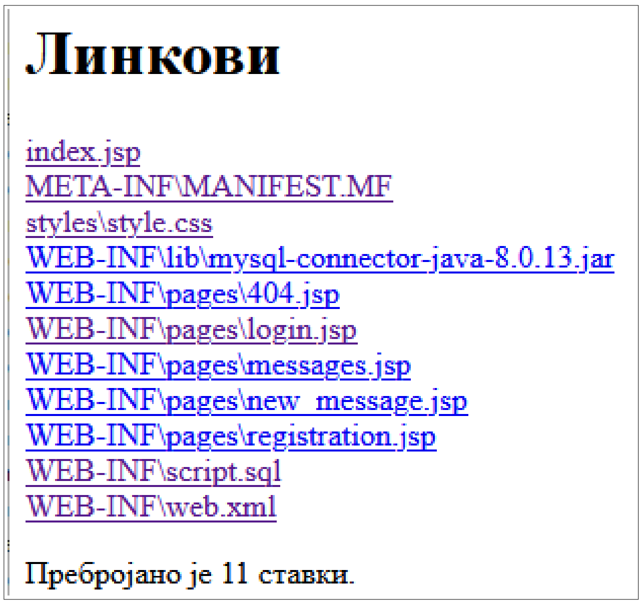
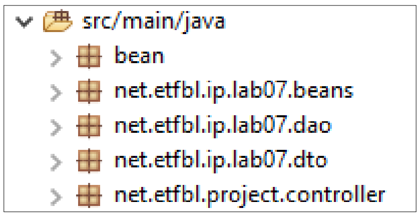
INTERNET PROGRAMIRANJE
VJEZBE – 07 – JSP M2
VIDEO I MATERIJALI
PREGLED I ANALIZA



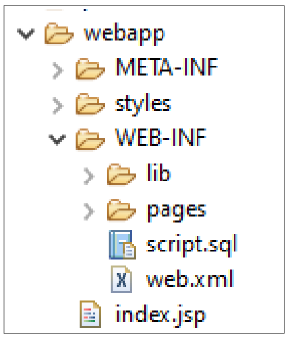
Линкови

index.jsp
META-INF\MANIFEST.MF
styles\style.css
WEB-INF\lib\mysql-connector-java-8.0.13.jar
WEB-INF\pages\404.jsp
WEB-INF\pages\login.jsp
WEB-INF\pages\messages.jsp
WEB-INF\pages\new_message.jsp
WEB-INF\pages\registration.jsp
WEB-INF\script.sql
WEB-INF\web.xml

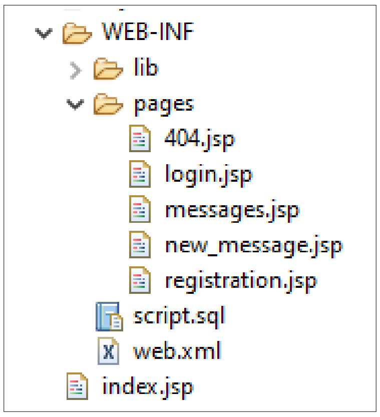
Пребројано је 11 ставки.



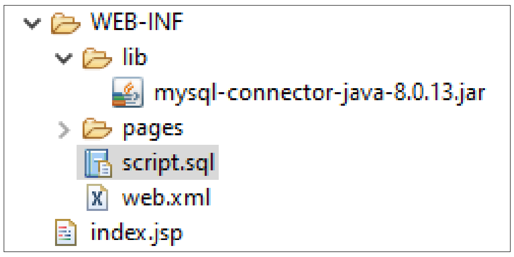
```
src/main/java
  bean
  net.etfbl.ip.lab07.beans
  net.etfbl.ip.lab07.dao
  net.etfbl.ip.lab07.dto
  net.etfbl.project.controller
```



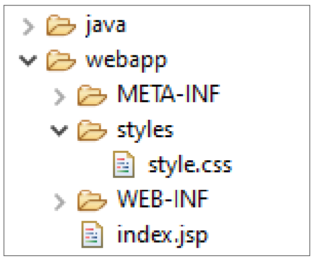
```
webapp
  META-INF
  styles
  WEB-INF
    lib
    pages
    script.sql
    web.xml
  index.jsp
```



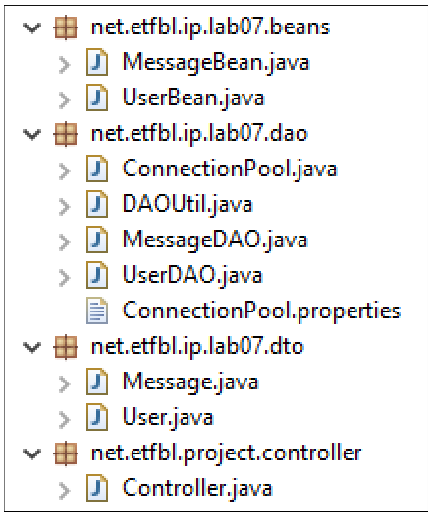
```
WEB-INF
  lib
  pages
    404.jsp
    login.jsp
    messages.jsp
    new_message.jsp
    registration.jsp
  script.sql
  web.xml
index.jsp
```



```
WEB-INF
    lib
        mysql-connector-java-8.0.13.jar
    pages
    script.sql
    web.xml
index.jsp
```



```
java
webapp
    META-INF
    styles
        style.css
    WEB-INF
    index.jsp
```

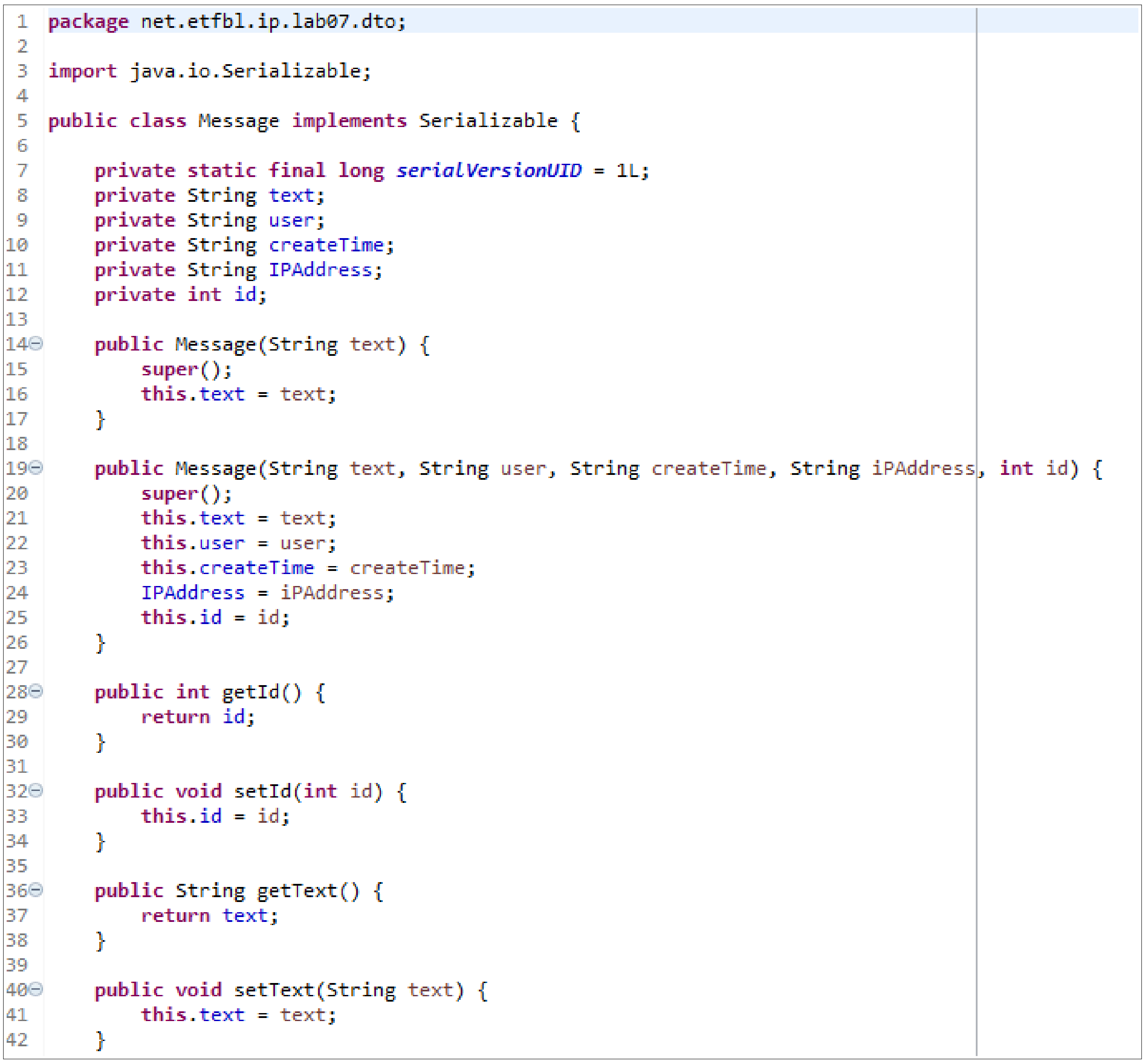


```
net.etfbl.ip.lab07.beans
    MessageBean.java
    UserBean.java
net.etfbl.ip.lab07.dao
    ConnectionPool.java
    DAOUtil.java
    MessageDAO.java
    UserDAO.java
    ConnectionPool.properties
net.etfbl.ip.lab07.dto
    Message.java
    User.java
net.etfbl.project.controller
    Controller.java
```

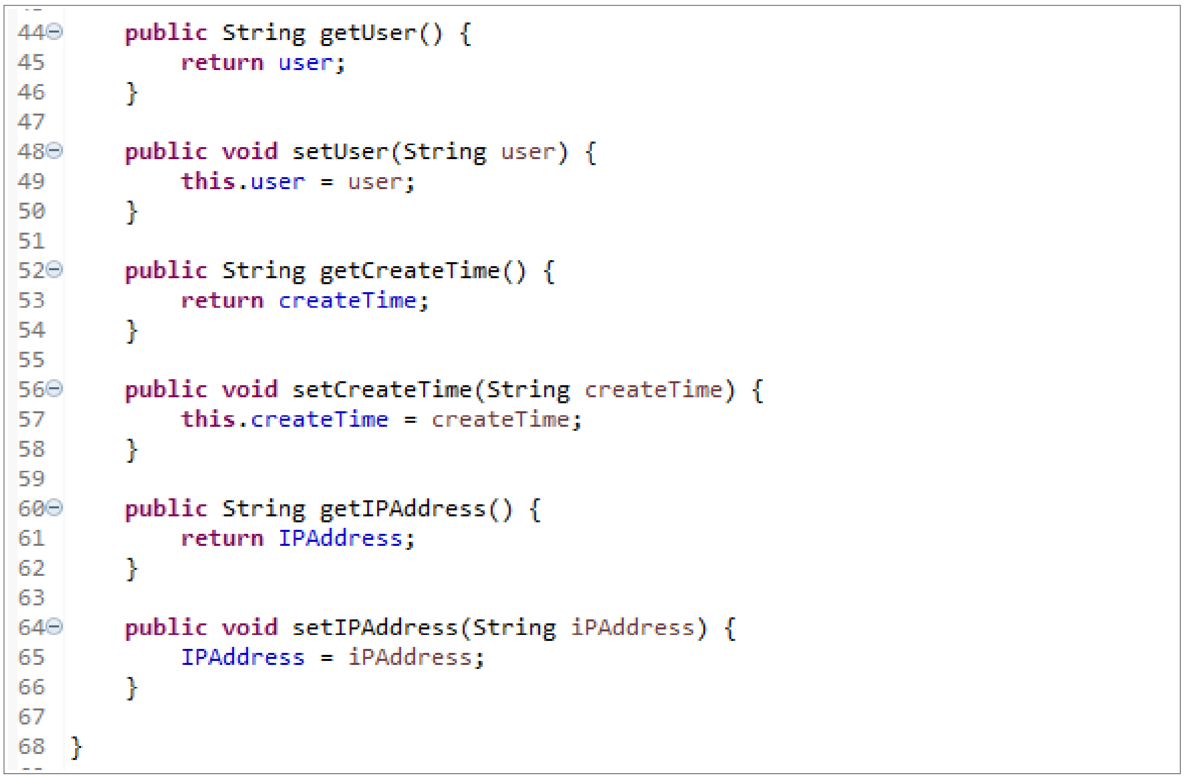
SCRIPT.SQL

```sql
create schema etf_lab07;
use etf_lab07;

create table user (
    id int auto_increment primary key,
    username varchar(50) not null,
    password varchar(50) not null,
    first_name varchar(50) not null,
    last_name varchar(50) not null
);

create table message (
    id int auto_increment primary key,
    text varchar(50) not null,
    user varchar(50) not null,
    create_time varchar(50) not null,
    ip_address varchar(50) not null
);
```

MESSAGE.JAVA

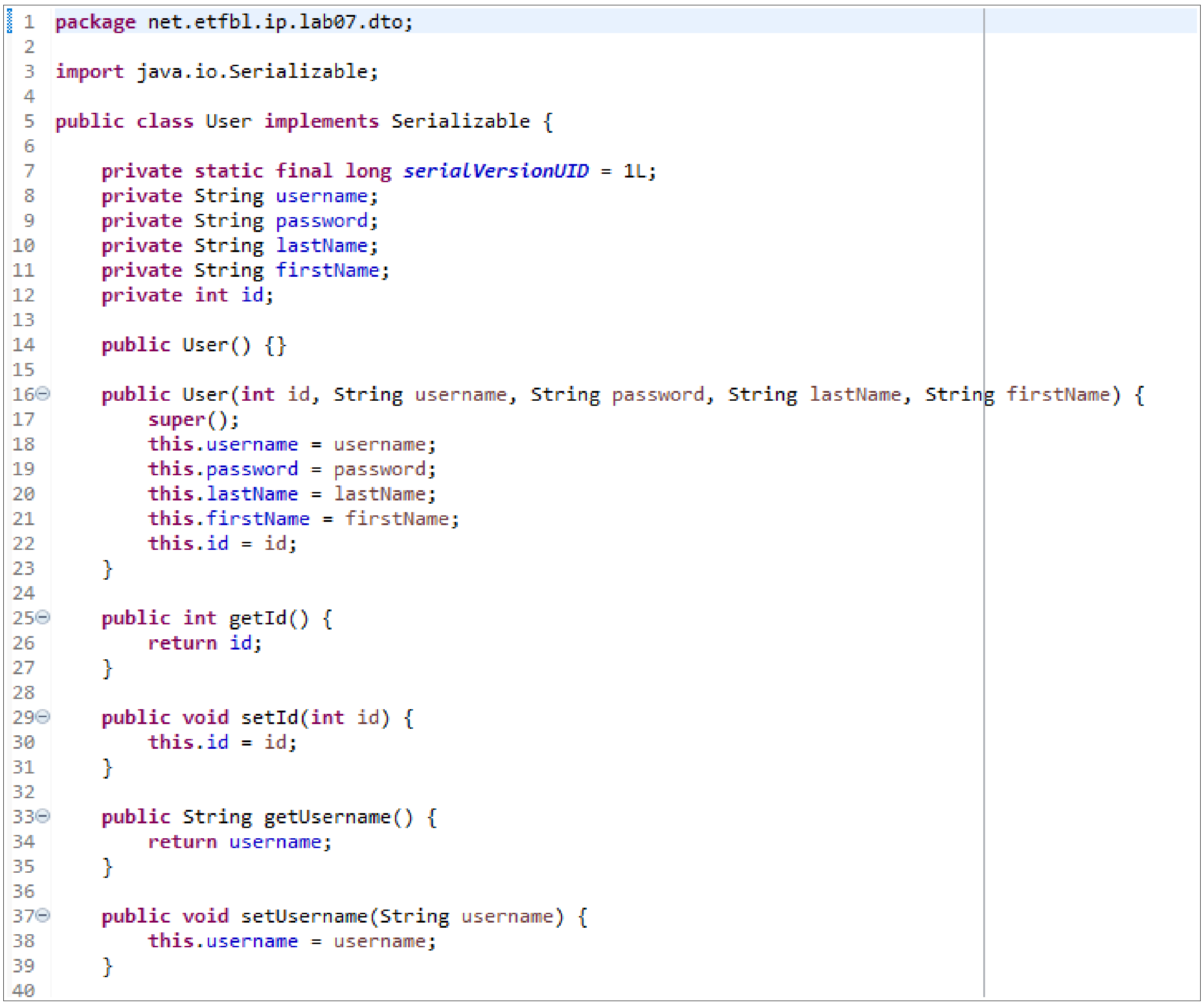


```java
package net.etfbl.ip.lab07.dto;

import java.io.Serializable;

public class Message implements Serializable {

    private static final long serialVersionUID = 1L;
    private String text;
    private String user;
    private String createTime;
    private String IPAddress;
    private int id;

    public Message(String text) {
        super();
        this.text = text;
    }

    public Message(String text, String user, String createTime, String iPAddress, int id) {
        super();
        this.text = text;
        this.user = user;
        this.createTime = createTime;
        IPAddress = iPAddress;
        this.id = id;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getText() {
        return text;
    }

    public void setText(String text) {
        this.text = text;
    }
```

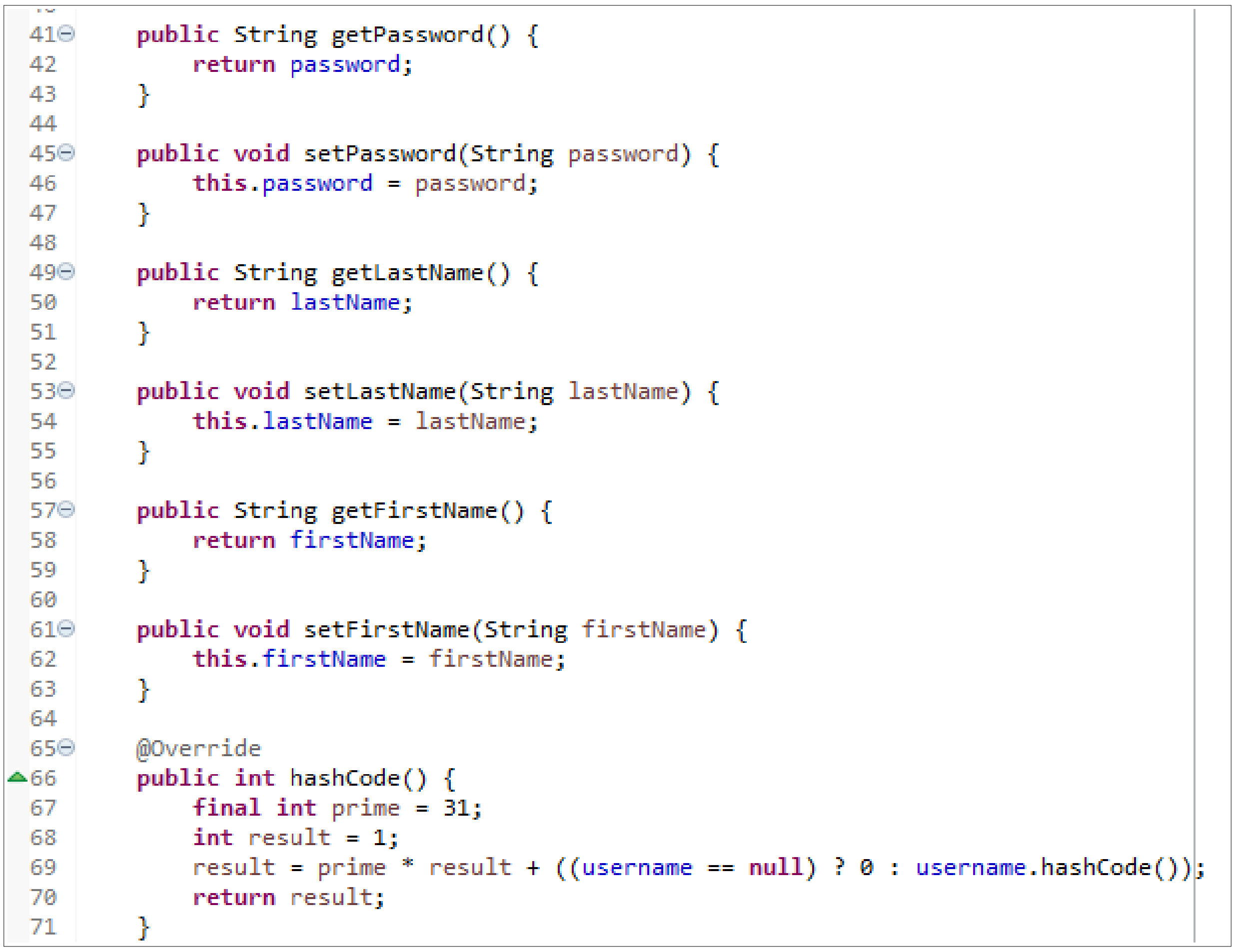


```java
    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    public String getCreateTime() {
        return createTime;
    }

    public void setCreateTime(String createTime) {
        this.createTime = createTime;
    }

    public String getIPAddress() {
        return IPAddress;
    }

    public void setIPAddress(String iPAddress) {
        IPAddress = iPAddress;
    }

}
```

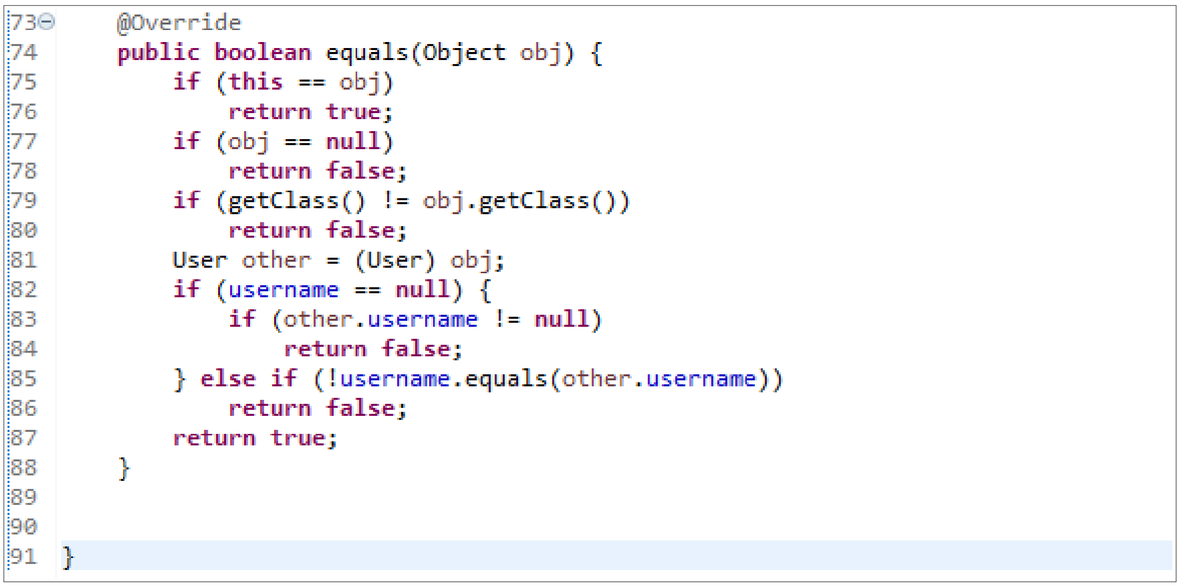
USER.JAVA



```java
package net.etfbl.ip.lab07.dto;

import java.io.Serializable;

public class User implements Serializable {

    private static final long serialVersionUID = 1L;
    private String username;
    private String password;
    private String lastName;
    private String firstName;
    private int id;

    public User() {}

    public User(int id, String username, String password, String lastName, String firstName) {
        super();
        this.username = username;
        this.password = password;
        this.lastName = lastName;
        this.firstName = firstName;
        this.id = id;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

```

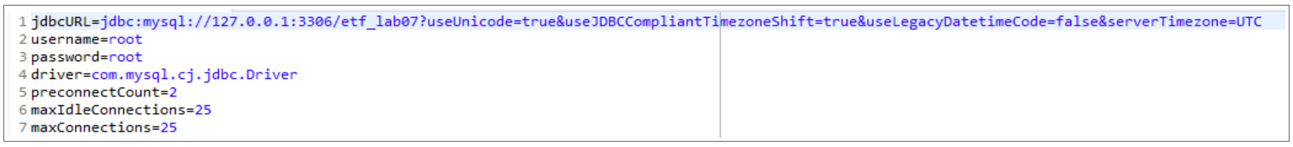


```java
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((username == null) ? 0 : username.hashCode());
        return result;
    }
```



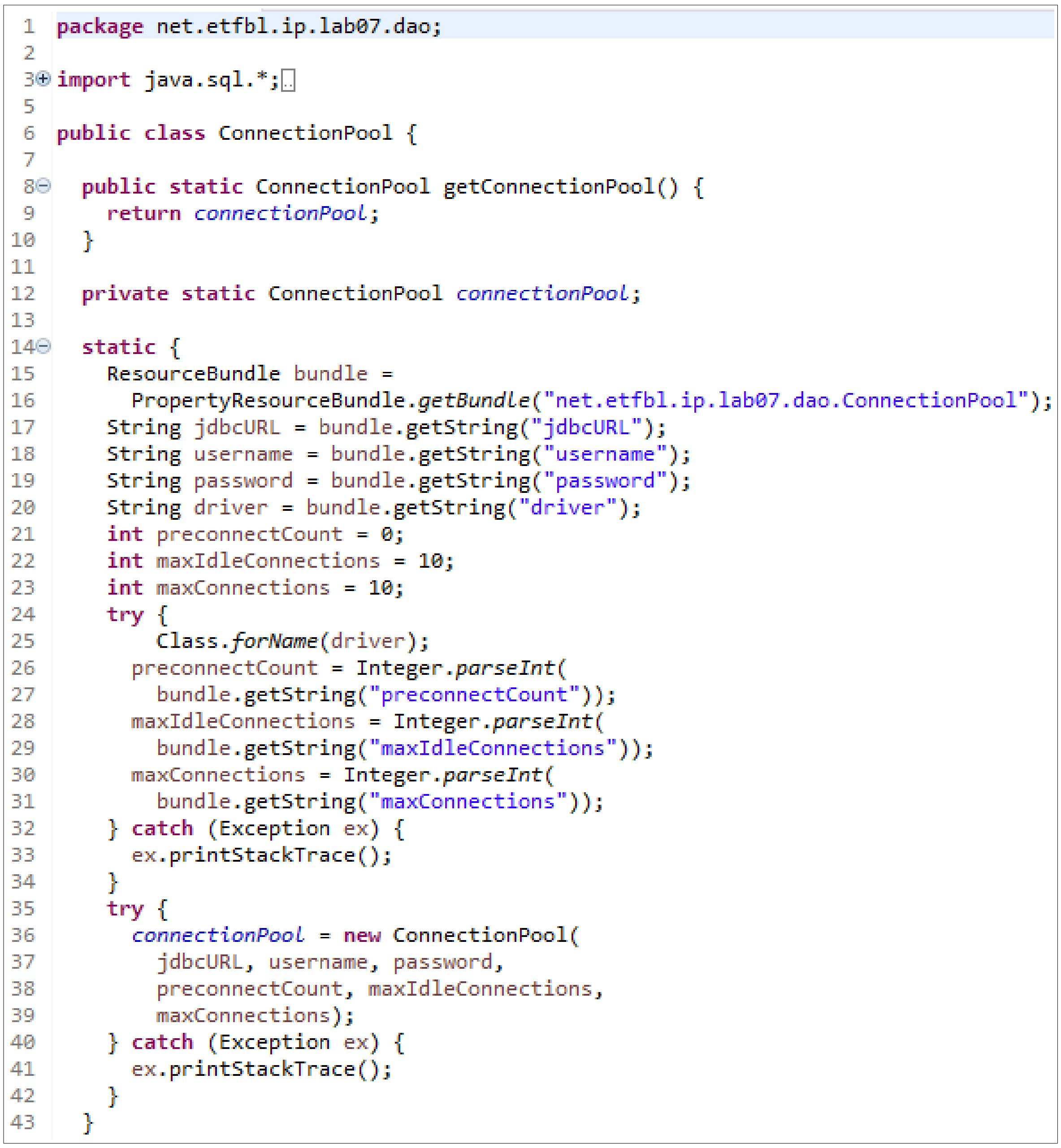
```java
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        User other = (User) obj;
        if (username == null) {
            if (other.username != null)
                return false;
        } else if (!username.equals(other.username))
            return false;
        return true;
    }


}
```

CONNECTION_POOL.PROPERTIES

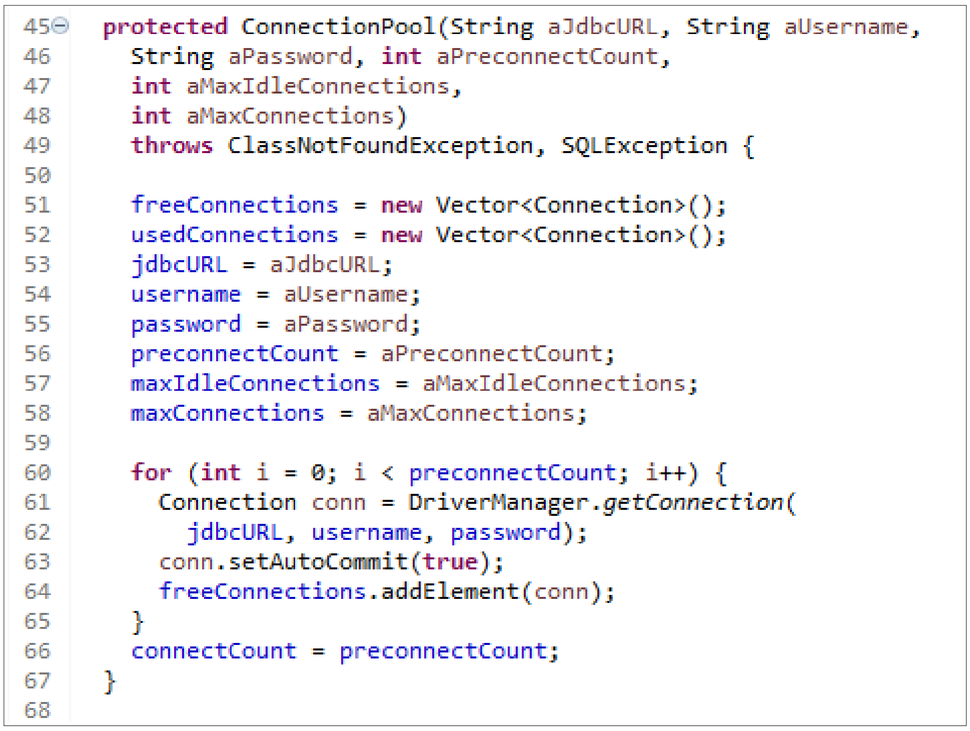


```
jdbcURL=jdbc:mysql://127.0.0.1:3306/etf_lab07?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
username=root
password=root
driver=com.mysql.cj.jdbc.Driver
preconnectCount=2
maxIdleConnections=25
maxConnections=25
```

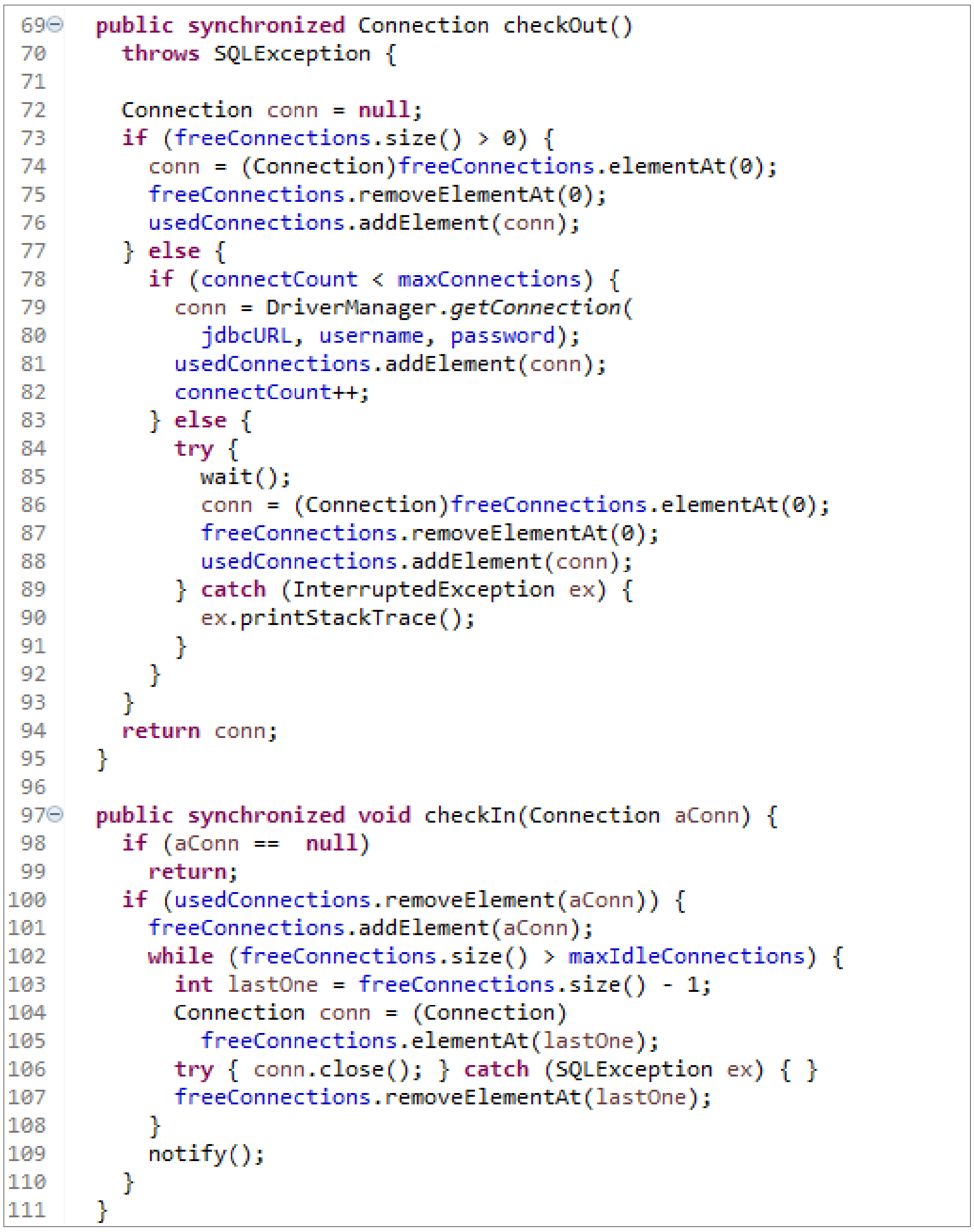
CONNECTION_POOL.JAVA



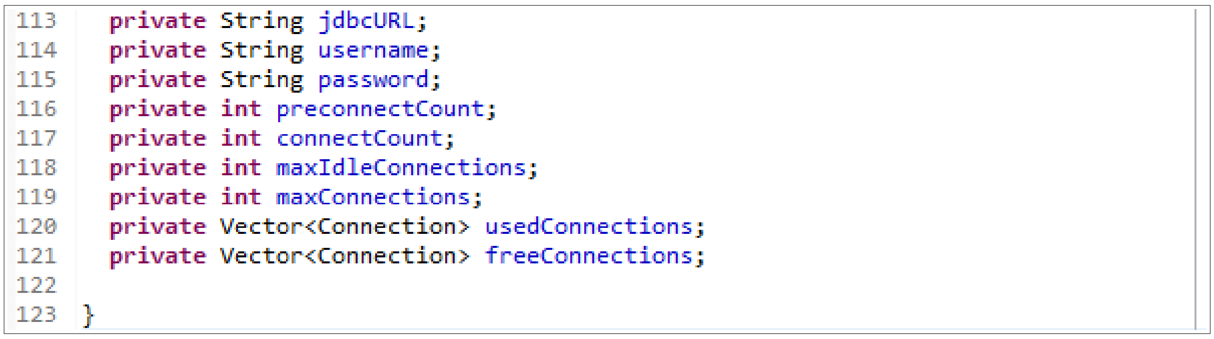
```java
package net.etfbl.ip.lab07.dao;

import java.sql.*;

public class ConnectionPool {

  public static ConnectionPool getConnectionPool() {
    return connectionPool;
  }

  private static ConnectionPool connectionPool;

  static {
    ResourceBundle bundle =
      PropertyResourceBundle.getBundle("net.etfbl.ip.lab07.dao.ConnectionPool");
    String jdbcURL = bundle.getString("jdbcURL");
    String username = bundle.getString("username");
    String password = bundle.getString("password");
    String driver = bundle.getString("driver");
    int preconnectCount = 0;
    int maxIdleConnections = 10;
    int maxConnections = 10;
    try {
        Class.forName(driver);
      preconnectCount = Integer.parseInt(
        bundle.getString("preconnectCount"));
      maxIdleConnections = Integer.parseInt(
        bundle.getString("maxIdleConnections"));
      maxConnections = Integer.parseInt(
        bundle.getString("maxConnections"));
    } catch (Exception ex) {
      ex.printStackTrace();
    }
    try {
      connectionPool = new ConnectionPool(
        jdbcURL, username, password,
        preconnectCount, maxIdleConnections,
        maxConnections);
    } catch (Exception ex) {
      ex.printStackTrace();
    }
  }
```



```java
protected ConnectionPool(String aJdbcURL, String aUsername,
  String aPassword, int aPreconnectCount,
  int aMaxIdleConnections,
  int aMaxConnections)
  throws ClassNotFoundException, SQLException {

  freeConnections = new Vector<Connection>();
  usedConnections = new Vector<Connection>();
  jdbcURL = aJdbcURL;
  username = aUsername;
  password = aPassword;
  preconnectCount = aPreconnectCount;
  maxIdleConnections = aMaxIdleConnections;
  maxConnections = aMaxConnections;

  for (int i = 0; i < preconnectCount; i++) {
    Connection conn = DriverManager.getConnection(
      jdbcURL, username, password);
    conn.setAutoCommit(true);
    freeConnections.addElement(conn);
  }
  connectCount = preconnectCount;
}

```

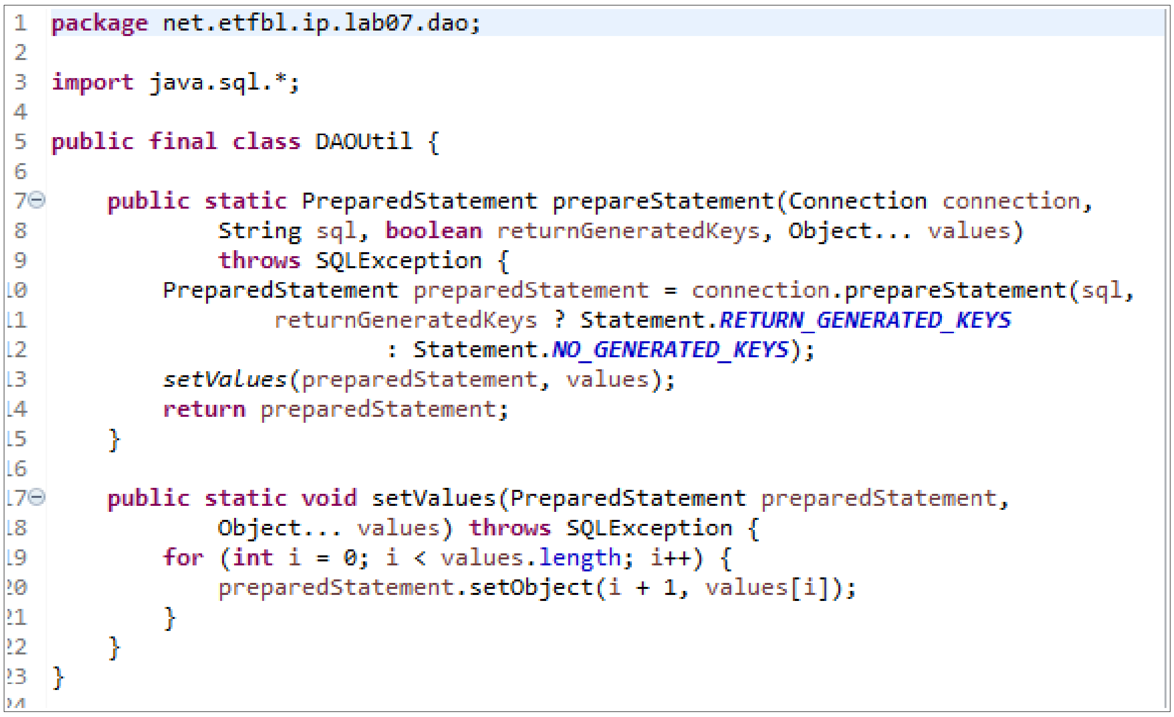


```java
public synchronized Connection checkOut()
  throws SQLException {

  Connection conn = null;
  if (freeConnections.size() > 0) {
    conn = (Connection)freeConnections.elementAt(0);
    freeConnections.removeElementAt(0);
    usedConnections.addElement(conn);
  } else {
    if (connectCount < maxConnections) {
      conn = DriverManager.getConnection(
        jdbcURL, username, password);
      usedConnections.addElement(conn);
      connectCount++;
    } else {
      try {
        wait();
        conn = (Connection)freeConnections.elementAt(0);
        freeConnections.removeElementAt(0);
        usedConnections.addElement(conn);
      } catch (InterruptedException ex) {
        ex.printStackTrace();
      }
    }
  }
  return conn;
}

public synchronized void checkIn(Connection aConn) {
  if (aConn ==  null)
    return;
  if (usedConnections.removeElement(aConn)) {
    freeConnections.addElement(aConn);
    while (freeConnections.size() > maxIdleConnections) {
      int lastOne = freeConnections.size() - 1;
      Connection conn = (Connection)
        freeConnections.elementAt(lastOne);
      try { conn.close(); } catch (SQLException ex) { }
      freeConnections.removeElementAt(lastOne);
    }
    notify();
  }
}
```



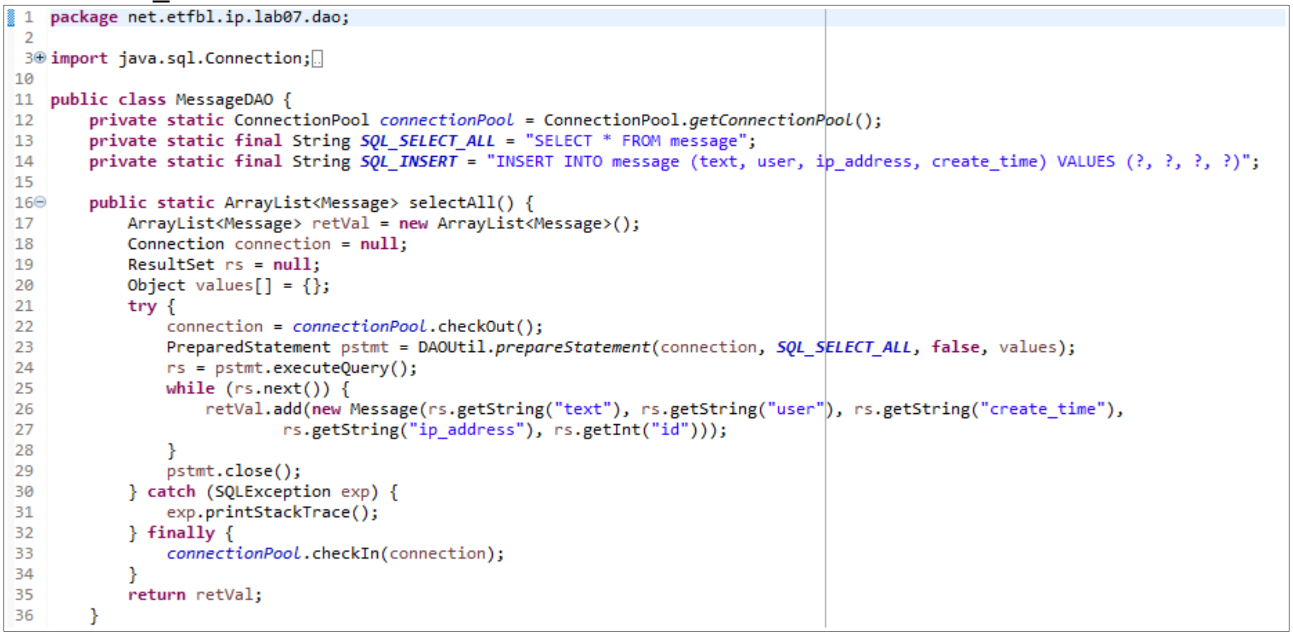
```java
    private String jdbcURL;
    private String username;
    private String password;
    private int preconnectCount;
    private int connectCount;
    private int maxIdleConnections;
    private int maxConnections;
    private Vector<Connection> usedConnections;
    private Vector<Connection> freeConnections;

}
```

DAO_UTIL.JAVA

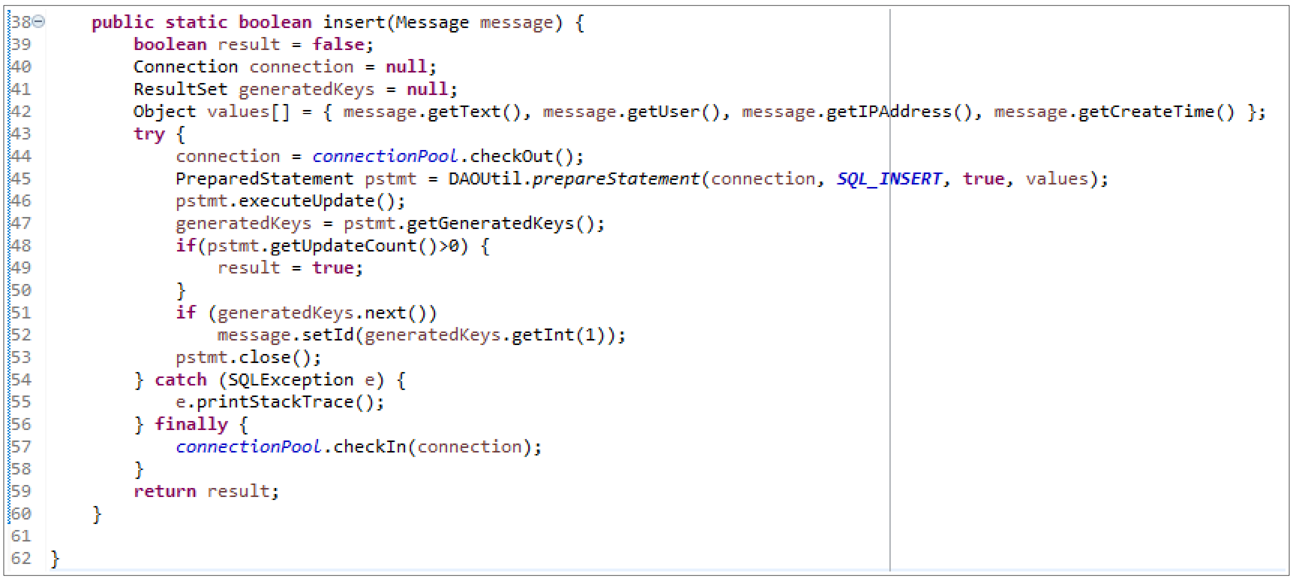


```java
package net.etfbl.ip.lab07.dao;

import java.sql.*;

public final class DAOUtil {

    public static PreparedStatement prepareStatement(Connection connection,
            String sql, boolean returnGeneratedKeys, Object... values)
            throws SQLException {
        PreparedStatement preparedStatement = connection.prepareStatement(sql,
                returnGeneratedKeys ? Statement.RETURN_GENERATED_KEYS
                        : Statement.NO_GENERATED_KEYS);
        setValues(preparedStatement, values);
        return preparedStatement;
    }

    public static void setValues(PreparedStatement preparedStatement,
            Object... values) throws SQLException {
        for (int i = 0; i < values.length; i++) {
            preparedStatement.setObject(i + 1, values[i]);
        }
    }
}
```

MESSAGE_DAO.JAVA

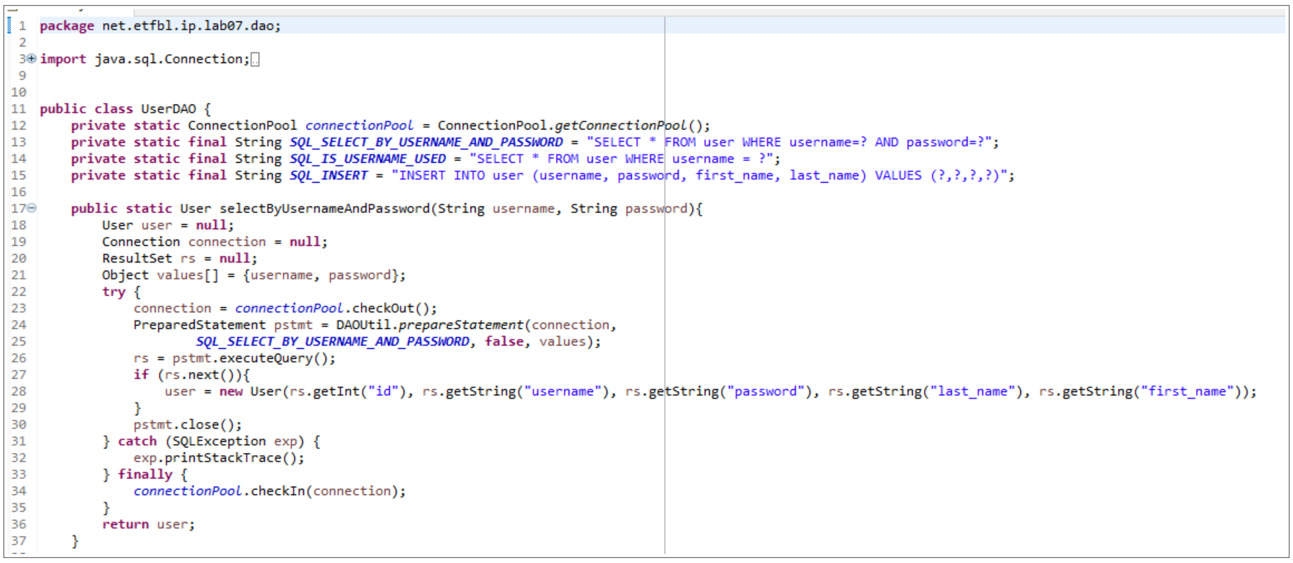


```java
package net.etfbl.ip.lab07.dao;

import java.sql.Connection;

public class MessageDAO {
    private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
    private static final String SQL_SELECT_ALL = "SELECT * FROM message";
    private static final String SQL_INSERT = "INSERT INTO message (text, user, ip_address, create_time) VALUES (?, ?, ?, ?)";

    public static ArrayList<Message> selectAll() {
        ArrayList<Message> retVal = new ArrayList<Message>();
        Connection connection = null;
        ResultSet rs = null;
        Object values[] = {};
        try {
            connection = connectionPool.checkOut();
            PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_SELECT_ALL, false, values);
            rs = pstmt.executeQuery();
            while (rs.next()) {
                retVal.add(new Message(rs.getString("text"), rs.getString("user"), rs.getString("create_time"),
                        rs.getString("ip_address"), rs.getInt("id")));
            }
            pstmt.close();
        } catch (SQLException exp) {
            exp.printStackTrace();
        } finally {
            connectionPool.checkIn(connection);
        }
        return retVal;
    }
```

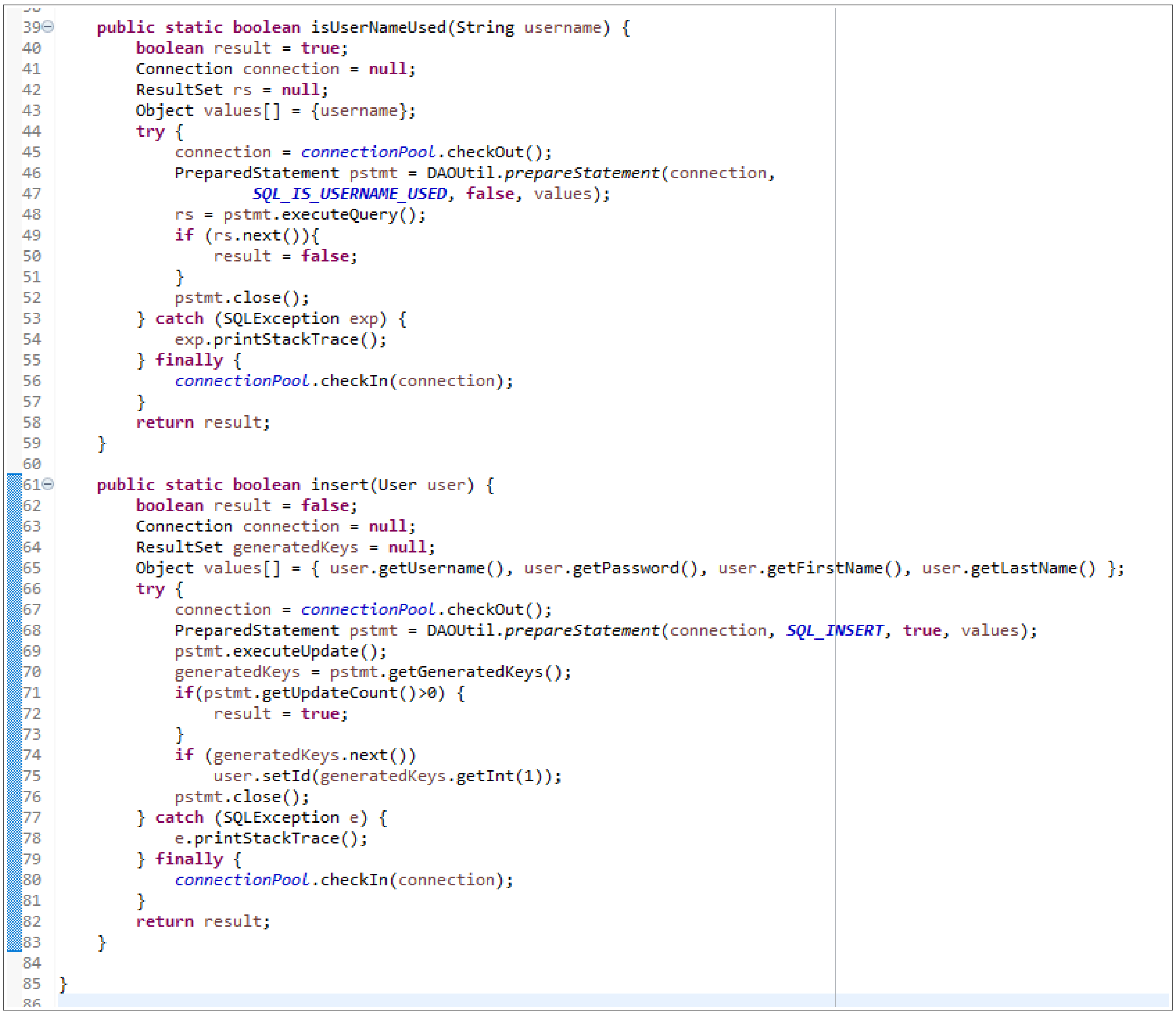


```java
    public static boolean insert(Message message) {
        boolean result = false;
        Connection connection = null;
        ResultSet generatedKeys = null;
        Object values[] = { message.getText(), message.getUser(), message.getIPAddress(), message.getCreateTime() };
        try {
            connection = connectionPool.checkOut();
            PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_INSERT, true, values);
            pstmt.executeUpdate();
            generatedKeys = pstmt.getGeneratedKeys();
            if(pstmt.getUpdateCount()>0) {
                result = true;
            }
            if (generatedKeys.next())
                message.setId(generatedKeys.getInt(1));
            pstmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            connectionPool.checkIn(connection);
        }
        return result;
    }

}
```

USER_DAO.JAVA

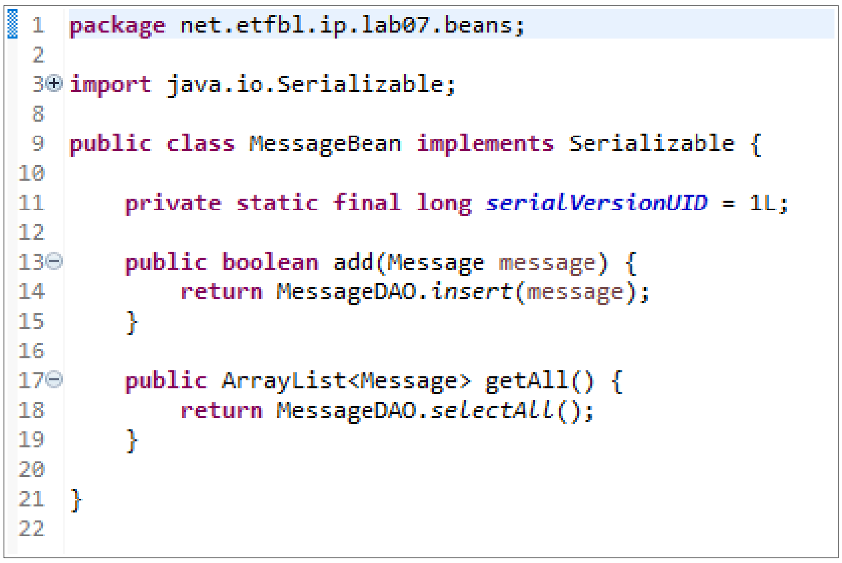


```java
package net.etfbl.ip.lab07.dao;

import java.sql.Connection;


public class UserDAO {
    private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
    private static final String SQL_SELECT_BY_USERNAME_AND_PASSWORD = "SELECT * FROM user WHERE username=? AND password=?";
    private static final String SQL_IS_USERNAME_USED = "SELECT * FROM user WHERE username = ?";
    private static final String SQL_INSERT = "INSERT INTO user (username, password, first_name, last_name) VALUES (?,?,?,?)";

    public static User selectByUsernameAndPassword(String username, String password){
        User user = null;
        Connection connection = null;
        ResultSet rs = null;
        Object values[] = {username, password};
        try {
            connection = connectionPool.checkOut();
            PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
                    SQL_SELECT_BY_USERNAME_AND_PASSWORD, false, values);
            rs = pstmt.executeQuery();
            if (rs.next()){
                user = new User(rs.getInt("id"), rs.getString("username"), rs.getString("password"), rs.getString("last_name"), rs.getString("first_name"));
            }
            pstmt.close();
        } catch (SQLException exp) {
            exp.printStackTrace();
        } finally {
            connectionPool.checkIn(connection);
        }
        return user;
    }
```



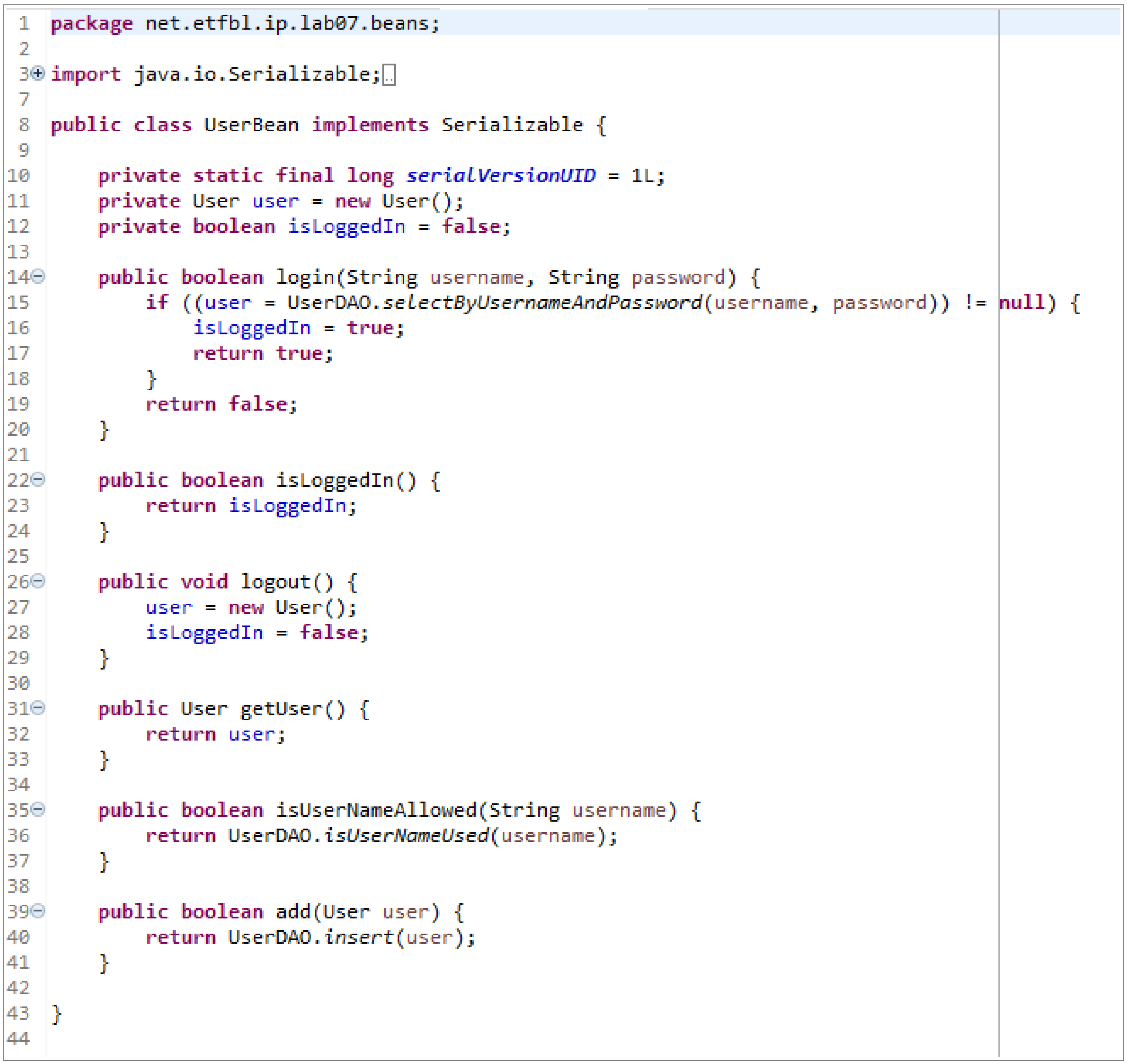
```java
    public static boolean isUserNameUsed(String username) {
        boolean result = true;
        Connection connection = null;
        ResultSet rs = null;
        Object values[] = {username};
        try {
            connection = connectionPool.checkOut();
            PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
                    SQL_IS_USERNAME_USED, false, values);
            rs = pstmt.executeQuery();
            if (rs.next()){
                result = false;
            }
            pstmt.close();
        } catch (SQLException exp) {
            exp.printStackTrace();
        } finally {
            connectionPool.checkIn(connection);
        }
        return result;
    }

    public static boolean insert(User user) {
        boolean result = false;
        Connection connection = null;
        ResultSet generatedKeys = null;
        Object values[] = { user.getUsername(), user.getPassword(), user.getFirstName(), user.getLastName() };
        try {
            connection = connectionPool.checkOut();
            PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_INSERT, true, values);
            pstmt.executeUpdate();
            generatedKeys = pstmt.getGeneratedKeys();
            if(pstmt.getUpdateCount()>0) {
                result = true;
            }
            if (generatedKeys.next())
                user.setId(generatedKeys.getInt(1));
            pstmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            connectionPool.checkIn(connection);
        }
        return result;
    }

}
```

MESSAGE_BEAN.JAVA



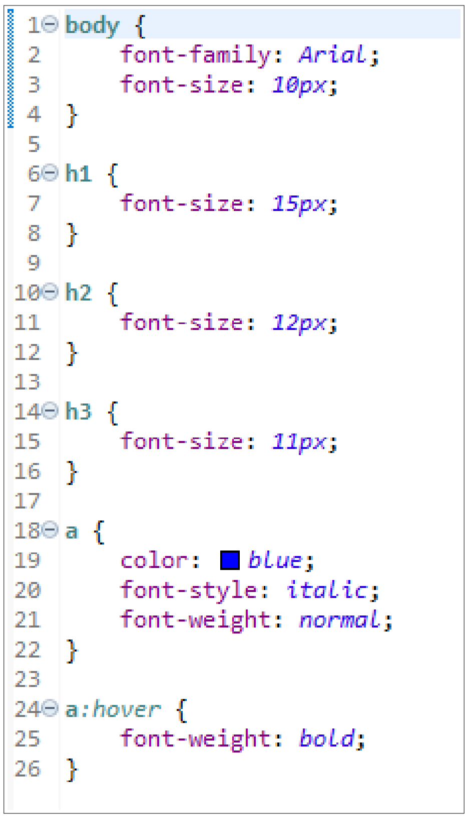
```java
package net.etfbl.ip.lab07.beans;

import java.io.Serializable;

public class MessageBean implements Serializable {

    private static final long serialVersionUID = 1L;

    public boolean add(Message message) {
        return MessageDAO.insert(message);
    }

    public ArrayList<Message> getAll() {
        return MessageDAO.selectAll();
    }

}

```

USER_BEAN.JAVA



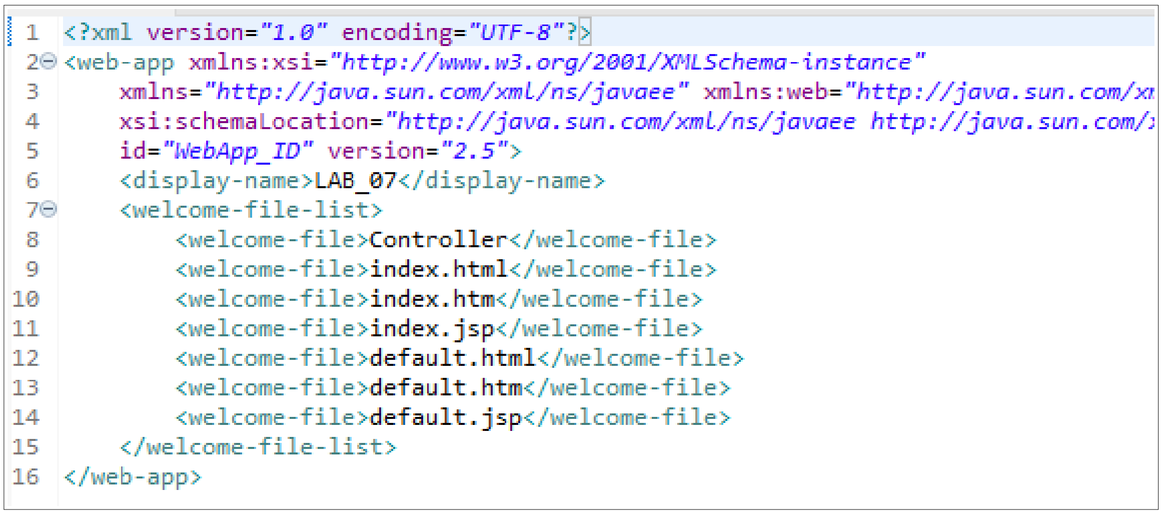
```java
package net.etfbl.ip.lab07.beans;

import java.io.Serializable;

public class UserBean implements Serializable {

    private static final long serialVersionUID = 1L;
    private User user = new User();
    private boolean isLoggedIn = false;

    public boolean login(String username, String password) {
        if ((user = UserDAO.selectByUsernameAndPassword(username, password)) != null) {
            isLoggedIn = true;
            return true;
        }
        return false;
    }

    public boolean isLoggedIn() {
        return isLoggedIn;
    }

    public void logout() {
        user = new User();
        isLoggedIn = false;
    }

    public User getUser() {
        return user;
    }

    public boolean isUserNameAllowed(String username) {
        return UserDAO.isUserNameUsed(username);
    }

    public boolean add(User user) {
        return UserDAO.insert(user);
    }

}
```

STYLES.CSS



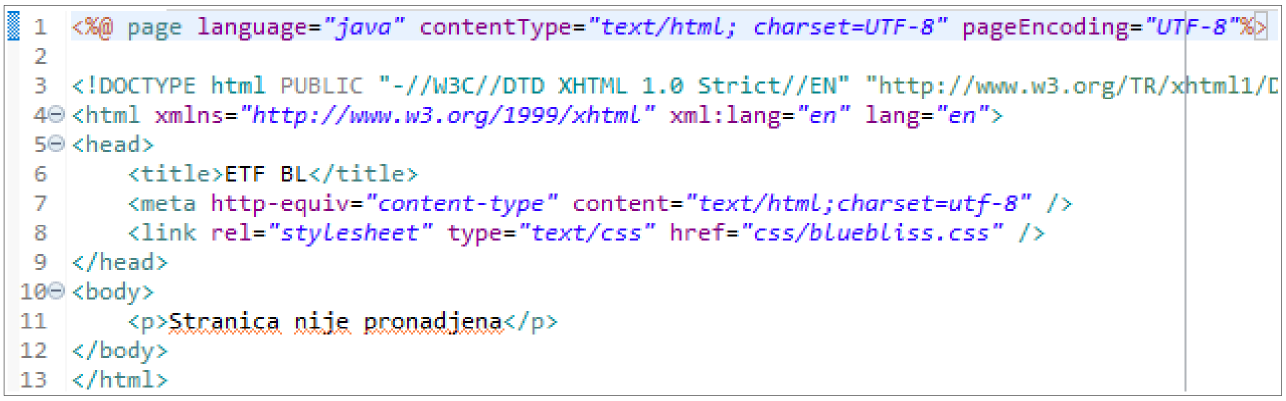
```css
body {
    font-family: Arial;
    font-size: 10px;
}

h1 {
    font-size: 15px;
}

h2 {
    font-size: 12px;
}

h3 {
    font-size: 11px;
}

a {
    color: ■ blue;
    font-style: italic;
    font-weight: normal;
}

a:hover {
    font-weight: bold;
}
```

WEB.XML



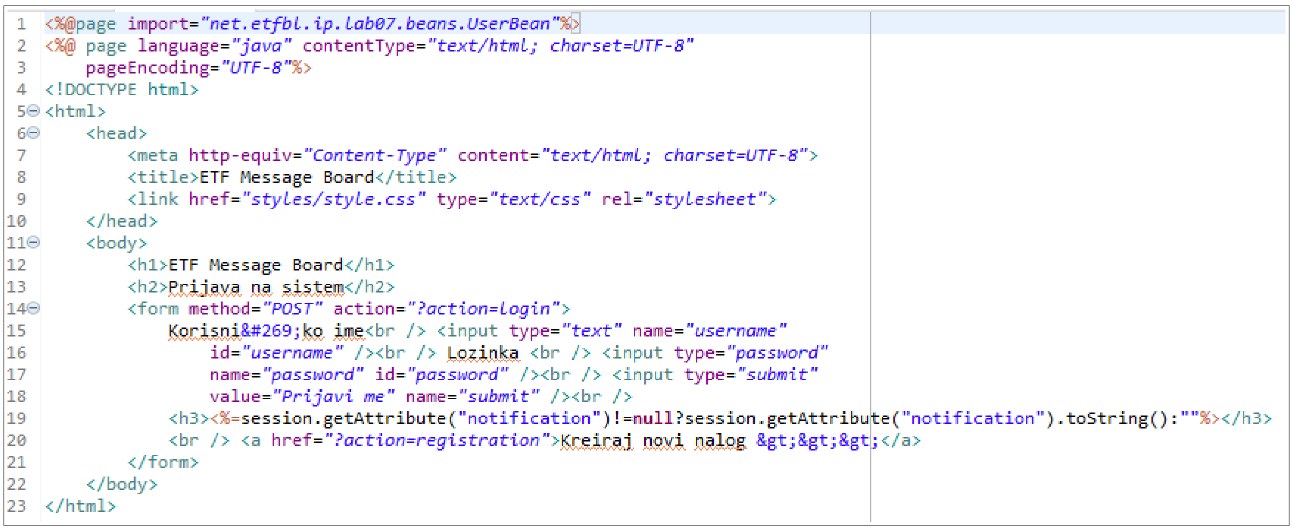
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xı
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/:
    id="WebApp_ID" version="2.5">
    <display-name>LAB_07</display-name>
    <welcome-file-list>
        <welcome-file>Controller</welcome-file>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

404.jsp



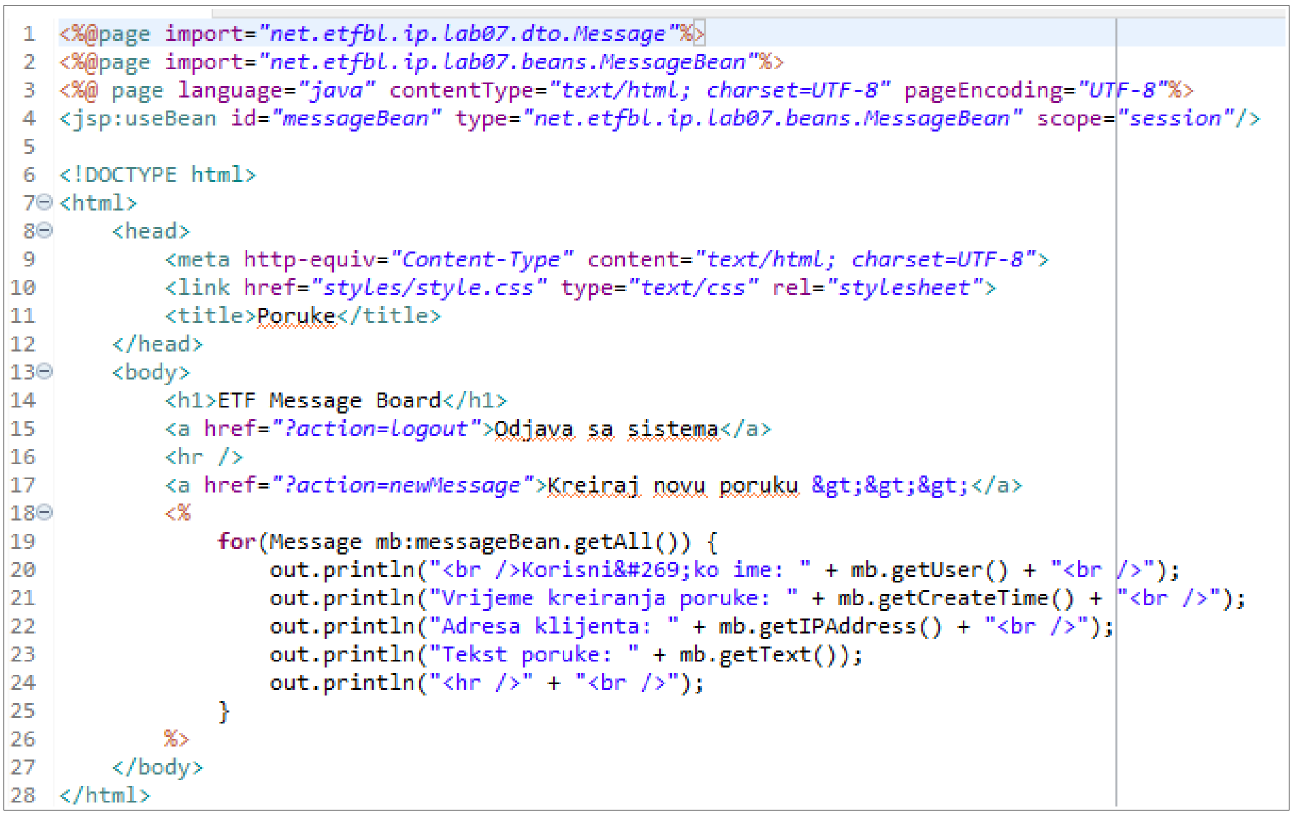
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/D
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>ETF BL</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="css/bluebliss.css" />
</head>
<body>
    <p>Stranica nije pronadjena</p>
</body>
</html>
```

LOGIN.JSP



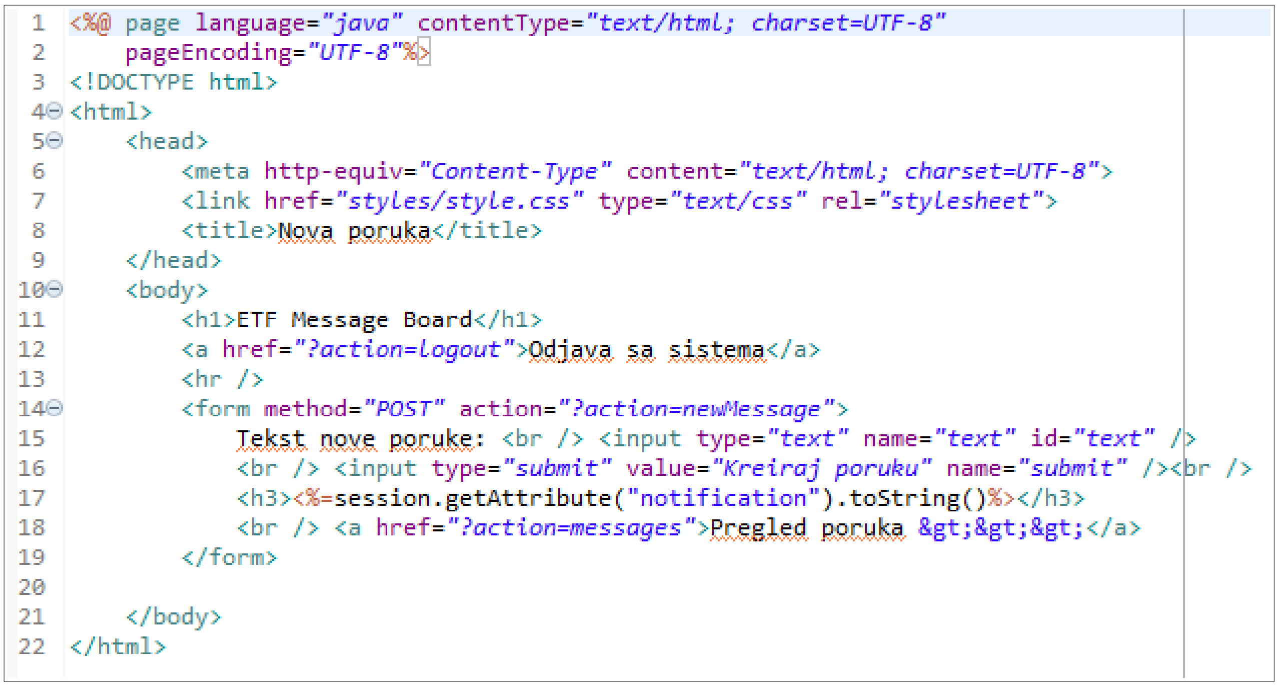
```
<%@page import="net.etfbl.ip.lab07.beans.UserBean"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>ETF Message Board</title>
        <link href="styles/style.css" type="text/css" rel="stylesheet">
    </head>
    <body>
        <h1>ETF Message Board</h1>
        <h2>Prijava na sistem</h2>
        <form method="POST" action="?action=login">
            Korisni&#269;ko ime<br /> <input type="text" name="username"
                id="username" /><br /> Lozinka <br /> <input type="password"
                name="password" id="password" /><br /> <input type="submit"
                value="Prijavi me" name="submit" /><br />
            <h3><%=session.getAttribute("notification")!=null?session.getAttribute("notification").toString():""%></h3>
            <br /> <a href="?action=registration">Kreiraj novi nalog &gt;&gt;&gt;</a>
        </form>
    </body>
</html>
```

MESSAGES.JSP



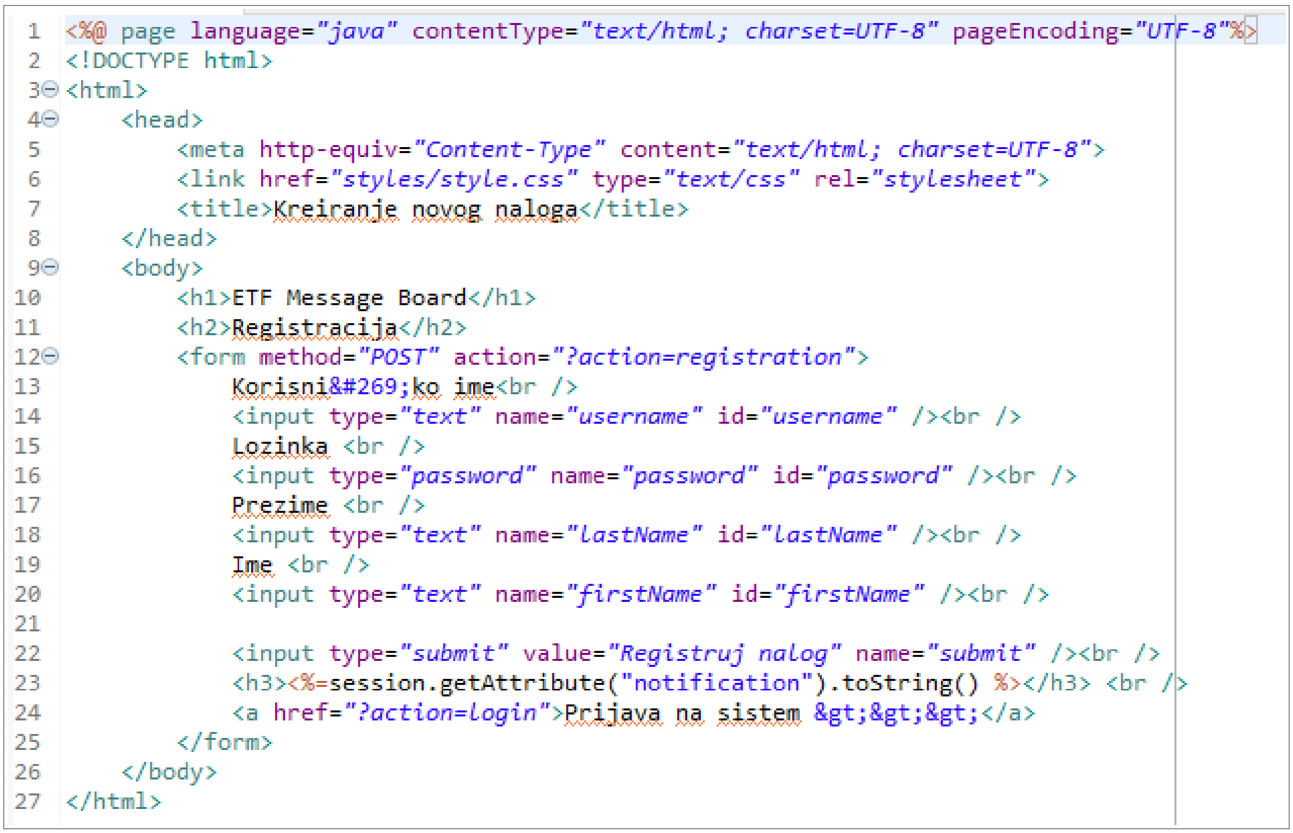
```
<%@page import="net.etfbl.ip.lab07.dto.Message"%>
<%@page import="net.etfbl.ip.lab07.beans.MessageBean"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<jsp:useBean id="messageBean" type="net.etfbl.ip.lab07.beans.MessageBean" scope="session"/>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link href="styles/style.css" type="text/css" rel="stylesheet">
        <title>Poruke</title>
    </head>
    <body>
        <h1>ETF Message Board</h1>
        <a href="?action=logout">Odjava sa sistema</a>
        <hr />
        <a href="?action=newMessage">Kreiraj novu poruku &gt;&gt;&gt;</a>
        <%
            for(Message mb:messageBean.getAll()) {
                out.println("<br />Korisni&#269;ko ime: " + mb.getUser() + "<br />");
                out.println("Vrijeme kreiranja poruke: " + mb.getCreateTime() + "<br />");
                out.println("Adresa klijenta: " + mb.getIPAddress() + "<br />");
                out.println("Tekst poruke: " + mb.getText());
                out.println("<hr />" + "<br />");
            }
        %>
    </body>
</html>
```

NEW_MESSAGE.JSP



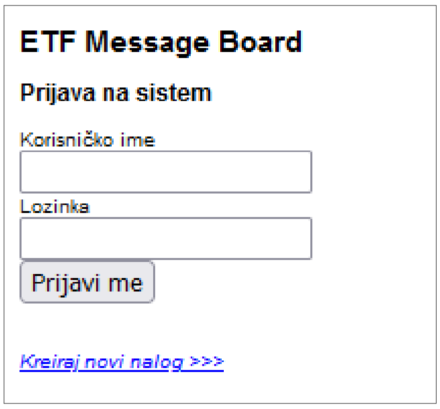
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link href="styles/style.css" type="text/css" rel="stylesheet">
        <title>Nova poruka</title>
    </head>
    <body>
        <h1>ETF Message Board</h1>
        <a href="?action=logout">Odjava sa sistema</a>
        <hr />
        <form method="POST" action="?action=newMessage">
            Tekst nove poruke: <br /> <input type="text" name="text" id="text" />
            <br /> <input type="submit" value="Kreiraj poruku" name="submit" /><br />
            <h3><%=session.getAttribute("notification").toString()%></h3>
            <br /> <a href="?action=messages">Pregled poruka &gt;&gt;&gt;</a>
        </form>

    </body>
</html>
```

REGISTRATOION.JSP



```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link href="styles/style.css" type="text/css" rel="stylesheet">
        <title>Kreiranje novog naloga</title>
    </head>
    <body>
        <h1>ETF Message Board</h1>
        <h2>Registracija</h2>
        <form method="POST" action="?action=registration">
            Korisni&#269;ko ime<br />
            <input type="text" name="username" id="username" /><br />
            Lozinka <br />
            <input type="password" name="password" id="password" /><br />
            Prezime <br />
            <input type="text" name="lastName" id="lastName" /><br />
            Ime <br />
            <input type="text" name="firstName" id="firstName" /><br />

            <input type="submit" value="Registruj nalog" name="submit" /><br />
            <h3><%=session.getAttribute("notification").toString() %></h3> <br />
            <a href="?action=login">Prijava na sistem &gt;&gt;&gt;</a>
        </form>
    </body>
</html>
```

LOGIN – DESIGN


ETF Message Board

Prijava na sistem

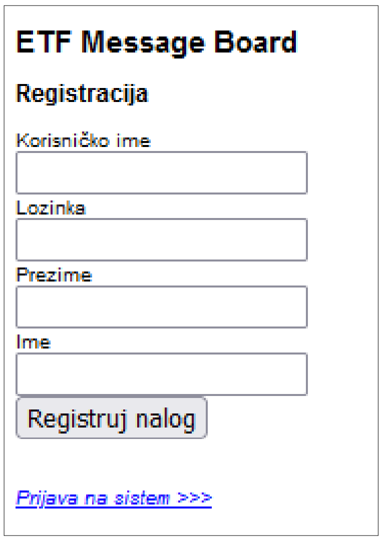
Korisničko ime

Lozinka

Prijavi me

Kreiraj novi nalog >>>

REGISTER – DESIGN


ETF Message Board

Registracija

Korisničko ime

Lozinka

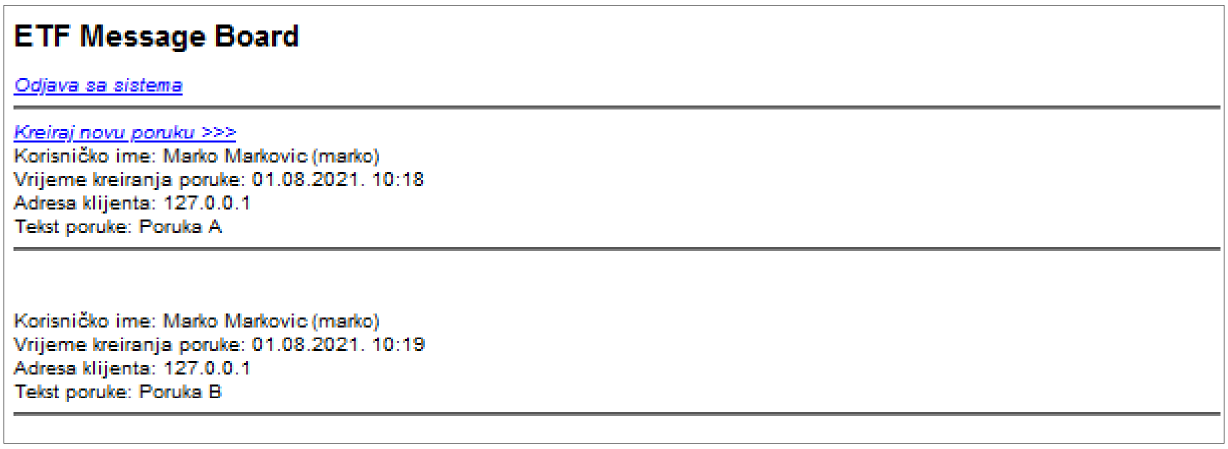
Prezime

Ime

Registruj nalog

Prijava na sistem >>>

MESSAGES – LIST

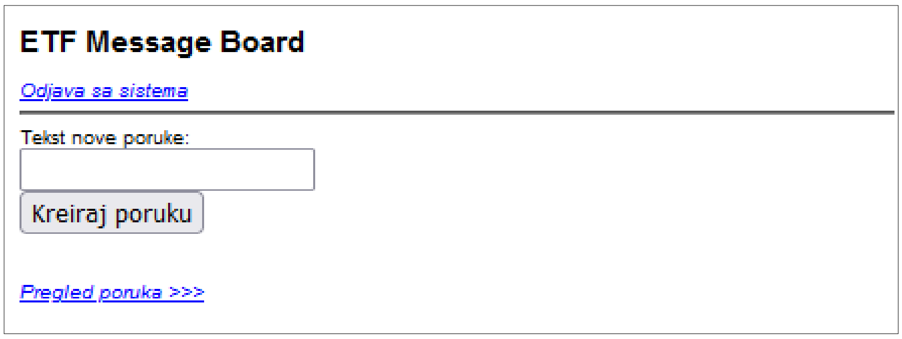

ETF Message Board

Odjava sa sistema

Kreiraj novu poruku >>>
Korisničko ime: Marko Markovic (marko)
Vrijeme kreiranja poruke: 01.08.2021. 10:18
Adresa klijenta: 127.0.0.1
Tekst poruke: Poruka A

Korisničko ime: Marko Markovic (marko)
Vrijeme kreiranja poruke: 01.08.2021. 10:19
Adresa klijenta: 127.0.0.1
Tekst poruke: Poruka B

NEW – MESSAGE – DESIGN


ETF Message Board

Odjava sa sistema

Tekst nove poruke:

Pregled poruka >>>