Index

Subject Description:

The Identity Management software

This Identity Management software is quite basic. The main goal is to manage users of an Information System. As many basic concepts, it can be much improved, especially when you want to bring security to this management.

In computing, **identity management** describes the management of individual principals, their authentication, authorization, and privileges within or across system and enterprise boundaries with the goal of increasing security and productivity while decreasing cost, downtime and repetitive tasks. The terms "Identity Management" and "Identity and Access Management" (or IAM) are used interchangeably in the area of Identity access management, while identity management itself falls under the umbrella of IT Security.

Identity-management systems, products, applications and platforms manage identifying and ancillary data about entities that include individuals, computer-related hardware and applications.

The application will be able to :

- Access, create, update and delete user information
- Persist users data in a database.
- Capable of good performance
- Access, create, update and delete user information
- Persist users data in a database

Subject analysis:

1. **Major features:**

This application comport a scenario which authenticates a user, and makes him use the Identity management through predefined methods

- Create an Identity
- Update an Identity
- Delete an Identity

These steps have made use the very basics of the Java language.

2. Application Feasibility:

In the real-world context of engineering online systems, identity management can involve three basic functions:

The pure identity function:

Creation, management and deletion of identities without regard to access or entitlements.

User access:

User access enables users to assume a specific digital identity across applications, which enables access controls to be assigned and evaluated against this identity. The use of a single identity for a given user across multiple systems eases tasks for administrators and users. It simplifies access monitoring and verification and allows the organization to minimize excessive privileges granted to one user. User access can be tracked from initiation to termination of user access.

Services:

Organizations continue to add services for both internal users and by customers. Many such services require identity management to properly provide these services. Increasingly, identity management has been partitioned from application functions so that a single identity can serve many or even all of an organization's activities.

3. Data description:

For database we use MYSQL:

For identity management system we use, display name, email id, id, password as the access variable and columns in database.

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. The database which is used to create this application is SQL database. There is one database is used to include data into ,update data and delete data from tables and the login credentials and the user type whereas other stores the user details. The connector used in this application to connect to the database is "JDBC" connector.

4 .Expected results:

Expected result is based on activity diagram.

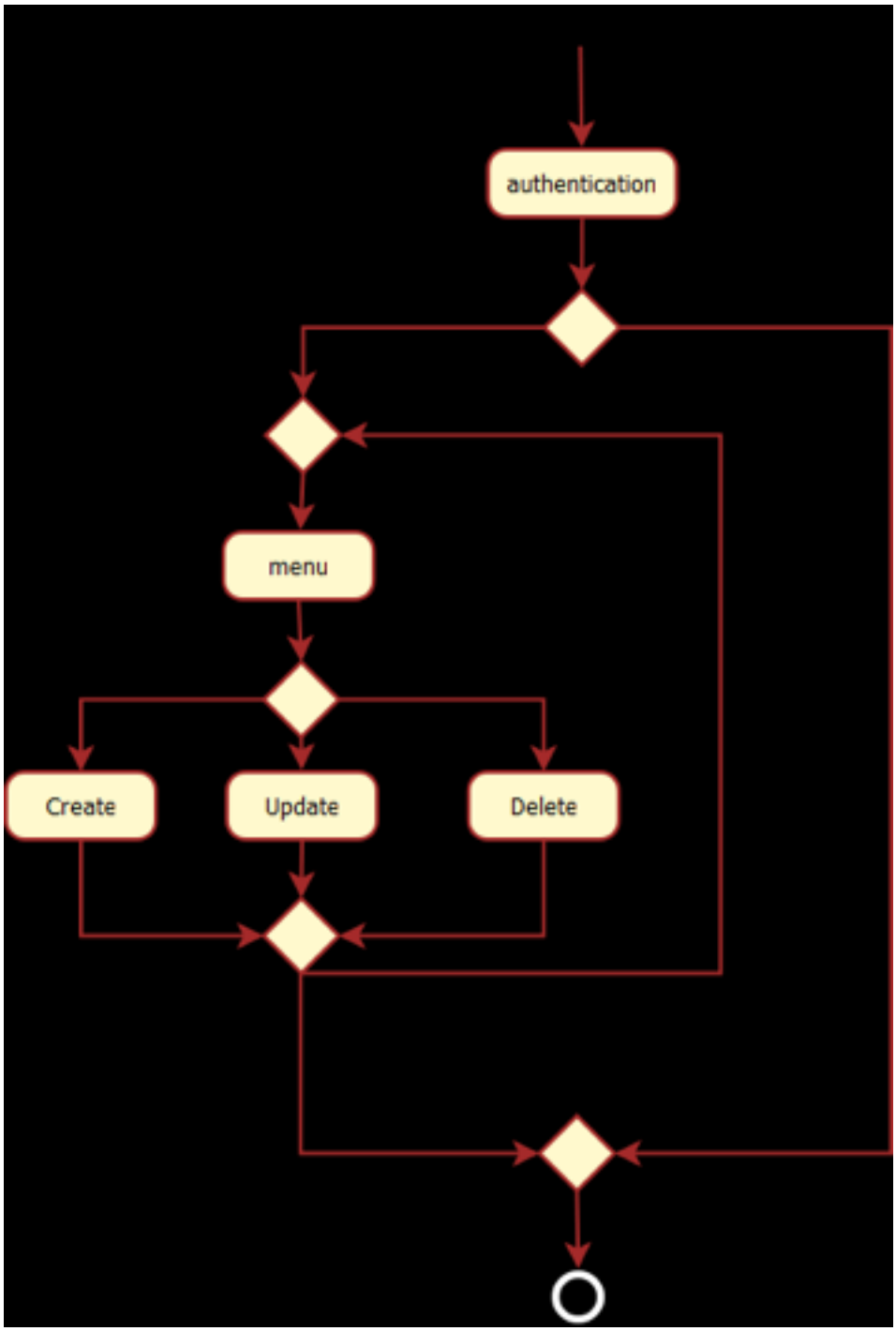
It is printed on console as I don't create GUI interface.

- **Scope of the application (limits, evolutions):**

The Identity Management project has been established to review the identity and access management processes.
The mechanisms that serve identity and access management of IT services can also be leveraged to manage more effectively access to services and facilities, e.g. library services, buildings.

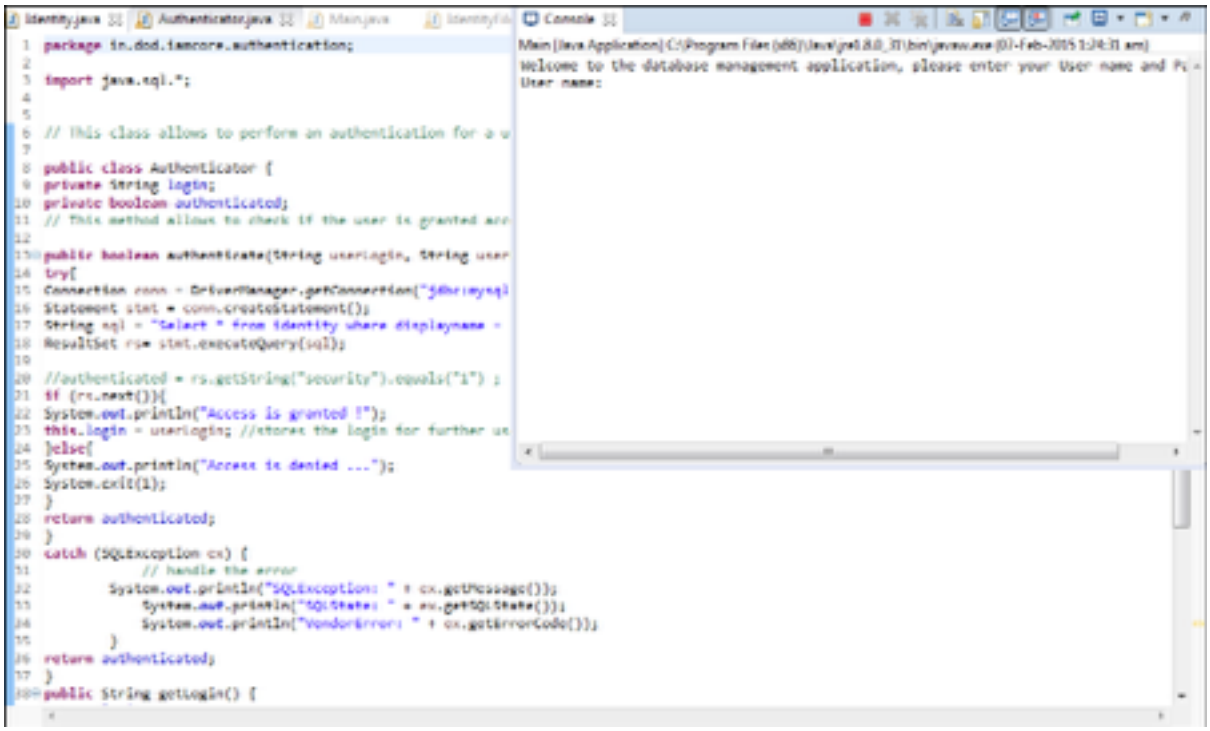
Conception:

Global application flow:

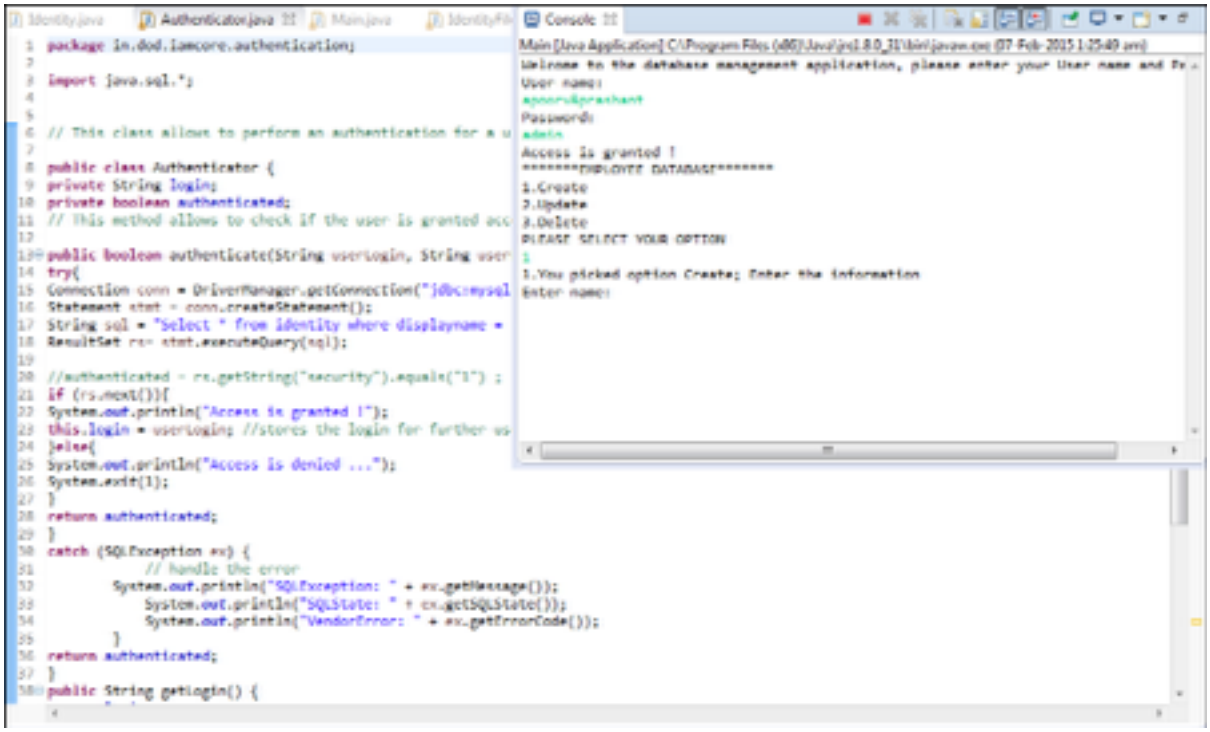


Console operations description:

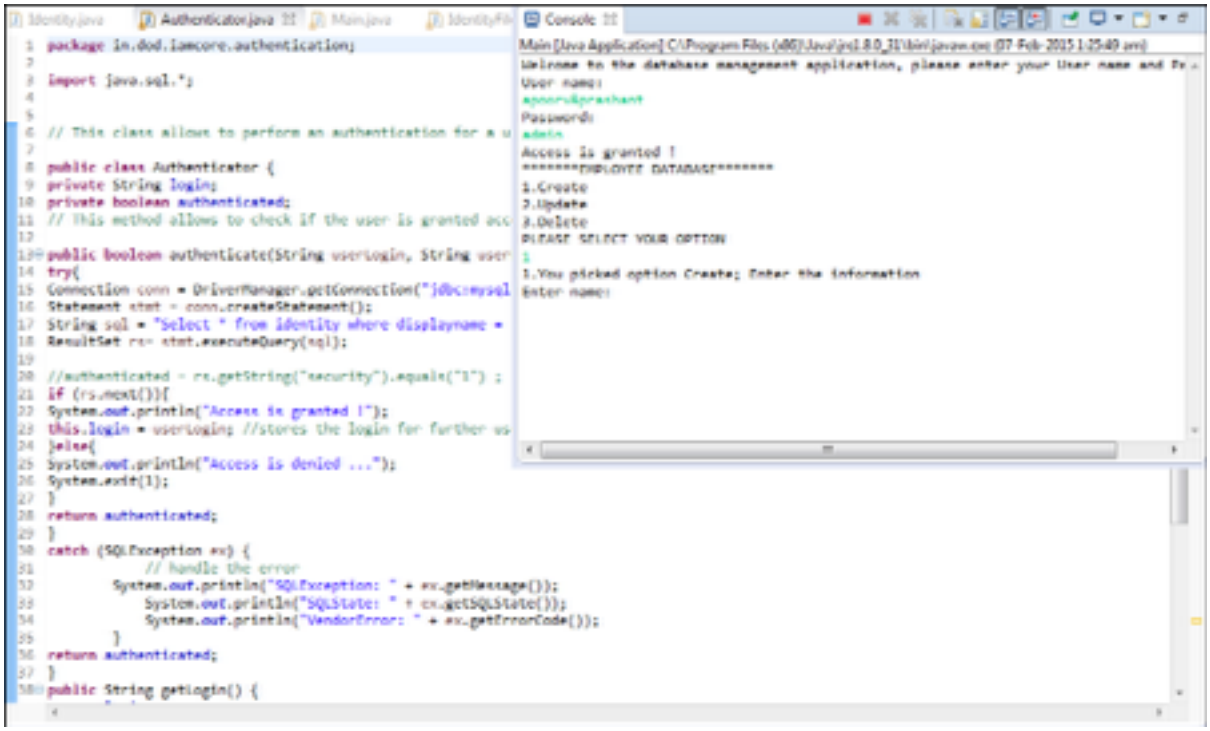
1. Login:



2. . Authentication using Login and Password.

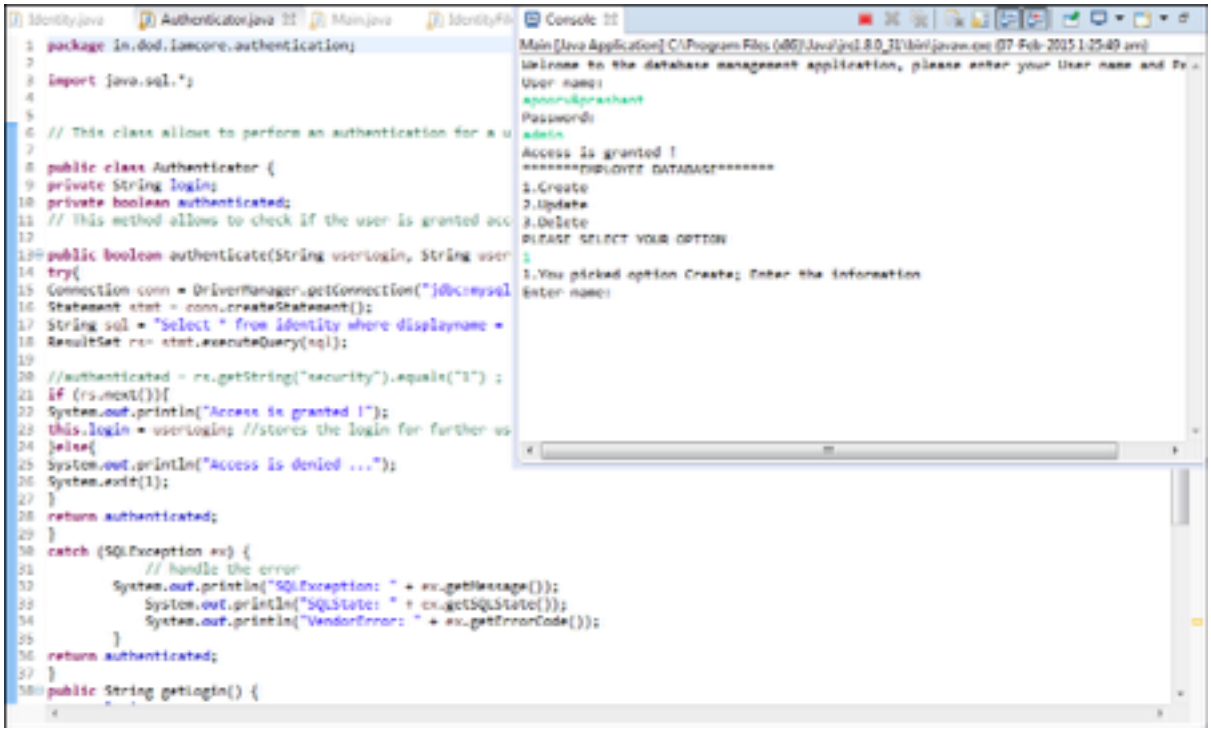


3.Connect to Database to check credentials.

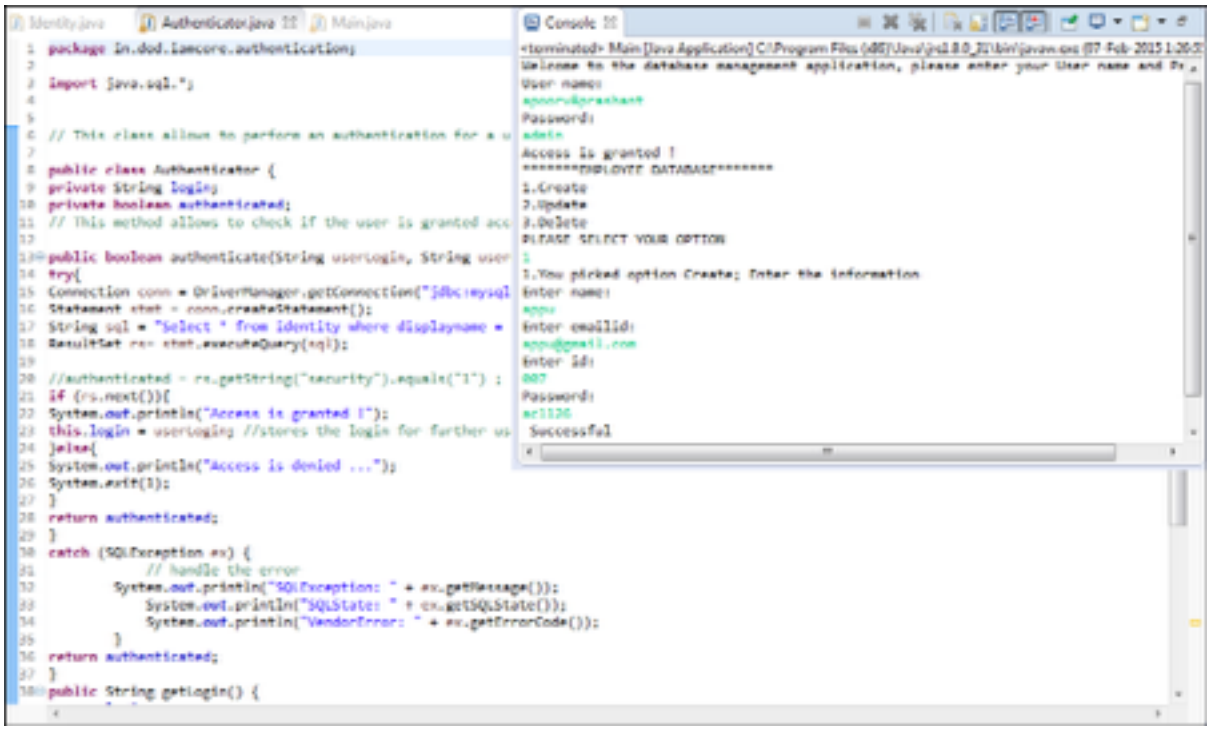


4.If Authentication fails start again.

5. If Authentication is successful then provide menu to Create, Update, Delete or Exit.



6. operation is successful



7. User can choose to continue or exit the application:

Configuration instructions:

Technology Used:-

The following technologies are used for the creating of this application:-

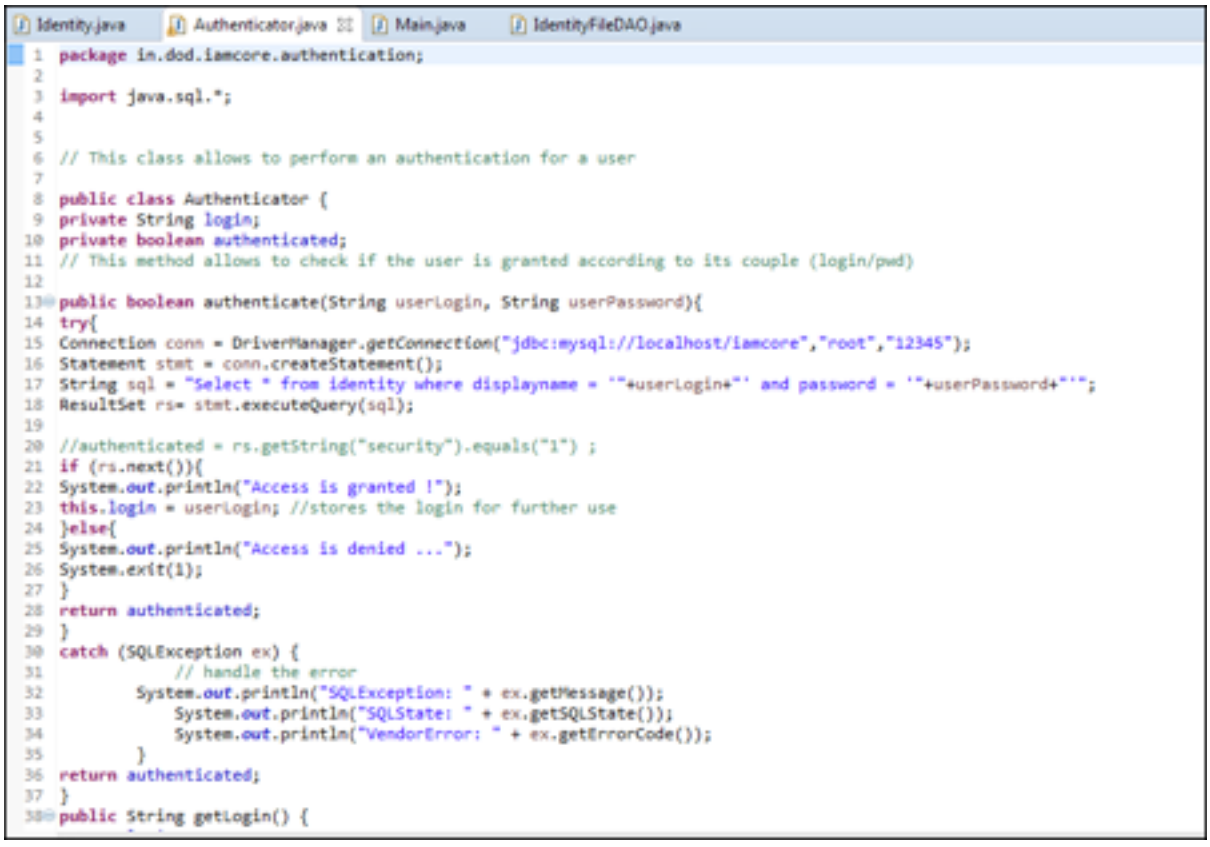
i. Eclipse IDE tool

ii.java version 1.7.0_45

iii.MySQL database

Commented Screenshot:

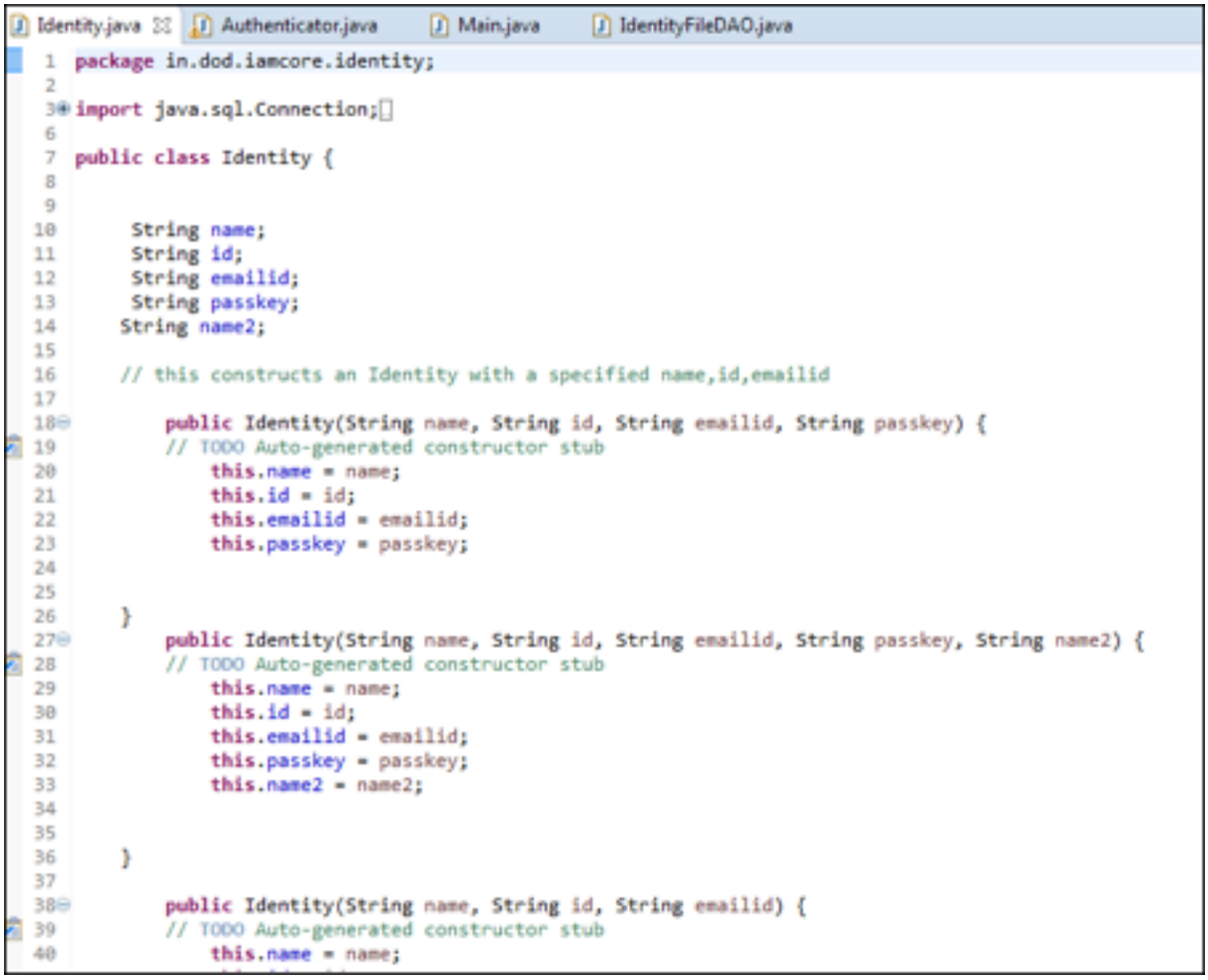
1.Authenticator,java



Identity.java | Authenticator.java | Main.java | IdentityFileDAO.java

```
1  package in.dod.iamcore.authentication;
2
3  import java.sql.*;
4
5
6  // This class allows to perform an authentication for a user
7
8  public class Authenticator {
9  private String login;
10 private boolean authenticated;
11 // This method allows to check if the user is granted according to its couple (login/pwd)
12
13 public boolean authenticate(String userLogin, String userPassword){
14 try{
15 Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/iamcore","root","12345");
16 Statement stmt = conn.createStatement();
17 String sql = "Select * from identity where displayname = '"+userLogin+"' and password = '"+userPassword+"'";
18 ResultSet rs= stmt.executeQuery(sql);
19
20 //authenticated = rs.getString("security").equals("1") ;
21 if (rs.next()){
22 System.out.println("Access is granted !");
23 this.login = userLogin; //stores the login for further use
24 }else{
25 System.out.println("Access is denied ...");
26 System.exit(1);
27 }
28 return authenticated;
29 }
30 catch (SQLException ex) {
31         // handle the error
32        System.out.println("SQLException: " + ex.getMessage());
33            System.out.println("SQLState: " + ex.getSQLState());
34            System.out.println("VendorError: " + ex.getErrorCode());
35        }
36 return authenticated;
37 }
38 public String getLogin() {
```

2.Identity.java:



```
Identity.java   Authenticator.java   Main.java   IdentityFileDAO.java
 1  package in.dod.iamcore.identity;
 2
 3  import java.sql.Connection;
 6
 7  public class Identity {
 8
 9
10      String name;
11      String id;
12      String emailid;
13      String passkey;
14     String name2;
15
16     // this constructs an Identity with a specified name,id,emailid
17
18         public Identity(String name, String id, String emailid, String passkey) {
19         // TODO Auto-generated constructor stub
20             this.name = name;
21             this.id = id;
22             this.emailid = emailid;
23             this.passkey = passkey;
24
25
26     }
27         public Identity(String name, String id, String emailid, String passkey, String name2) {
28         // TODO Auto-generated constructor stub
29             this.name = name;
30             this.id = id;
31             this.emailid = emailid;
32             this.passkey = passkey;
33             this.name2 = name2;
34
35
36     }
37
38         public Identity(String name, String id, String emailid) {
39         // TODO Auto-generated constructor stub
40             this.name = name;
```

3:Main.java

```
package in.dod.iamcore.Main;

import in.dod.iamcore.authentication.*;

public class Main {

    public static void main(String[] args) {
        //Iamlog logger = new Iamlog("Main");
        String passkey=null;

        try{
        IdentityDAO identityDAO = new IdentityFileDAO();

        //logger.log("Beginning of the program", "INFO");
        Scanner scan = new Scanner(System.in);

        System.out.println("Welcome to the database management application, please enter your User name and Password");
        System.out.println("User name:");

        String userName = scan.nextLine();
        //scan.nextLine();

        System.out.println("Password:");

        String password = scan.nextLine();
        //scan.nextLine();
        Authenticator authenticator = new Authenticator();

        /*logger.log("try to authenticate the user with this couple " + userName
                + " / " + password);*/
        boolean isAuthenticated = authenticator
                .authenticate(userName, password);

        /*if (isAuthenticated) {
            System.out.println("You are logged in");
```

4:Databaseconectivity.java:

```
package in.dod.iamcore.Main;

import java.sql.*;

public class Databaseconnection {
    public static void main(String[] args) {

            Scanner scn = new Scanner(System.in);
              try {

              Connection conn1 = DriverManager.getConnection("jdbc:mysql://localhost/Identity?"
              +"user=root&password=admin");

              System.out.println("Welcome");
              System.out.print("select the name: ");
                String name = scn.nextLine();

                System.out.print("Enter name: ");
                String name1 = scn.nextLine();

                System.out.print("Enter emailId: ");
                String emailId = scn.nextLine();

                System.out.print("Enter id: ");
                 String id = scn.nextLine();

                Statement stmt = conn1.createStatement();
               //insert into table
                String sql = "insert into Database " +
                    "VALUES ('"+name1+"','"+emailId+"','"+id+"')";
                stmt.executeUpdate(sql);
               System.out.println(" Successfull\n");

               //update table
              String sql2="update Identity; set  name='"+name1+"', emailId='"+emailId+"',Id='"+id+"' WHERE  name='"+name+"'";
              stmt.executeUpdate(sql2);
              System.out.println(" Update Successful\n");
```

5: IdentityDao.java

Bibliography:

1. **SCJP java book.**
2. **http://www.roseindia.net/programming-tutorial/core-java**
3. **Youtube java tutorial**
4. www.javatpoint.com