- 14) **all_users:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των εξεταζόμενων. Αυτή η ενέργεια γίνεται μόνο από τον διαχειριστή και επιστρέφει τα στοιχεία των εξεταζόμενων (id,username,password,ονοματεπώνυμο) σε μορφή JSON.
- 15) **exam_info:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των εξετάσεων και του κέντρου που αντιστοιχούν από τον υπεύθυνο τους. Δέχεται το username ενός υπεύθυνου και επιστρέφει τα στοιχεία των εξετάσεων (id,ημερομηνία και ώρα,όνομα κέντρου,διεύθυνση και κατάσταση) σε μορφή JSON.
- 16) **users_ava_m:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των εξεταζόμενων που δεν έχουν αντιστοιχηθεί ακόμη σε κάποια εξέταση. Επιστρέφει τα στοιχεία των υποψήφιων εξεταζόμενων (id,username,password,ονοματεπώνυμο) σε μορφή JSON.
- 17) **users_ava_r:** Το service αυτό είναι υπεύθυνο για την εμφάνιση όλων των χρηστών τύπου 3(εξεταζόμενων) που δύναται(εφόσον δεν έχουν αντιστοιχηθεί και προφανώς δεν έχουν συμμετάσχει σε κάποια εξέταση) να αντιστοιχηθούν ως υπεύθυνοι σε κάποιο εξεταστικό κέντρο. Η ενέργεια αυτή γίνεται μόνο από την διαχειριστή και επιστρέφει τα στοιχεία των ήδη υπεύθυνων εξεταστικών κέντρων και των υποψήφιων υπεύθυνων (id,username,password,ονοματεπώνυμο) σε μορφή JSON.
- 18) **users_match:** Το service αυτό είναι υπεύθυνο για την αντιστοίχιση των διαθέσιμων εξεταζόμενων σε κάποια εξέταση Η ενέργεια αυτή γίνεται από τον

υπεύθυνο εξεταστικού κέντρου, δέχεται το id της εξέτασης και το id του χρήστη και επιστρέφει κατάλληλο μήνυμα.

- 19) **users_re:** Το service αυτό είναι υπεύθυνο για την αντιστοίχιση των διαθέσιμων υπεύθυνων ή υποψήφιων υπεύθυνων σε κάποιο εξεταστικό κέντρο. Η ενέργεια αυτή γίνεται από τον διαχειριστή, δέχεται το id του εξεταστικού κέντρου, το id του χρήστη και επιστρέφει κατάλληλο μήνυμα.

Αναφορές συστήματος:

(Όλες οι παρακάτω ενέργειες γίνονται μόνο από τον διαχειριστή !!)

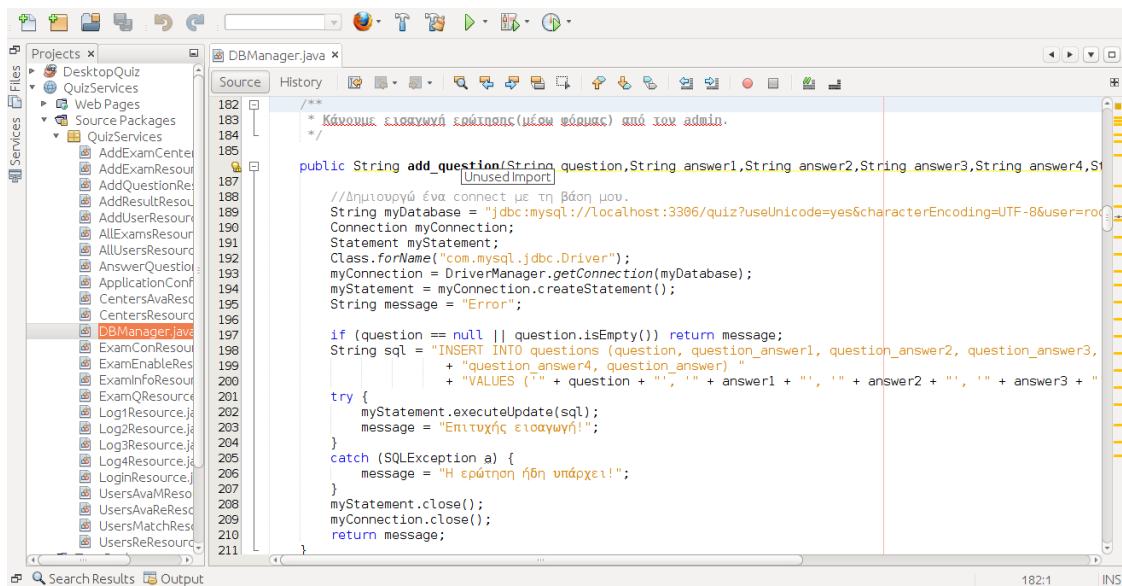
20) **log1:** Το service αυτό είναι υπεύθυνο για την εμφάνιση των αποτελεσμάτων με μορφή συνολικού θετικού αθροίσματος με βάση των κωδικό εξέτασης. Επιστρέφει τα στοιχεία των εξεταζόμενων (username,password,ονοματεπώνυμο) καθώς και στοιχεία του εξεταστικού κέντρου(ημερομηνία και διεύθυνση), μαζί με το συνολικό του άθροισμα στην εξέταση σε μορφή JSON.

21) **log2:** Το service αυτό είναι υπεύθυνο για την εμφάνιση των αποτελεσμάτων ανά εξεταστικό κέντρο με βάση την ημερομηνία εξέτασης. Δέχεται το id του εξεταστικού κέντρου και επιστρέφει τα ίδια στοιχεία με την παραπάνω αναφορά σε μορφή JSON.

22) **log3:** Το service αυτό είναι υπεύθυνο για την εμφάνιση των αποτελεσμάτων ανά εξέταση. Δέχεται το id της εξέτασης και επιστρέφει τα ίδια στοιχεία με την πρώτη αναφορά σε μορφή JSON.

23) **log4:** Το service αυτό είναι υπεύθυνο για την εμφάνιση των αποτελεσμάτων για δοθέντα κωδικό εξεταζόμενου. Δέχεται το id του εξεταζόμενου και επιστρέφει τον κωδικό της ερώτησης,την σωστή απάντηση,την απάντηση του εξεταζόμενου και την ώρα απάντησης σε μορφή JSON.

Screenshots από τον κώδικα των web services:



```
182  /**
183   * Κάνουμε εισαγωγή ερώτησης(μέσω φόρμας) από τον admin.
184   */
185
186  public String add_question(String question,String_answer1,String_answer2,String_answer3,String_answer4,String_answer5)
187  {
188      // Δημιουργώ ένα connect με τη βάση μου.
189      String myDatabase = "jdbc:mysql://localhost:3306/quiz?useUnicode=yes&characterEncoding=UTF-8&user=root&password=root";
190      Connection myConnection;
191      Statement myStatement;
192      Class.forName("com.mysql.jdbc.Driver");
193      myConnection = DriverManager.getConnection(myDatabase);
194      myStatement = myConnection.createStatement();
195      String message = "Error";
196
197      if (question == null || question.isEmpty()) return message;
198      String sql = "INSERT INTO questions (question, question_answer1, question_answer2, question_answer3, "
199          + "question_answer4, question_answer5) "
200          + "VALUES ('" + question + "','" + answer1 + "','" + answer2 + "','" + answer3 + "','" + answer4 + "','" + answer5 + "')";
201      try {
202          myStatement.executeUpdate(sql);
203          message = "Επιτυχής εισαγωγή!";
204      }
205      catch (SQLException a) {
206          message = "Η ερώτηση ήδη υπάρχει!";
207      }
208      myStatement.close();
209      myConnection.close();
210      return message;
211  }
```


The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a project named 'DesktopQuiz' with a 'QuizServices' package containing various resource files. The code editor displays the 'DBManager.java' file, specifically the 'AllExams()' method. The method is annotated with a Javadoc comment in Greek. It uses a MySQL JDBC connection to a database at 'localhost:3306/quiz'. It executes a 'SELECT * FROM exams' query and returns the results as a JSON array of objects, each containing exam details like id, datetime, center_name, and status.

```
422 /**  
423  * Επιστρέφει όλες τις εξετάσεις.  
424  */  
425  
426 public String AllExams() throws ClassNotFoundException, SQLException {  
427  
428     String myDatabase = "jdbc:mysql://localhost:3306/quiz?useUnicode=yes&characterEncoding=UTF-8&user=root&password=";  
429     Connection myConnection;  
430     Statement myStatement;  
431     Class.forName("com.mysql.jdbc.Driver");  
432     myConnection = DriverManager.getConnection(myDatabase);  
433     myStatement = myConnection.createStatement();  
434  
435     JsonBuilderFactory factory = Json.createBuilderFactory(null);  
436     JsonObjectBuilder jsonB = factory.createObjectBuilder();  
437     JsonArrayBuilder jsonA = factory.createArrayBuilder();  
438     String sql = "SELECT * from exams ";  
439     ResultSet rs = myStatement.executeQuery(sql);  
440     while (rs.next()) {  
441         jsonA = jsonA.add(factory.createObjectBuilder()  
442             .add("id", rs.getString("exam_id"))  
443             .add("datetime", rs.getString("exam_datetime"))  
444             .add("center_name", rs.getString("exam_center_id"))  
445             .add("status", rs.getString("exam_status")));  
446     }  
447     jsonB = jsonB.add("all_exams", jsonA);  
448     myStatement.close();  
449     myConnection.close();  
450     return jsonB.build().toString();  
451 }
```

The screenshot shows the same IDE environment, but now displaying the 'log1()' method in 'DBManager.java'. This method is also annotated with a Javadoc comment in Greek. It uses the same MySQL JDBC connection. It executes a complex SQL query that joins 'exams', 'users', 'exam_matches', and 'results' tables. The query filters for a specific exam center and returns detailed information including exam id, datetime, center name, center address, user name, and the result. The results are returned as a JSON object with a 'results' key.

```
636  
637 public String log1() throws ClassNotFoundException, SQLException {  
638  
639     String myDatabase = "jdbc:mysql://localhost:3306/quiz?useUnicode=yes&characterEncoding=UTF-8&user=root&password=";  
640     Connection myConnection;  
641     Statement myStatement;  
642     Class.forName("com.mysql.jdbc.Driver");  
643     myConnection = DriverManager.getConnection(myDatabase);  
644     myStatement = myConnection.createStatement();  
645  
646     JsonBuilderFactory factory = Json.createBuilderFactory(null);  
647     JsonObjectBuilder jsonB = factory.createObjectBuilder();  
648     JsonArrayBuilder jsonA = factory.createArrayBuilder();  
649     String sql = "SELECT exams.exam_id, exam_datetime, exam_center_name, exam_center_addr, username, result  
650         + \"WHERE users.user_id = exam_matches.user_id AND results.match_id = exam_matches.match_id AND e  
651         + \"AND exam_centers.exam_center_id = exams.exam_center_id\";  
652     ResultSet rs = myStatement.executeQuery(sql);  
653     while (rs.next()) {  
654         jsonA = jsonA.add(factory.createObjectBuilder()  
655             .add("id", rs.getString("exam_id"))  
656             .add("datetime", rs.getString("exam_datetime"))  
657             .add("center_name", rs.getString("exam_center_name"))  
658             .add("center_addr", rs.getString("exam_center_addr"))  
659             .add("username", rs.getString("username"))  
660             .add("result", rs.getString("result")));  
661     }  
662     jsonB = jsonB.add("results", jsonA);  
663     myStatement.close();  
664     myConnection.close();  
665     return jsonB.build().toString();  
666 }
```

Web application:

Η web εφαρμογή είναι από τα βασικότερα κομμάτια της εργασίας καθώς εδώ μπορεί ο κάθε χρήστης(διαχειριστής, υπεύθυνος και εξεταζόμενος) να εκτελέσει τις ενέργειες που δικαιούται. Στο web κομμάτι κάνουμε χρήση servlet για την διαχείριση http requests – response, αλλά και χρήση session για να κρατάμε τα attributes ου μας είναι απαραίτητα προκτυμένου να προχωρά η διαδικασία εξέτασης, πιστοποίησης ή αντιστοίχισης κτλπ. Επίσης μέσω των sessions επιτυγχάνουμε την ακεραιότητα μίας εξέτασης, δηλαδή δεν υπάρχει κίνδυνος για τις μεταβλητές μας σε τυχόν άλλες εξετάσεις που μπορεί να πραγματοποιούνται. Η web εφαρμογή αποτελείται από 3 ουσιαστικά σελίδες, την **LoginPage**, την **HomePage** και την **Logout**.

LoginPage:

Είναι η αρχική σελίδα στην οποία κατευθύνονται όλοι οι χρήστες προκυμένου να καταχωρήσουν τα στοιχεία εισόδου τους (username, password). Εφόσον τα στοιχεία αυτά είναι αληθή, τότε με την εκμετάλευση των web services που είδαμε παραπάνω παίρνουμε το αποτέλεσμα της login και της exam_con, δηλαδή το ρόλο του χρήστη και αν δικαιούται να λάβει μέρος σε κάποια εξέταση. Στην συνέχεια κατευθύνει τον χρήστη στην επόμενη σελίδα.

HomePage:

Έπειτα από την επιτυχή είσοδο του χρήστη, ανακατευθύνεται στην κύρια σελίδα της εφαρμογής όπου και πραγματοποιούνται οι βασικές λειτουργίες και για τους τρεις διακριτούς ρόλους.

- Στην περίπτωση που συνδεθεί εξεταζόμενος:

1. Πραγματοποιείται εξέταση πιστοποίησης γνώσεων για τα σήματα του ΚΟΚ, με τυχαίες ερωτήσεις από την αποθήκη των ερωτήσεων στη βάση.
2. Εμφανίζεται κατάλληλο μήνυμα διότι ο χρήστης είτε δεν έχει αντιστοιχηθεί σε εξέταση είτε έχει συμμετάσχει ήδη σε αυτήν.

Εκμεταλευόμαστε τις δυνατότητες του session, ώστε να αποθηκεύουμε στα attributes βασικές πληροφορίες που μας είναι απαραίτητες για την κάθε ενέργεια. Στην συγκεκριμένη περίπτωση από την χρήση του web service **exam**, παίρνουμε 5 τυχαίες ερωτήσεις όπου θα πρέπει να αποθηκεύουμε τις απαντήσεις του εξεταζόμενου για κάθε ερώτηση, το id της ερώτησης και η σωστή απάντηση στην ερώτηση ώστε να βγαζουμε

το result και τέλος μέσω arraylist να κρατάμε τις αναπάντητες ερωτήσεις ώστε να επιστρέφονται μέχρι να τις απαντήσει ο εξεταζόμενος και να λήξει η συνεδρία.

- Στην περίπτωση που συνδεθεί υπεύθυνος:

1. Αντιστοιχεί εξεταζόμενους σε εξετάσεις.
2. Ενεργοποιεί/Απενεργοποιεί τις διαθέσιμες εξετάσεις στις οποίες είναι υπεύθυνος.

- Στην περίπτωση που συνδεθεί διαχειριστής:

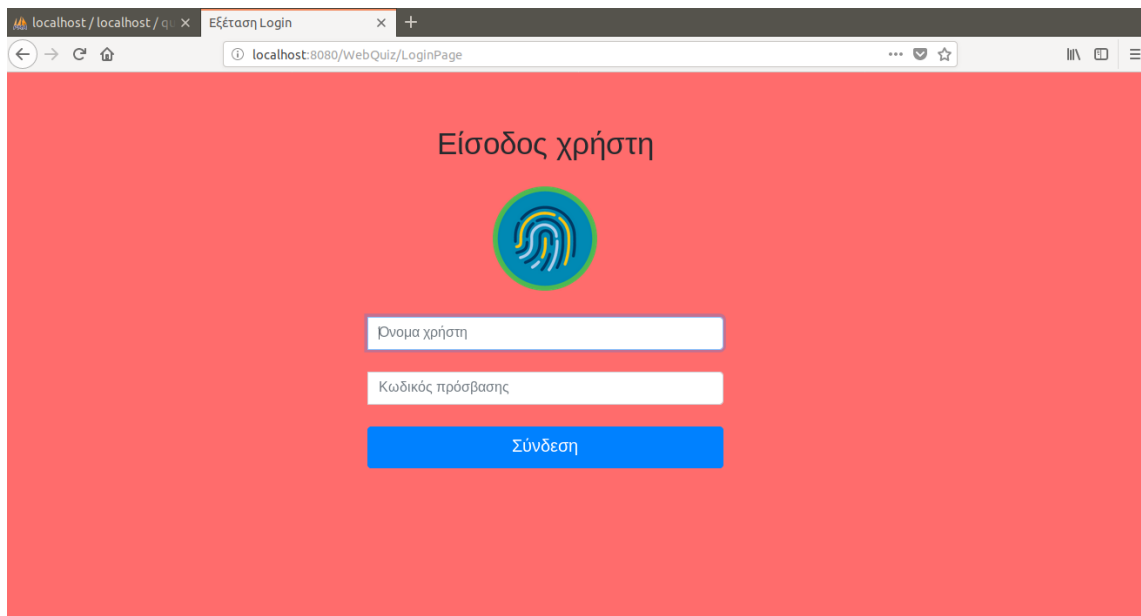
1. Εισάγει ερωτήσεις μέσω φόρμας.
2. Εισάγει χρήστες μέσω φόρμας.
3. Εισάγει εξεταστικά κέντρα.
4. Εισάγει εξετάσεις.
5. Πιστοποιεί υπεύθυνους εξεταστικών κέντρων.
6. Εμφανίζει τις 4 αναφορές του συστήματος.

Logout:

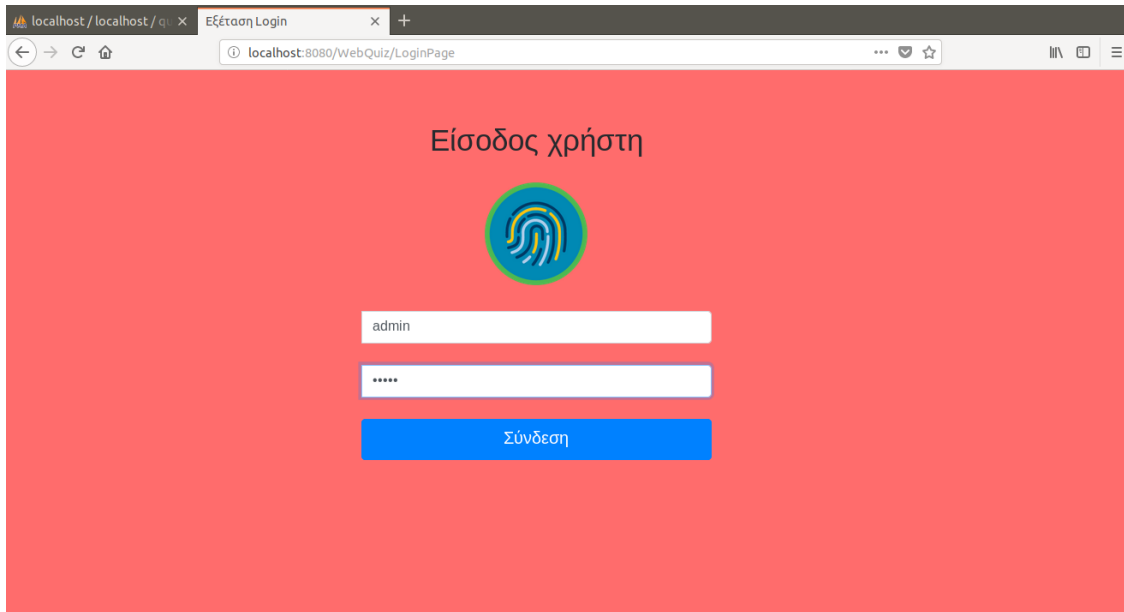
Στην συγκεκριμένη σελίδα χρησιμοποιούμε το servlet ώστε να αποσυνδεθεί ο εκάστοτε χρήστης από την εφαρμογή. Αυτό επιτυγχάνεται μέσω της διαγραφής του session και όλων των attributes που έχουμε αποθηκευμένα. Στην συνέχεια γίνεται ανακατεύθυνση στη LoginPage, στην περίπτωση του εξεταζόμενου δεν επιτρέπεται η έξοδος εφόσον έχει απαντηθεί έστω και μία ερώτηση, θα πρέπει να απαντηθούν και οι 5 ερωτήσεις.

Screenshots από την εκτέλεση του web application:

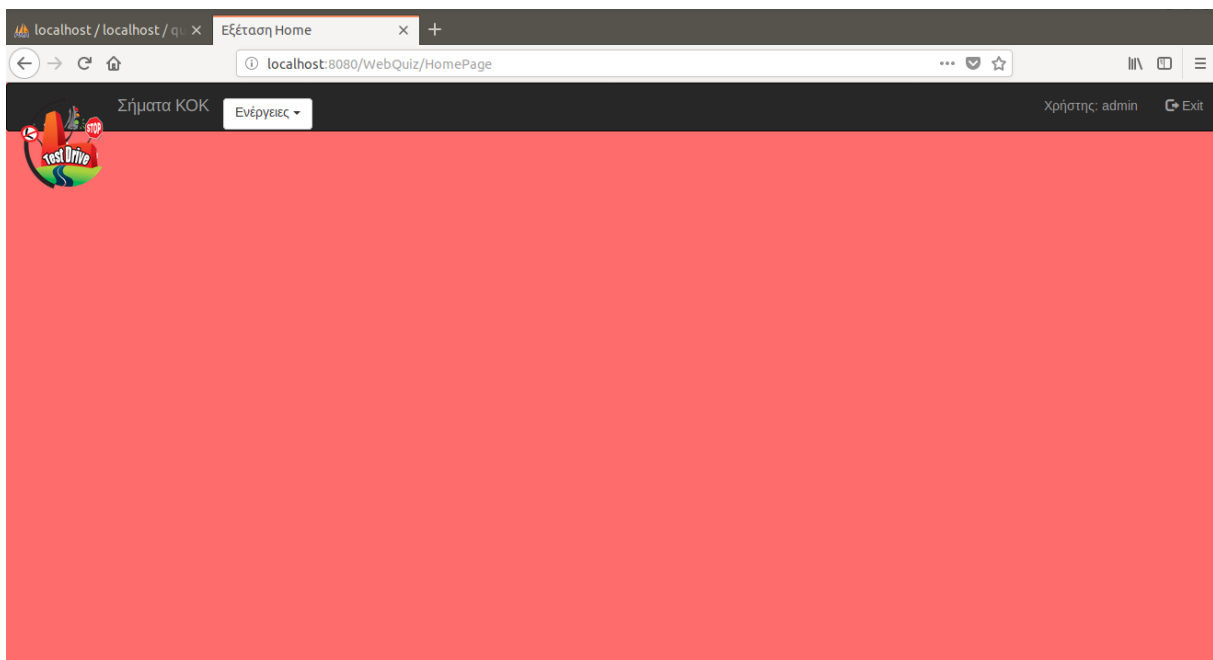
- Αρχική οθόνη



- Αρχική οθόνη με στοιχεία του διαχειριστή



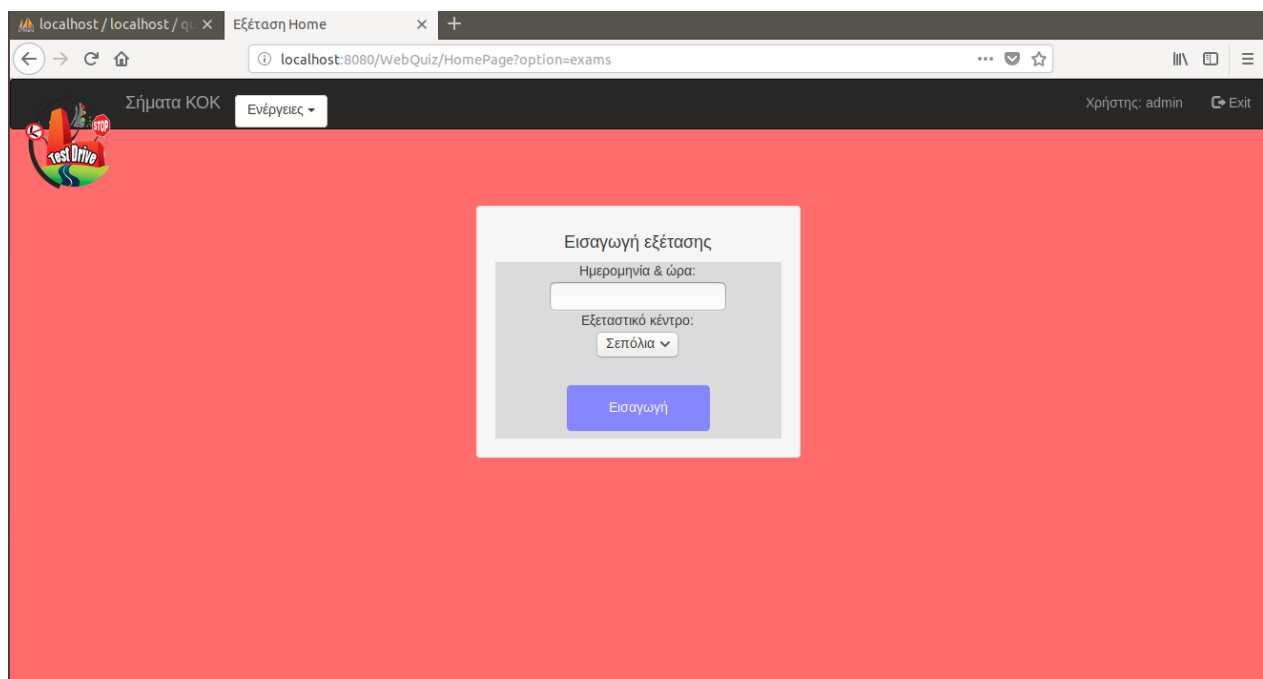
- Η κύρια οθόνη με τις διαθέσιμες ενέργειες για τον διαχειριστή.



- Φόρμα για εισαγωγή ερώτησης από τον διαχειριστή.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/WebQuiz/HomePage?option=questions_form'. The page has a dark header with the text 'Σήματα ΚΟΚ' and a dropdown menu 'Ενέργειες'. On the right side of the header, it says 'Χρήστης: admin' and 'Exit'. A sidebar menu on the left contains the following items: 'Εισαγωγή ερώτησης Μέσω φόρμας', 'Εισαγωγή χρήστη Μέσω φόρμας', 'Εισαγωγή κέντρου Κέντρο', 'Εισαγωγή εξέτασης Εξέταση', 'Πιστοποίηση υπεύθυνου Πιστοποίηση', and 'Αναφορές συστήματος Αναφορά 1η', 'Αναφορά 2η', 'Αναφορά 3η', 'Αναφορά 4η'. The main content area has a red background and features a central white box titled 'Εισαγωγή ερώτησης'. Inside this box, there is a form with the following fields: 'Ερώτηση:', 'Απάντηση 1:', 'Απάντηση 2:', 'Απάντηση 3:', 'Απάντηση 4:', 'Σωστή απάντηση:', and a blue button labeled 'Εισαγωγή'.

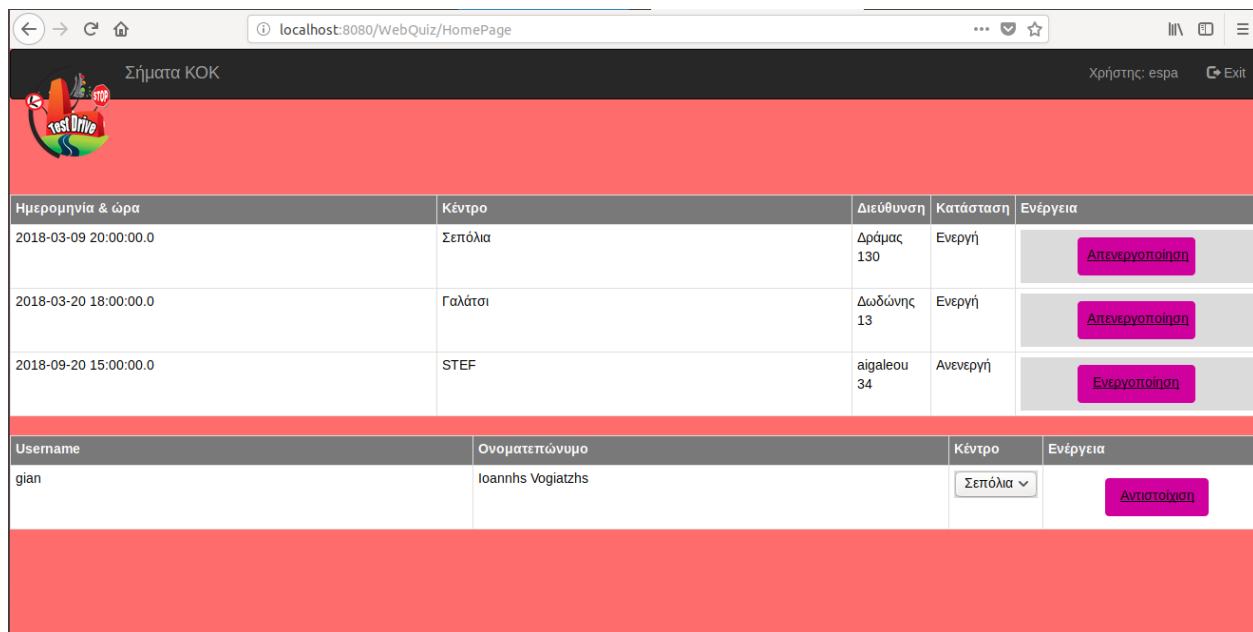
- Φόρμα εισαγωγής εξέτασης από τον διαχειριστή.



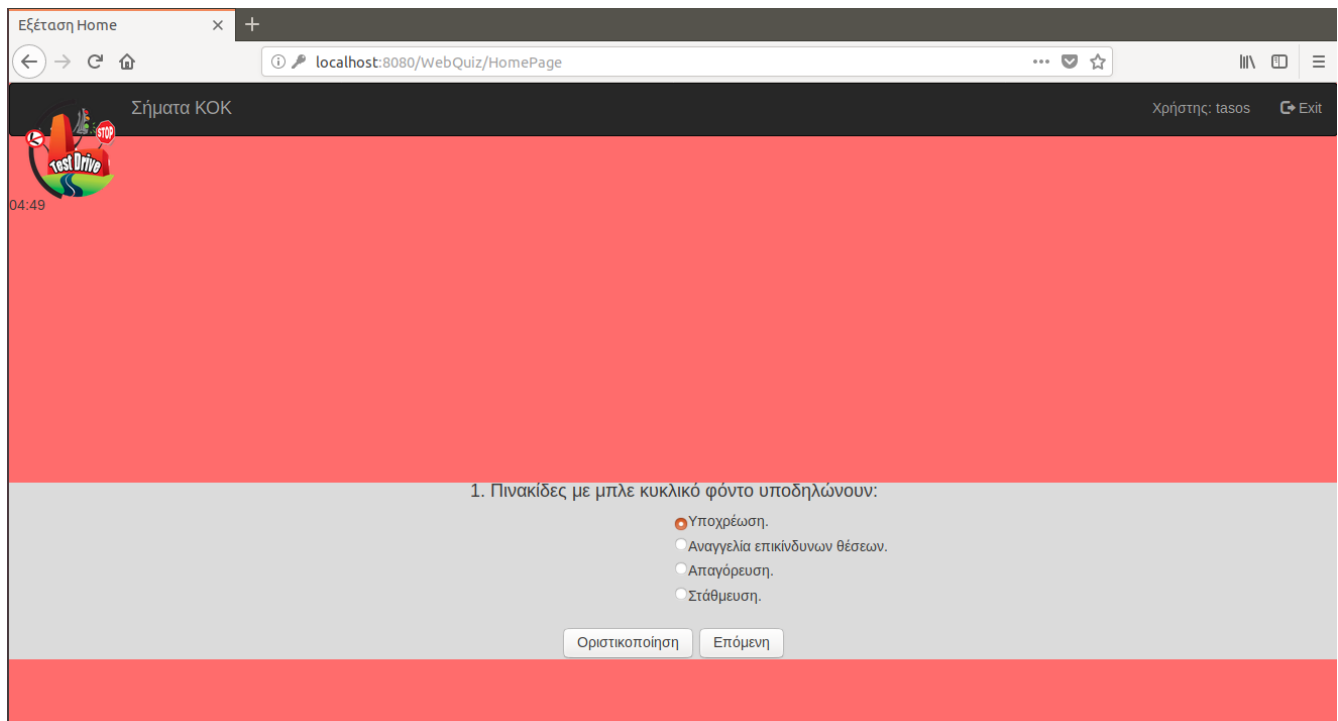
- Εμφάνιση αναφοράς του συστήματος με βάση την επιλογή της εξέτασης.

ID	Χρόνος	Εξεταστικό κέντρο	Διεύθυνση	Username	Αποτέλεσμα
1	2018-03-09 20:00:00.0	Σεπόλια	Δράμας 130	tasos	1

- Κύρια οθόνη για τον υπεύθυνο εξεταστικών κέντρων και των διαθέσιμων ενεργειών του, αντιστοίχιση εξεταζόμενων με εξετάσεις και ενεργοποίηση/απενεργοποίηση εξετάσεων.



- Στην περίπτωση ενός εξεταζόμενου έχουμε παρακάτω ένα παράδειγμα εκτέλεσης μιας εξέτασης.



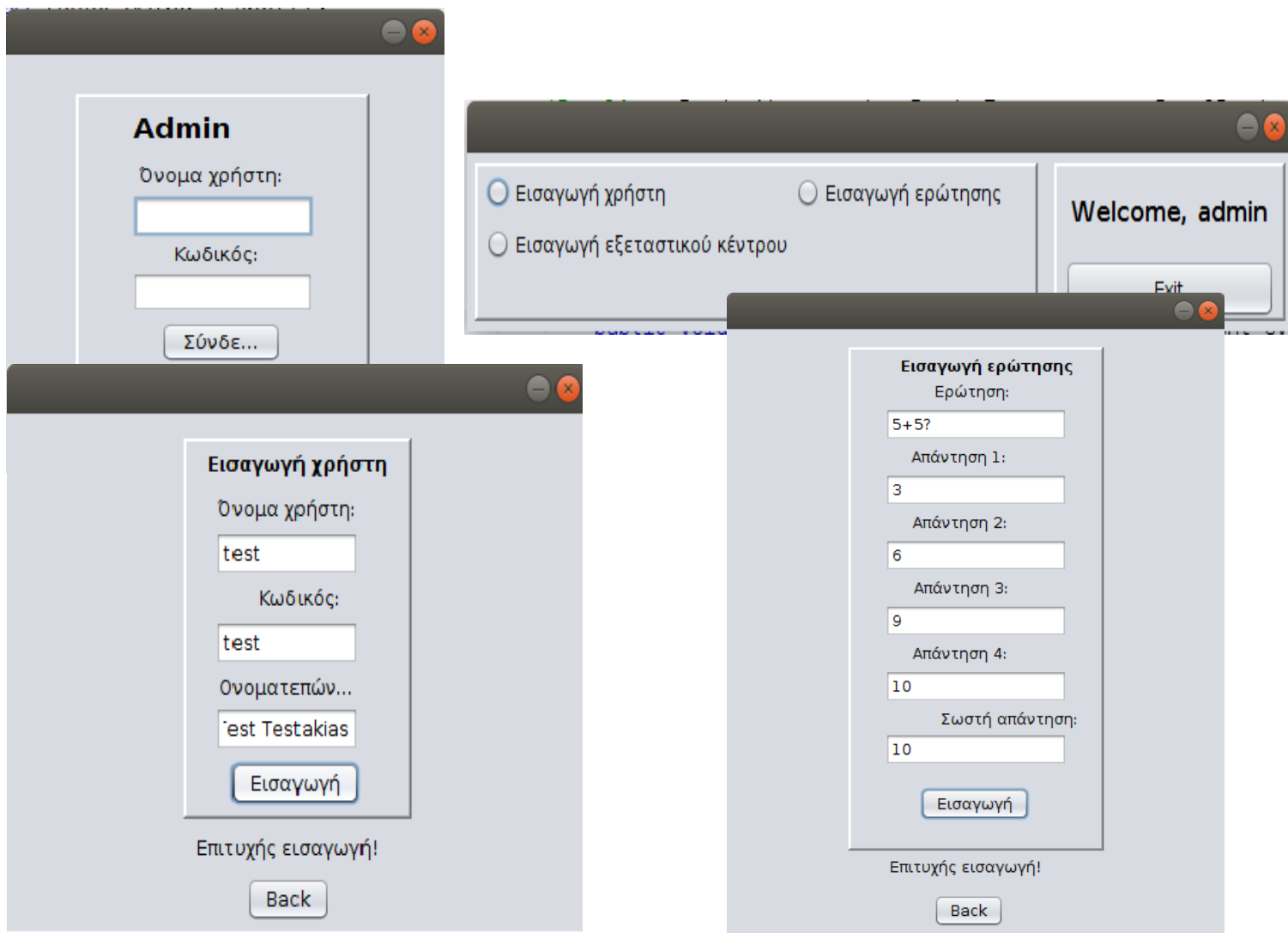
Desktop application:

Το Desktop κομμάτι της εργασίας δημιουργήθηκε εξολοκλήρου για τον διαχειριστή του συστήματος εξετάσεων ΚΟΚ, έτσι ώστε να μην είναι απαραίτητο να πηγαίνει στη web app και να επιτελεί τις διάφορες λειτουργίες που χρειάζεται. Η υλοποίηση της εφαρμογής έχει γίνει με GUI (Jframe, RadioButtons κτλπ) για καλύτερη γραφική απεικόνιση της εφαρμογής. Για όσες ενέργειες εκτελούνται και σε αυτήν την εφαρμογή εκμεταλεύόμαστε τα web services, καλώντας τα και εκκλυρονώντας τις απαραίτητες ενέργειες.

- Αρχεία .java που δημιουργήθηκαν:

- **Login.java:** Είναι η πρώτη σελίδα, στην οποία ο εκάστοτε διαχειριστής εισάγει τα στοιχεία του (username και password) και παίρνει πρόσβαση στην κύρια σελίδα.
- **Main.java:** Είναι η κύρια σελίδα της εφαρμογής μας στην οποία ο admin μπορεί να επιλέξει μια σειρά ενεργιών που θα δούμε παρακάτω.
- **AddUser.java:** Σε αυτή τη σελίδα γίνεται εισαγωγή χρηστών.
- **AddQuestion.java:** Σε αυτή τη σελίδα γίνεται εισαγωγή ερωτήσεων.
- **AddCenter.java:** Σε αυτή τη σελίδα γίνεται εισαγωγή εξεταστικών κέντρων.

Screenshots από την εκτέλεση της Desktop εφαρμογής:



ΤΕΛΟΣ ΕΡΓΑΣΙΑΣ