

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον
παγκόσμιο ιστό»

3	Τελική Εργασία
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Θοδωρής Χαραλαμπόπουλος Π18169
	Αντώνης Καλαμπόγιας Π18050
	Γιώργος Καλούδης Π18054
Ημερομηνία παράδοσης	7-7-2020



Εκφώνηση της άσκησης

Ολοκλήρωση 3-tier εφαρμογής για την έκδοση λογαριασμών κινητής τηλεφωνίας Στόχοι εργασίας: Ολοκλήρωση λειτουργικότητας 3-tier εφαρμογής, ολοκλήρωση server-side τεχνολογιών (servlets, jsp, βάση δεδομένων), 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 τις λειτουργίες (μεθόδους) που ορίσατε στις προηγούμενες ασκήσεις.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Γενική Περιγραφή της λύσης.....	4
1.1	Εμφάνιση συγκεκριμένων πληροφοριών στη σελίδα	11
2	Κώδικας προγράμματος	14
2.1	Κώδικας της κλάσης Admin.....	15
2.2	Κώδικας της κλάσης Encryption.....	16
2.3	Κώδικας της κλάσης Users	17
2.4	Κώδικας της κλάσης Customer	18
3	Βιβλιογραφικές Πηγές.....	21



1 Γενική Περιγραφή της λύσης

Το πρόγραμμα της εργασίας υλοποιεί μία 3-tier εφαρμογή έκδοσης και διαχείρισης λογαριασμών κινητής τηλεφωνίας. Η εφαρμογή εξυπηρετεί 3 είδη χρηστών(Client,Seller,Admin).

Μέσα από την αρχική σελίδα δίνεται η δυνατότητα στους χρήστες να κάνουν login ή register.

Κάποια credentials για μερικούς χρήστες:

Username: tlarmuth1	Password: ab123G	Client
Username: mstolz2	Password: 123	Client
Username: dwatkiss9	Password: ghT35	Seller
Username: tnunan8	Password: abc	Seller
Username: hpockey1	Password: 12pl	Admin
Username: agrief0	Password: abge4	Admin



Registration

Name:

Surname:

Username:

Password:

AFM:

Type:

Program:

Register

[Back](#)

Η συγκεκριμένη εγγραφή χρησιμοποιείται μόνο για την εγγραφή χρήστη τύπου client.

Client

Ο client έχει τις εξής λειτουργίες που μπορεί να εκτελέσει:



1. Προβολή λογαριασμού
2. Ιστορικό κλήσεων
3. Εξόφληση λογαριασμού

1) Προβολή λογαριασμού

Προβολή λογαριασμού	Προβολή λογαριασμού			
Ιστορικό κλήσεων				
Εξόφληση λογαριασμού				
	Όνομα:	Cullin		
	Επίθετο:	Tyrone		
	Όνομα Χρήστη:	tlarmuth1		
	ΑΦΜ:	949795955		
	Τηλέφωνο:	+31 711 279 3170		
	Πρόγραμμα:	800 minutes 800 sms		
	Ημερομηνία εγγραφής:	2020-03-07 15:11:46		

2) Ιστορικό κλήσεων

Προβολή λογαριασμού	Ιστορικό κλήσεων			
Ιστορικό κλήσεων				
Εξόφληση λογαριασμού				
	Διαρκία Κλήσης	Κλήση απο	Κλήση προς	Ημερομηνία κλήσης
	19	+86 496 991 2771	+86 285 439 4126	2019-08-06 08:03:09
	2	+370 989 384 1205	+55 978 815 9132	2019-08-15 16:18:46



3)Εξόφληση λογαριασμού

Στην εξόφληση λογαριασμού αν έχει πληρωθεί ο λογαριασμός εμφανίζει στο χρήστη κατάλληλο μήνυμα ότι ο λογαριασμός είναι πληρωμένος αλλιώς εμφανίζεται κατάλληλο κουμπί για την πληρωμή.

Προβολή λογαριασμού	Εξόφληση λογαριασμού
Ιστορικό κλήσεων	
Εξόφληση λογαριασμού	
	Username: tlarmuth1
	Αριθμος Λογαριασμου: 39
	Εχει πληρωθει;: Payed !
	Ποσο: 5
	For more information about your account visit the first tab "Προβολή λογαριασμού"

Seller

Ο seller έχει τις εξής λειτουργίες που μπορεί να εκτελέσει:

1. Καταχώρηση νέου πελάτη
2. Προβολή διαθέσιμων πακέτων
3. Αλλαγή προγράμματος πελάτη
4. Έκδοση λογαριασμού

1)Καταχώρηση νέου πελάτη



Καταχώρηση νέου πελάτη	Καταχώρηση νέου πελάτη και επιλογή προγράμματος τηλεφωνίας
Προβολή διαθέσιμων πακέτων	
Αλλαγή προγράμματος πελάτη	
Έκδοση λογαριασμού	

Name:

Surname:

Username:

Password:

AFM:

Type:

Program:

2) Προβολή διαθέσιμων πακέτων

Καταχώρηση νέου πελάτη	Προγράμματα
Προβολή διαθέσιμων πακέτων	
Αλλαγή προγράμματος πελάτη	
Έκδοση λογαριασμού	

Id	Όνομα Προγράμματος	Τιμή/λεπτό	Τιμή/MB	Τιμή/sms	Διαθέσιμα λεπτά	Διαθέσιμα MB	Διαθέσιμα sms
1	1500 mb 1500 minutes	1	1	0	1500	1500	0
2	500 mb 500 minutes 500 sms	2	1	1	500	500	500
3	500 mb	0	1	0	0	500	0
4	800 minutes 800 sms	2	0	1	800	0	800
5	2 GB	0	2	0	0	2000	0

3) Αλλαγή προγράμματος πελάτη



Καταχώρηση νέου πελάτη	<h3>Αλλαγή προγράμματος πελάτη</h3> <p>Όνομα: <input type="text"/></p> <p>Επίθετο: <input type="text"/></p> <p>Πρόγραμμα: <input type="text" value="Επιλέξτε πρόγραμμα"/></p> <p><input type="button" value="Αλλαγή"/></p>
Προβολή διαθέσιμων πακέτων	
Αλλαγή προγράμματος πελάτη	
Έκδοση λογαριασμού	

4) Έκδοση λογαριασμού

Καταχώρηση νέου πελάτη	<h3>Έκδοση λογαριασμού</h3> <p>Όνομα: <input type="text"/></p> <p>Επίθετο: <input type="text"/></p> <p><input type="button" value="Έκδοση λογαριασμού"/></p>
Προβολή διαθέσιμων πακέτων	
Αλλαγή προγράμματος πελάτη	
Έκδοση λογαριασμού	

Admin

Ο admin έχει τις εξής λειτουργίες που μπορεί να εκτελέσει:

1. Δημιουργία νέου πωλητή
2. Διαγραφή πωλητή
3. Δημιουργία νέου προγράμματος

1) Δημιουργία νέου πωλητή



Δημιουργία νέου πωλητή	<h3>Δημιουργία νέου πωλητή</h3> <hr/> <p>Name: <input type="text"/></p> <p>Surname: <input type="text"/></p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p>Type: <input type="text" value="Seller"/></p> <p><input type="button" value="Καταχώρηση"/></p>
Διαγραφή πωλητή	
Δημιουργία νέου προγράμματος	

2) Διαγραφή πωλητή

Δημιουργία νέου πωλητή	<h3>Διαγραφή πωλητή</h3> <hr/> <p>Name: <input type="text"/></p> <p>Surname: <input type="text"/></p> <p><input type="button" value="Διαγραφή"/></p>
Διαγραφή πωλητή	
Δημιουργία νέου προγράμματος	

3) Δημιουργία νέου προγράμματος



Δημιουργία νέου πωλητή

Διαγραφή πωλητή

Δημιουργία νέου προγράμματος

Δημιουργία νέου προγράμματος

Όνομα προγράμματος:

Διαθέσιμα λεπτά:

Διαθέσιμα MB:

Διαθέσιμα sms:

Τιμή/λεπτό:

Τιμή/mb:

Τιμή/sms:

Καταχώρηση

1.1 Εμφάνιση συγκεκριμένων πληροφοριών στη σελίδα

Στη σελίδα του Seller για να εμφανιστούν τα διαθέσιμα προγράμματα όπως εδώ

Προγράμματα

Id	Όνομα Προγράμματος	Τιμή/λεπτό	Τιμή/MB	Τιμή/sms	Διαθέσιμα λεπτά	Διαθέσιμα MB	Διαθέσιμα sms
1	1500 mb 1500 minutes	1	1	0	1500	1500	0
2	500 mb 500 minutes 500 sms	2	1	1	500	500	500
3	500 mb	0	1	0	0	500	0
4	800 minutes 800 sms	2	0	1	800	0	800
5	2 GB	0	2	0	0	2000	0

Γίνεται μέσω της σελίδας jsp.

```
<%ConnectToDB con1 = new ConnectToDB();
con1.loadPrograms();
StringBuilder templ = con1.displayPrograms();
out.println(templ.toString());
%>
```



Η οποία καλεί την `loadPrograms` από την κλάση `ConnectToDb` η οποία αποθηκεύει της πληροφορίες σε μια λίστα και τα επιστρέφει με `StringBuilder` με τη μορφή `html`.

```
public StringBuilder loadProgramsName() //loads all programs from the database and puts
{
    programsName = new ArrayList<>();
    StringBuilder temp;
    temp = new StringBuilder();
    try
    {
        Class.forName (DATABASE_DRIVER);
        Connection con = DriverManager.getConnection(DATABASE_URL, getProperties());
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery( sql: "select * from plans;");
        while(rs.next())
        {
            String progName;
            progName = rs.getString( columnName: "ProgramName");
            programsName.add("<option>" + progName + "</option>");
        }
        if (programsName.size() > 0)
        {
            for (String program : programsName)
            {
                temp.append(program);
            }
        }
        else
        {
            System.out.println("");
        }
        con.close();
    }
}
```

Επίσης με παρόμοιο τρόπο βάζει τα ονόματα των προγραμμάτων μέσα από τη βάση στο `selection` της `html`.

Πρόγραμμα: Επιλέξτε πρόγραμμα

- 1500 mb 1500 minutes
- 500 mb 500 minutes 500 sms
- 500 mb
- 800 minutes 800 sms
- 2 GB

```
<select name="programe">
    <option value="" selected disabled hidden>Επιλέξτε πρόγραμμα</option>
    <%
        ConnectToDB con2 = new ConnectToDB();
        StringBuilder temp2 = con2.loadProgramsName();
        out.println(temp2);
    %>
</select>
```



Όπως επίσης και στη σελίδα του customer για να εμφανίζει τις πληροφορίες του

Προβολή λογαριασμού

Όνομα: Cullin
Επίθετο: Tyrone
Όνομα Χρήστη: tlarmuth1
ΑΦΜ: 949795955
Τηλέφωνο: +31 711 279 3170
Πρόγραμμα: 800 minutes 800 sms
Ημερομηνία εγγραφής: 2020-03-07 15:11:46



2 Κώδικας προγράμματος

Ξεκινώντας με το login, ο χρήστης εισάγει ένα username και ένα password και πατώντας το κουμπί του login γίνεται action στο LoginServlet μέσα από το οποίο καλείται η μέθοδος login της κλάσης customer και μέσω του salt και του salted_hash ελέγχονται τα στοιχεία.

```
String username = request.getParameter( S: "user");
String password = request.getParameter( S: "pwd");

ArrayList<String> arrayList = usr1.Login(username,password);    //login method returns a string with the type of the
if (arrayList.isEmpty())
{
    request.getRequestDispatcher( S: "/index.jsp").forward(request, response);
    arrayList.clear();
}
HttpSession session=request.getSession();
if(arrayList.get(0).equals("Seller"))
{
    session.setAttribute( S: "username",username);
    session.setAttribute( S: "type",arrayList.get(0));
    request.getRequestDispatcher( S: "/SellerPage.jsp").forward(request, response);
    arrayList.clear();
}
else if(arrayList.get(0).equals("Client"))
{
    session.setAttribute( S: "username",username);
    session.setAttribute( S: "type",arrayList.get(0));
    request.getRequestDispatcher( S: "/CustomerPage.jsp").forward(request, response);
    arrayList.clear();
}
else if(arrayList.get(0).equals("Admin"))
{
    session.setAttribute( S: "username",username);
    session.setAttribute( S: "type",arrayList.get(0));
    request.getRequestDispatcher( S: "/AdminPage.jsp").forward(request, response);
    arrayList.clear();
}
```



```
public ArrayList<String> Login(String username, String password){
    String hash="";
    String salt="";
    try
    {
        Class.forName (DATABASE_DRIVER);    //Specifies the driver

        Connection con = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection

        PreparedStatement statement = con.prepareStatement( "sql: \"select * from users where Username = ?\"");

        statement.setString( parameterIndex: 1, username);    //Passes first parameter

        ResultSet rs = statement.executeQuery(); //Executes the query

        if(rs.next())
        {
            salt = rs.getString( columnIndex: 3);
            hash = rs.getString( columnIndex: 4);
        }
        else
        {
            return userProperties;
        }
        if(Encryption.getHashMD5(password,salt).equals(hash)){
            userProperties.add(ConnectToDB.getUserType(username));
            userProperties.add(Integer.toString(ConnectToDB.getUserID(username)));
            UserIDofSellerOrAdminOrCustomer = userProperties.get(1);
        }
        else{
            return userProperties;
        }
    }
}
```

Η μέθοδος login επιστρέφει ένα array list με το τύπο του χρήστη και το id του.

Ανάλογα με το τύπο του χρήστη γίνεται ανακατεύθυνση στην απαιτούμενη σελίδα.

Μέσα από τις σελίδες των (admin,seller,client) ανάλογα με την ενέργεια που θέλουν να εκτελέσουν καλείται και το αντίστοιχο servlet(AdminServlet,SellerServlet,CustomerServlet) μέσα από τα οποία εκτελούνται οι αντίστοιχοι μέθοδοι.

2.1 Κώδικας της κλάσης Admin

Η μέθοδος createseller της admin παίρνει τις παραμέτρους από τη συμπλήρωση φόρμας και με τον δοσμένο κωδικό δημιουργείται ένα random salt και ένα salted_hash μέσω της κλάσης Encryption και τα εκχωρεί στη βάση δεδομένων.



```
public static boolean CreateSeller(String userName, String password, String FName, String LName,String date,String type)
{
    String salt,salted_hash;

    SecureRandom random = new SecureRandom();
    byte[] randomSalt = new byte[20];
    random.nextBytes(randomSalt);

    salt = randomSalt.toString();
    salted_hash = Encryption.getHashMD5(password,salt);

    int usrIDofCustomer = 0;

    //Create user
    try
    {
        Class.forName (DATABASE_DRIVER); //Specifies the driver

        Connection con = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection

        //Standard SQL statement (always WITH parameters)
        PreparedStatement statement = con.prepareStatement( "sql: insert into users (UserName, salt,salted_hash, Type) values (?, ?, ?, ?);");

        statement.setString( parameterIndex 1, userName); //Passes first parameter

        statement.setString( parameterIndex 2, salt); //Passes second parameter

        statement.setString( parameterIndex 3, salted_hash); //Passes third parameter
    }
}
```

Η μέθοδος createPlan παίρνει ως παραμέτρους τα στοιχεία ενός νέου προγράμματος και το εκχωρεί στη βάση.

```
public static void createPlan(String ProgramName, String ChargePerMin, String ChargePerMB, String ChargePerSMS, String MinutesToTalk, String MBtoSpend, String smsToSp)
{
    try
    {
        Class.forName (DATABASE_DRIVER);

        Connection con2 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
        //Standard SQL statement (always WITH parameters)
        PreparedStatement statement1 = con2.prepareStatement( "sql: insert into plans (ProgramName, ChargePerMin, ChargePerMB, ChargePerSMS, MinutesToTalk, MBtoSpend, s
    }
}
```

2.2 Κώδικας της κλάσης Encryption

Η συγκεκριμένη κλάση παίρνει το password και ένα random salt και ενώνοντας τα δημιουργεί ένα hash 32bit τύπου MD5.

```
package BasicClasses;

import ...

public class Encryption {
    public static String getHashMD5(String password,String salt) {
        final String toHash = salt + password;
        MessageDigest messageDigest = null;
        try {
            messageDigest = MessageDigest.getInstance("MD5");
        } catch (NoSuchAlgorithmException ex) {
            return "00000000000000000000000000000000";
        }
        messageDigest.update(toHash.getBytes(), offset: 0, toHash.length());
        String hashed = new BigInteger( signum: 1, messageDigest.digest()).toString( radix: 16);
        if (hashed.length() < 32) {
            hashed = "0" + hashed;
        }
        return hashed.toUpperCase();
    }
}
```




2.3 Κώδικας της κλάσης Users

Η κλάση Users περιέχει τις μεθόδους που είναι γενικού τύπου, δηλαδή login και register

Η μέθοδος login έχει αναφερθεί στην γενική περίληψη του κώδικα.

Η μέθοδος register παίρνει τα στοιχεία από τη φόρμα εγγραφής μέσω του registration servlet και καταχωρεί στη βάση τον νέο χρήστη.

```
public void Register(String userName, String password, String type, String FName, String LName, int AFM, String programName,String date)
{
    //First creates user (so the Customer/Seller can have the UserID (its required) )
    //Then creates the Customer/Seller and provides the newly created UserID
    String salt,salted_hash;

    SecureRandom random = new SecureRandom();
    byte[] randomSalt = new byte[20];
    random.nextBytes(randomSalt);

    salt = randomSalt.toString();
    salted_hash = Encryption.getHashMD5(password,salt);

    int usrIDofCustomer;

    //Create user
    try
    {
        Class.forName (DATABASE_DRIVER);    //Specifies the driver

        Connection con = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection

        //Standard SQL statement (always WITH parameters)
        PreparedStatement statement = con.prepareStatement( sql: "insert into users (UserName, salt,salted_hash, Type) values (?, ?, ?, ?);");

        statement.setString( parameterIndex: 1, userName);    //Passes first parameter

    }
    try
    {
        usrIDofCustomer = ConnectToDB.getUserID(userName); //gets the userID

        Connection con2 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
        //Standard SQL statement (always WITH parameters)
        PreparedStatement statement1 = con2.prepareStatement( sql: "insert into customers (FirstName, LastName, AFM,JoinedWhen, UserID) values (?, ?, ?, ?, ?);");

        statement1.setString( parameterIndex: 1, FName);    //Passes first parameter

        statement1.setString( parameterIndex: 2, LName);    //Passes second parameter

        statement1.setInt( parameterIndex: 3, AFM);    //Passes third parameter

        statement1.setString( parameterIndex: 4, date);

        statement1.setInt( parameterIndex: 5, usrIDofCustomer);

        statement1.executeUpdate();

        con2.close();
    }
    catch (Exception e)
    {
        System.out.println(e);
    }
}
```



2.4 Κώδικας της κλάσης Customer

Η κλάση customer περιέχει τις μεθόδους ClientShowDetails, CustomerCalls, SubscriptionViewer, SubscriptionViewerWhenTheBillsNOTPaid και PaySubscription.

```
public static ArrayList<String> ClientShowDetails(String providedUsername){
    //Called from CustomerPage.jsp when the client request to view his information
    ArrayList<String> UserInfo = new ArrayList<>(); //Array list where the info of user are saved
    try
    {
        Class.forName (DATABASE_DRIVER); //Specifies the driver

        Connection con3 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
        Statement stmt = con3.createStatement();
        //Create Connection to the DB

        ResultSet rs = stmt.executeQuery( sql: "SELECT * FROM users,customers,phonenumber,soldplans where users.Username="+providedUsername+" LIMIT 1;");
        //Execute query to import the users info

        if(rs.next()) //Now that rs holds the info the user requesting his info we add the values where they belong
        {
            String CustomerID;
            int UserID,PhoneNumID,PlanID;
            UserInfo.add(rs.getString( columnName: "Username"));
            UserInfo.add(rs.getString( columnName: "FirstName"));
            UserInfo.add(rs.getString( columnName: "LastName"));
            UserInfo.add(rs.getString( columnName: "AFM"));
            UserInfo.add(rs.getString( columnName: "JoinedWhen"));
            UserInfo.add(rs.getString( columnName: "PhoneNumber"));
            UserID = rs.getInt( columnName: "UserID");
            PhoneNumID = rs.getInt( columnName: "PhoneNumID");
            PlanID = rs.getInt( columnName: "PlanID");
            //NOTE from the previous query we don't get the Name of the program the Client is subscribed so
            stmt = con3.createStatement();
            rs = stmt.executeQuery( sql: "SELECT * FROM 'plans' WHERE PlanID = "+PlanID+" LIMIT 1;");

            //Now that we have all the needed values lets return them to the Client
            if(rs.next()){
                UserInfo.add(rs.getString( columnName: "ProgramName"));
                if (UserInfo.isEmpty())
                {
                    return null;
                }
                else {
                    return UserInfo;
                }
            }
        }
        else System.out.println("Oh no...\nSomething went wrong!!");
        con3.close();//Close the connection to the DB
    }

    catch (Exception e)
    {
        System.out.println(e);
    }
    return null;
}
```



```
public static String CustomerCalls(String ProvidedUsername){
    //Called from CustomerPage.jsp when the client request to view his call history
    String UserCallInfo="";
    try
    {
        Class.forName (DATABASE_DRIVER); //Specifies the driver

        Connection con4 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
        Statement stmt = con4.createStatement();
        //Create Connection to the DB

        ResultSet rs = stmt.executeQuery(
            "SELECT\n" +
            "    Duration,\n" +
            "    CallerNumber,\n" +
            "    CalleeNumber,\n" +
            "    CallMadeWhen\n" +
            "FROM\n" +
            "    users\n" +
            "JOIN customers ON customers.UserID = users.UserID\n" +
            "JOIN callsmade ON callsmade.CustomerID = customers.CustomerID\n" +
            "\n" +
            "WHERE\n" +
            "    users.Username = '"+ProvidedUsername+"';");
        //Execute query to import the users info

        while(rs.next())
        {
            String Duration,CallerNumber,CalleeNumber,CallMadeWhen;
            Duration = rs.getString( columnLabel: "Duration");
            CallerNumber = rs.getString( columnLabel: "CallerNumber");
            CalleeNumber = rs.getString( columnLabel: "CalleeNumber");
            CallMadeWhen = rs.getString( columnLabel: "CallMadeWhen");
            UserCallInfo += "<tr><td>" + Duration + "</td><td>" + CallerNumber + "</td><td>" + CalleeNumber + "</td><td>" + CallMadeWhen + "</td></tr>";
        }
        con4.close();//Close the connection to the DB
        return UserCallInfo.toString();
    }

    catch (Exception e)
    {
        System.out.println(e);
    }
    return null;
}
```



```
public static ArrayList<String> SubscriptionViewer(String ProvidedName){
    ArrayList<String> UserReturnSubInfo = new ArrayList<>();
    try
    {
        Class.forName (DATABASE_DRIVER); //Specifies the driver

        Connection con4 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
        Statement stmt = con4.createStatement();
        //Create Connection to the DB

        ResultSet tr = stmt.executeQuery( sql: "SELECT\n" +
            "    bills.isPaid,\n" +
            "    bills.BillID,\n" +
            "    bills.MoneyToPay,\n" +
            "    users.Username,\n" +
            "    customers.FirstName,\n" +
            "    customers.LastName,\n" +
            "    customers.AFM\n" +
            "FROM\n" +
            "    users\n" +
            "JOIN customers ON customers.UserID = users.UserID\n" +
            "JOIN bills ON bills.CustomerID = customers.CustomerID\n" +
            "JOIN soldplans ON soldplans.PlanID = bills.PlanID\n" +
            "WHERE\n" +
            "    users.Username = '" + ProvidedName + "'\n" +
            "LIMIT 1 ;");

        //Execute query to import the users info
    }

    public static ArrayList<String> SubscriptionViewerWhenTheBillIsNOTPaid(String ProvidedName){
        ArrayList<String> UserReturnSubInfo = new ArrayList<>();
        try
        {
            Class.forName (DATABASE_DRIVER); //Specifies the driver

            Connection con4 = DriverManager.getConnection(DATABASE_URL,ConnectToDB.getProperties()); //Creates a connection
            Statement stmt = con4.createStatement();
            //Create Connection to the DB

            ResultSet tr = stmt.executeQuery( sql: "SELECT\n" +
                "    bills.isPaid,\n" +
                "    bills.BillID,\n" +
                "    bills.MoneyToPay,\n" +
                "    users.Username,\n" +
                "    customers.FirstName,\n" +
                "    customers.LastName,\n" +
                "    customers.AFM\n" +
                "FROM\n" +
                "    users\n" +
                "JOIN customers ON customers.UserID = users.UserID\n" +
                "JOIN bills ON bills.CustomerID = customers.CustomerID\n" +
                "JOIN soldplans ON soldplans.PlanID = bills.PlanID\n" +
                "WHERE\n" +
                "    users.Username = '" + ProvidedName + "'\n" +
                "    and bills.ispaid = 0 LIMIT 1 ;");

            //Execute query to import the users info
        }
    }
}
```



3 Βιβλιογραφικές Πηγές

1. <https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/201102161747563cexy8tj>(Για την κλάση Encryption)
2. <https://stackoverflow.com/questions/123657/how-can-i-share-a-variable-or-object-between-two-or-more-servlets>
3. <https://www.codejava.net/java-ee/servlet/handling-html-form-data-with-java-servlet>
4. <https://stackoverflow.com/questions/4971877/how-to-transfer-data-from-jsp-to-servlet-when-submitting-html-form>