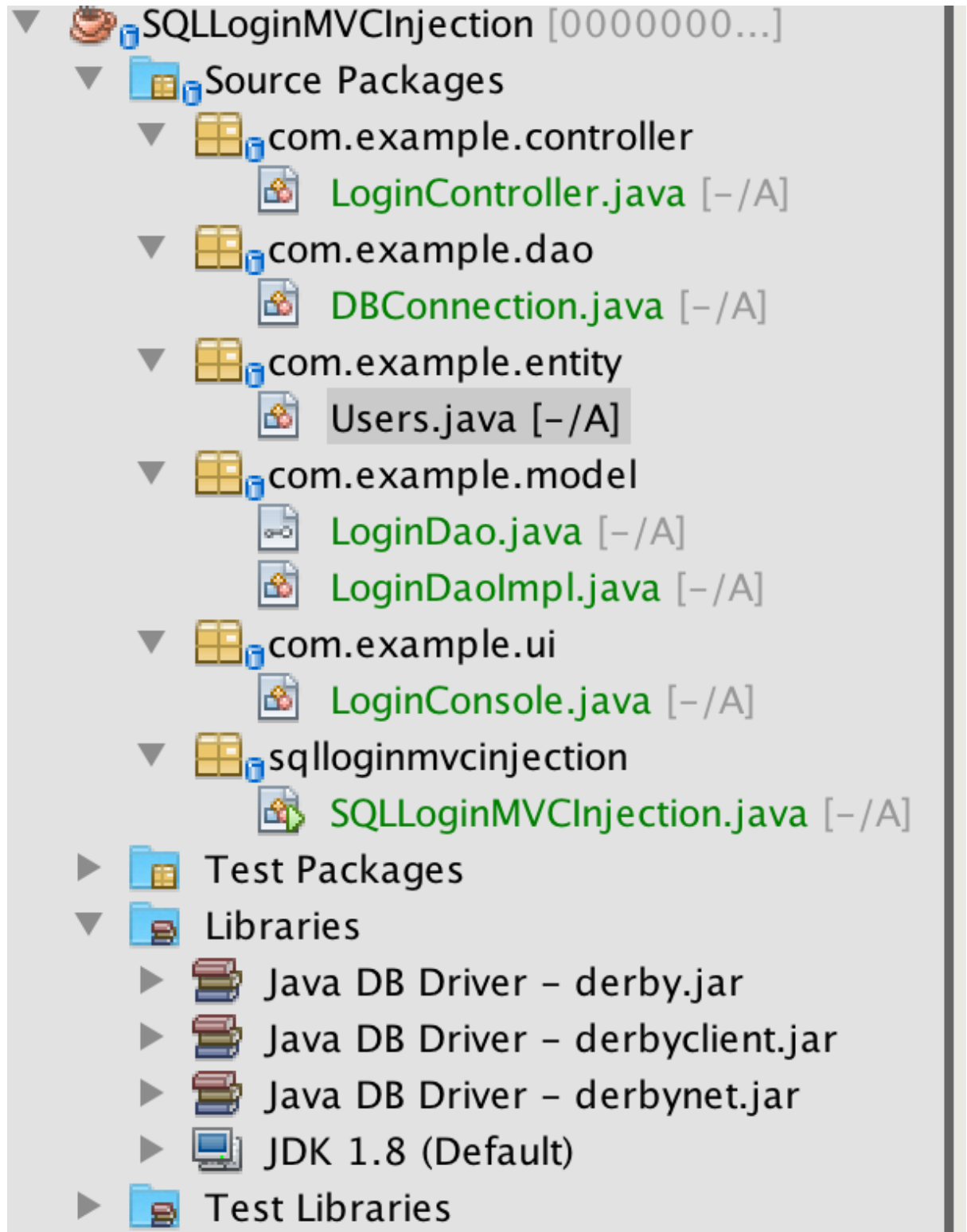
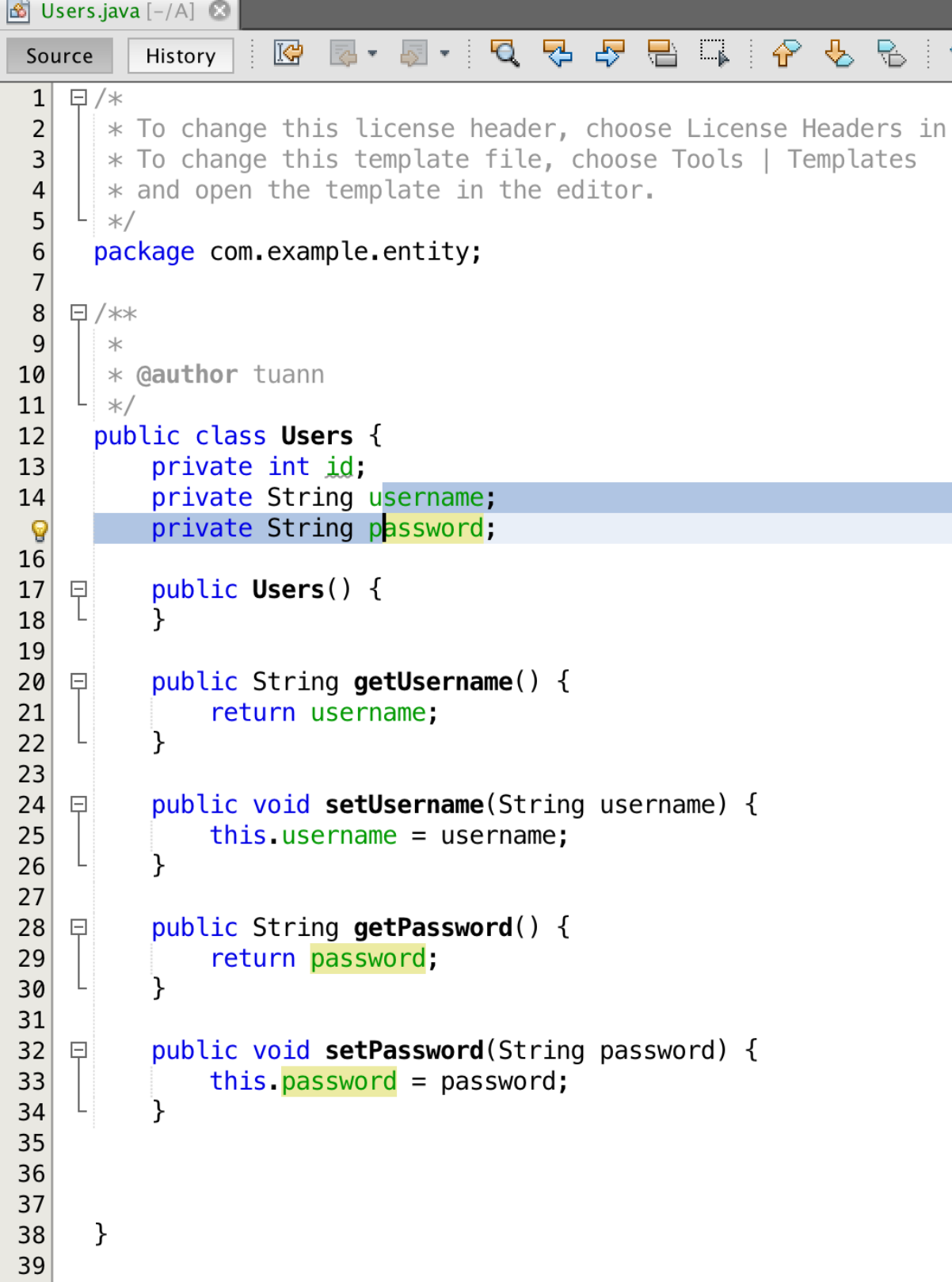


# MVC design pattern login demo với PreparedStatement & Statement

Tạo kiến trúc Project như hình dưới:

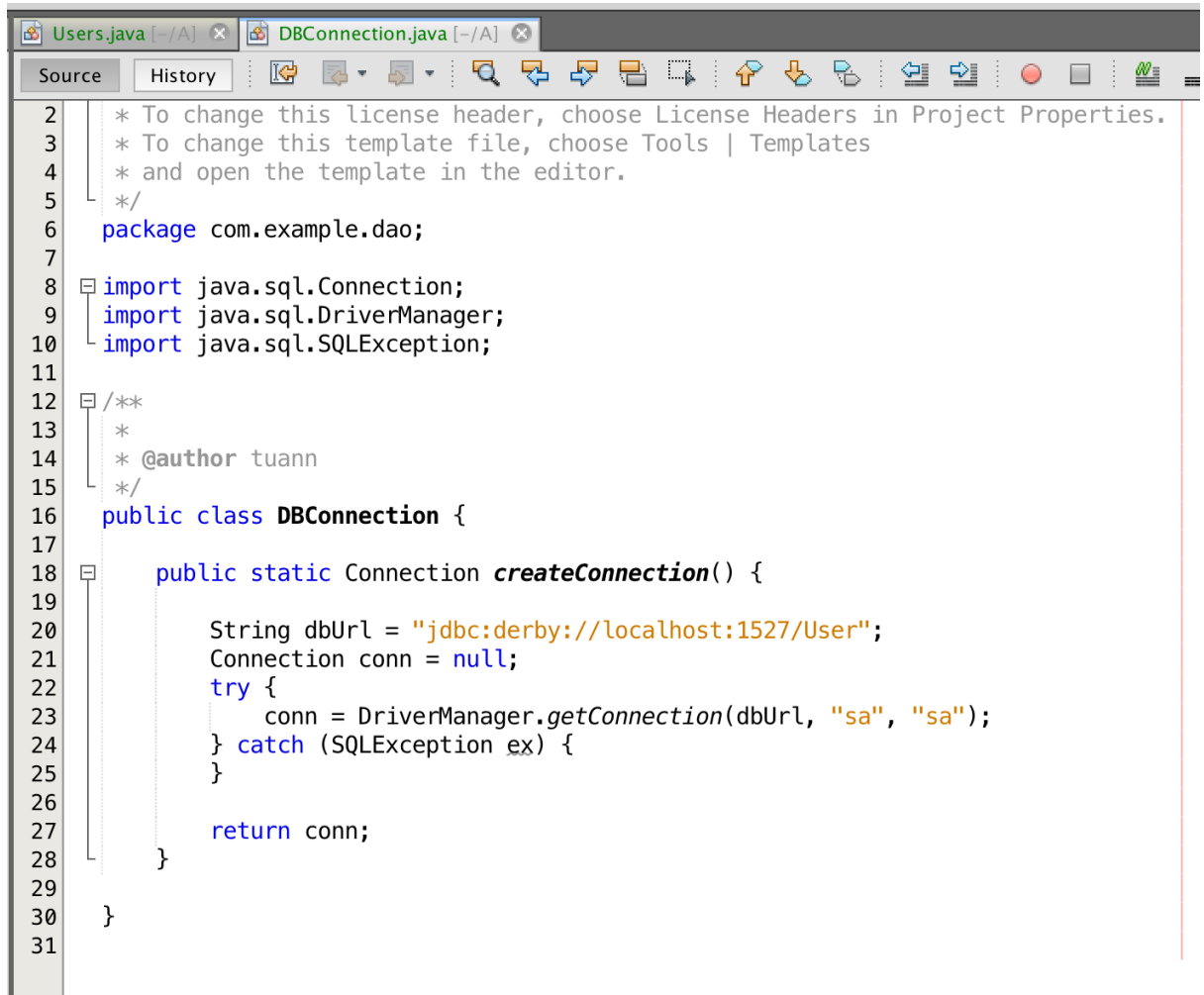


# 1. Tạo entity



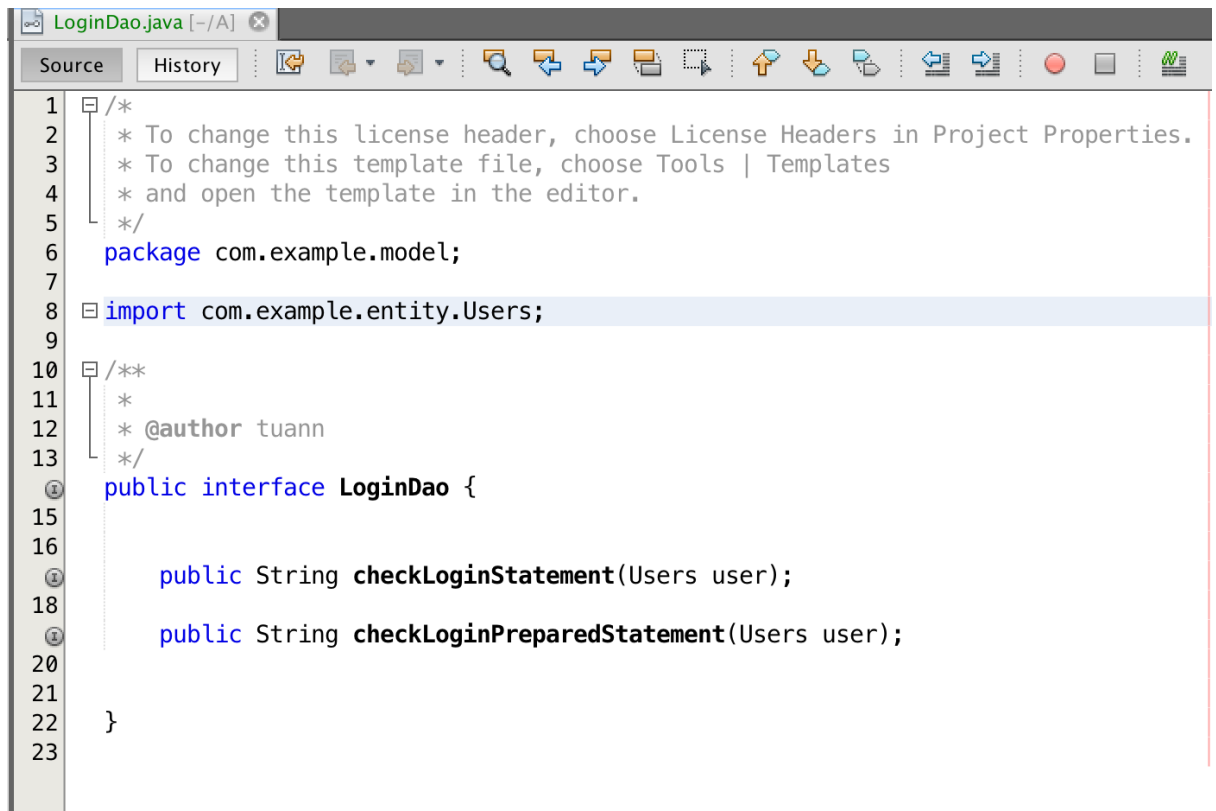
```
1  /*
2   * To change this license header, choose License Headers in
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package com.example.entity;
7
8  /**
9   *
10  * @author tuann
11  */
12  public class Users {
13      private int id;
14      private String username;
15      private String password;
16
17      public Users() {
18      }
19
20      public String getUsername() {
21          return username;
22      }
23
24      public void setUsername(String username) {
25          this.username = username;
26      }
27
28      public String getPassword() {
29          return password;
30      }
31
32      public void setPassword(String password) {
33          this.password = password;
34      }
35
36
37  }
38
39
```

## 2. Tạo DBConnection class sử dụng để kết nối với database Derby.



```
1  * To change this license header, choose License Headers in Project Properties.
2  * To change this template file, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package com.example.dao;
7
8  import java.sql.Connection;
9  import java.sql.DriverManager;
10 import java.sql.SQLException;
11
12 /**
13  *
14  * @author tuann
15  */
16 public class DBConnection {
17
18     public static Connection createConnection() {
19
20         String dbUrl = "jdbc:derby://localhost:1527/User";
21         Connection conn = null;
22         try {
23             conn = DriverManager.getConnection(dbUrl, "sa", "sa");
24         } catch (SQLException ex) {
25
26         }
27
28         return conn;
29     }
30 }
31
```

- Tạo interface **LoginDao**: Interface này giúp chúng ta định nghĩa các phương thức mà bài toán yêu cầu. Với bài toán phân tán(Distributed) thì đây sẽ là phương thức mà các chương trình khác(nằm trên các server khác nhau) có thể tương tác thông qua các interface này. Tức là các chương trình đó chỉ có thể nhìn thấy interface chứ không phải là các class biz implement code.



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package com.example.model;
7
8  import com.example.entity.Users;
9
10 /*
11  *
12  * @author tuann
13  */
14 public interface LoginDao {
15
16
17     public String checkLoginStatement(Users user);
18
19     public String checkLoginPreparedStatement(Users user);
20
21 }
22
23
```

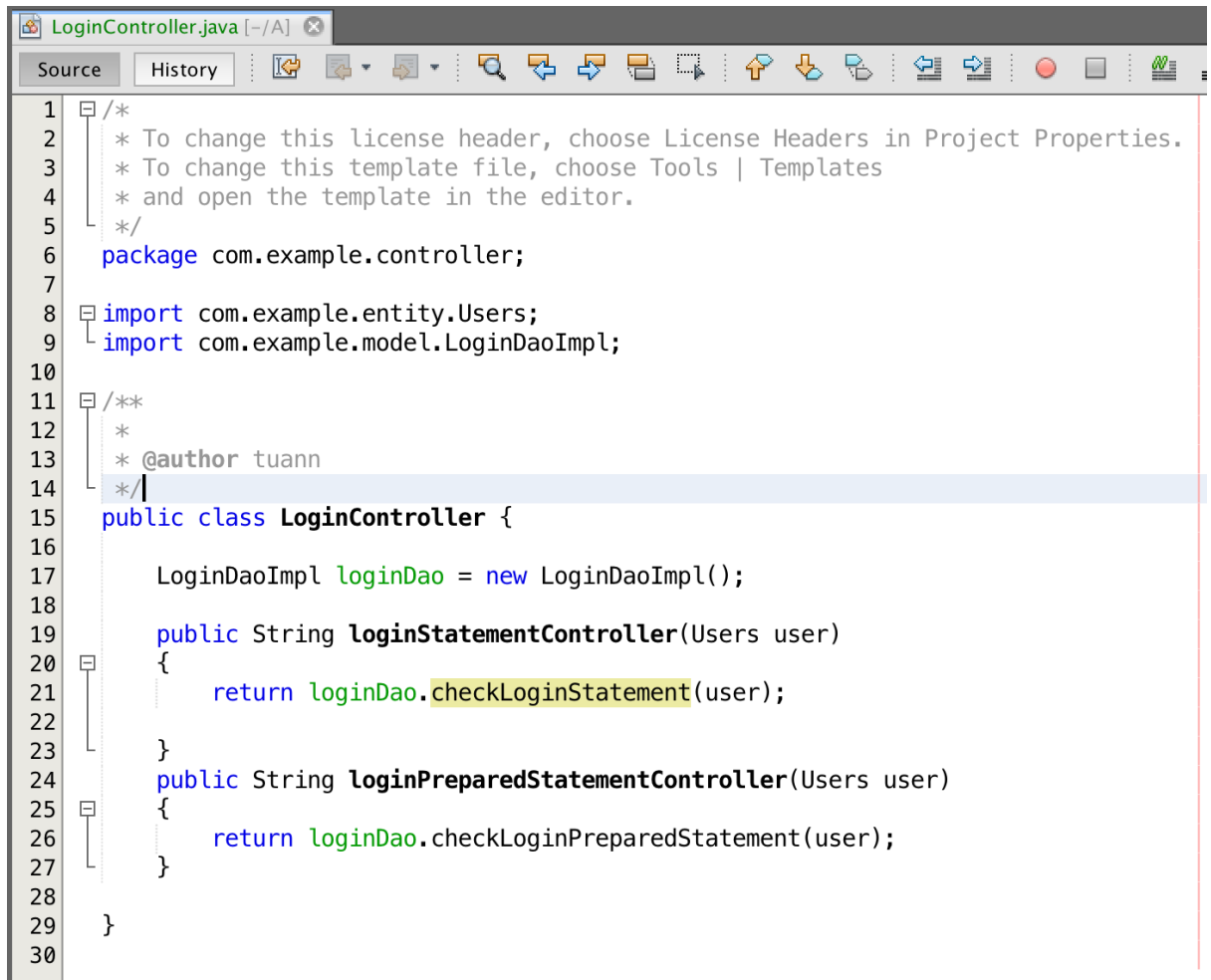
- Tạo class **LoginDaoImpl** implement interface nơi định nghĩa các biz methods. Ở class này sẽ là nơi thi triển code.

```
LoginDaoImpl.java [-/A]
Source History
1 package com.example.model;
2
3 import com.example.dao.DBConnection;
4 import com.example.entity.Users;
5 import java.sql.Connection;
6 import java.sql.PreparedStatement;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.sql.Statement;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12
13 /**
14  *
15  * @author tuann
16  */
17 public class LoginDaoImpl implements LoginDao {
18
19     private static final Connection conn = DBConnection.createConnection();
20     private static Statement stmt = null;
21     private static PreparedStatement pstmt = null;
22
23     @Override
24     public String checkLoginStatement(Users user) {
25
26         String dbQuery = "SELECT username FROM users WHERE username ='" + user.getUsername() + "'"
27             + " and password ='" + user.getPassword() + "' ";
28         try {
29             stmt = conn.createStatement();
30             ResultSet rs = stmt.executeQuery(dbQuery);
31             while (rs.next()) {
32                 System.out.println("Username is: " + rs.getString("username"));
33                 return rs.getString("username");
34             }
35             return "Not in the database";
36         } catch (SQLException ex) {
37             Logger.getLogger(LoginDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
38             return "false";
39         }
40     }
41
42 }
43
44 @Override
45 public String checkLoginPreparedStatement(Users user) {
46     String dbQuery = "SELECT username FROM users WHERE username like ? and password like ?";
47     try {
48         pstmt = conn.prepareStatement(dbQuery);
49         pstmt.setString(1, user.getUsername());
50         pstmt.setString(2, user.getPassword());
51         ResultSet rs = pstmt.executeQuery();
52         while(rs.next())
53         {
54             System.out.println("Username is: "+ rs.getString("username"));
55             return rs.getString("username");
56         }
57         return "not in the database";
58     } catch (SQLException ex) {
59         Logger.getLogger(LoginDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
60         return "false";
61     }
62 }
63
64 }
65
```

**3. Tạo controller: Controller sẽ tiếp nhận yêu cầu  
-> gọi model-> Dispatcher(đánh tiếng)-> View(**

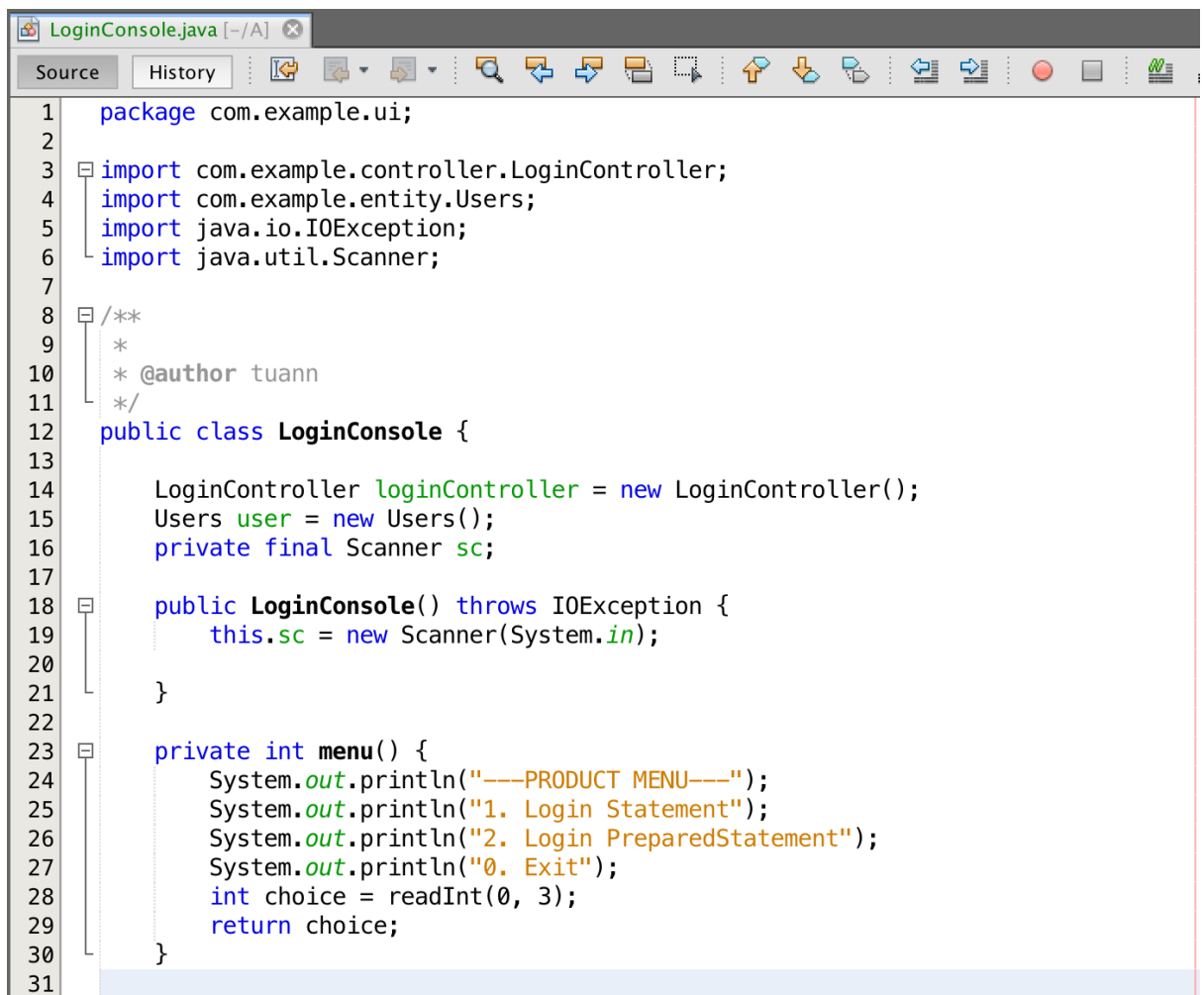
để view biết sẽ nhận được gì, hoặc gọi cái gì từ model)

- **LoginController class**



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package com.example.controller;
7
8  import com.example.entity.Users;
9  import com.example.model.LoginDaoImpl;
10
11 /**
12  *
13  * @author tuann
14  */
15 public class LoginController {
16
17     LoginDaoImpl loginDao = new LoginDaoImpl();
18
19     public String loginStatementController(Users user)
20     {
21         return loginDao.checkLoginStatement(user);
22     }
23
24     public String loginPreparedStatementController(Users user)
25     {
26         return loginDao.checkLoginPreparedStatement(user);
27     }
28
29 }
30
```

#### 4. Tạo ui(view) -> Tương tác với người dùng -> Controller -> Model ->View



```
1 package com.example.ui;
2
3 import com.example.controller.LoginController;
4 import com.example.entity.Users;
5 import java.io.IOException;
6 import java.util.Scanner;
7
8 /**
9  *
10  * @author tuann
11  */
12 public class LoginConsole {
13
14     LoginController loginController = new LoginController();
15     Users user = new Users();
16     private final Scanner sc;
17
18     public LoginConsole() throws IOException {
19         this.sc = new Scanner(System.in);
20     }
21
22     private int menu() {
23         System.out.println("----PRODUCT MENU----");
24         System.out.println("1. Login Statement");
25         System.out.println("2. Login PreparedStatement");
26         System.out.println("0. Exit");
27         int choice = readInt(0, 3);
28         return choice;
29     }
30
31 }
```

```

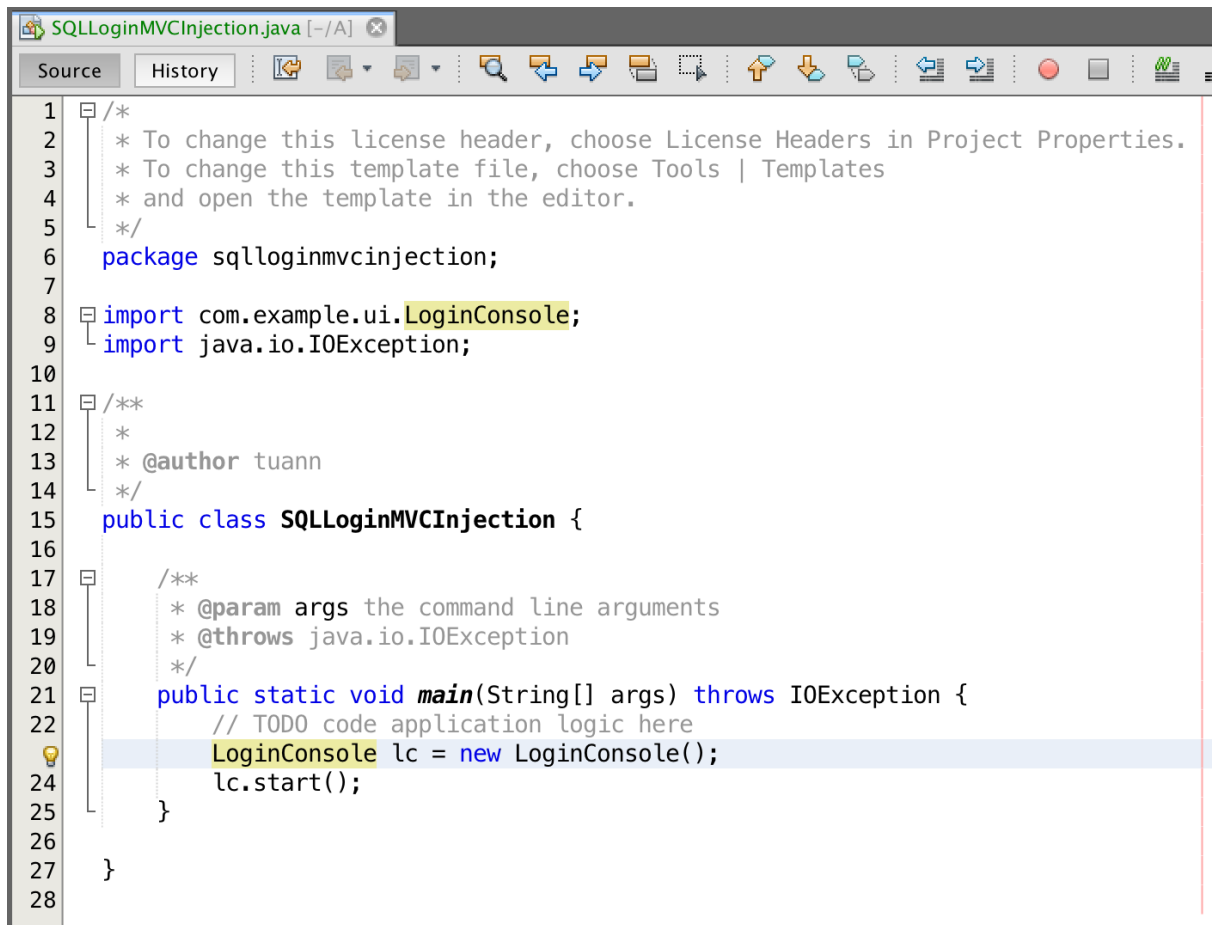
32 public void start() {
33     while (true) {
34         int choice = menu();
35         switch (choice) {
36             case 0:
37                 System.exit(0);
38                 break;
39             case 1:
40                 loginStatement();
41                 break;
42             case 2:
43                 loginPreparedStatement();
44                 break;
45             // case 3:
46             //     removeProduct();
47             //     break;
48             default:
49                 throw new AssertionError();
50         }
51     }
52 }
53
54 private int readInt(int min, int max) {
55     int choice;
56     while (true) {
57         try {
58             choice = Integer.parseInt(sc.nextLine());
59             if (choice >= min && choice <= max) {
60                 break;
61             }
62         } catch (NumberFormatException e) {
63         }
64     }
65     return choice;
66 }
67

```



```
68 private void loginStatement() {
69
70     System.out.println("Enter username:");
71     String username = sc.nextLine();
72     System.out.println("Enter password: ");
73     String password = sc.nextLine();
74     user.setUsername(username);
75     user.setPassword(password);
76     String result = loginController.loginStatementController(user);
77     System.out.println(result);
78
79 }
80
81 private void loginPreparedStatement() {
82
83     System.out.println("Enter username:");
84     String username = sc.nextLine();
85     System.out.println("Enter password: ");
86     String password = sc.nextLine();
87     user.setUsername(username);
88     user.setPassword(password);
89     String result = loginController.loginPreparedStatementController(user);
90     System.out.println(result);
91
92 }
93
94 }
95
```

## Finally Start and run với class main:



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package sqlloginmvcinjection;
7
8   import com.example.ui.LoginConsole;
9   import java.io.IOException;
10
11  /**
12   *
13   * @author tuann
14   */
15  public class SQLLoginMVCInjection {
16
17      /**
18       * @param args the command line arguments
19       * @throws java.io.IOException
20       */
21      public static void main(String[] args) throws IOException {
22          // TODO code application logic here
23          LoginConsole lc = new LoginConsole();
24          lc.start();
25      }
26
27  }
28
```