

XI`AN TECHNOLOGICAL UNIVERSITY

实验报告

实验课程名称 组件技术

专 业： 软件工程

班 级： 17060721

姓 名： 李正阳

学 号： 17060208112

实验学时： 2

指导教师： 白小军

成 绩：

2019 年 11 月 11 日

西安工业大学实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 专业 | 软件工程 | 班级 | 17060721 | 姓名 | 李正阳 | 学号 | 17060208112 |
| 实验课程 | 组件技术 | 指导教师 | 白小军 | 实验日期 | 2019/11/11 | 同实验者 | 无 |
| 实验项目 | JDBC技术 | | | | | | |
| 实验设备及器材 | PC机 | | | | | | |

1. 实验目的

1）熟悉JDBC的体系结构和常用编程接口；

2）能熟练使用Connection对象连接到常用数据库；

3）能熟练使用Statement、PrepareStatement及CallableStatement对象访问数据库，执行SQL语句并接收返回结果；

4）能熟练使用ResultSet对象处理返回的数据；

5）理解数据库连接池的设计思想，理解数据源的概念，能熟练配置和使用数据源以提高数据访问效率；

6）理解DAO设计模式的基本思想，能设计自己的VO对象和DAO对象以分离数据访问逻辑。

二、实验原理

1、JDBC的体系结构和和常用编程接口；

Java通过JDBC连接数据库操作共同协议，然后来操作不同公司开发的数据库。

常用编程接口：

Driver 驱动器

Connection 与数据库连接

ResultSet 类似一个临时表

DriverManage 用来管理数据库中的所有驱动程序。

Statement 在已建立连接的基础上向数据库发送SQL语句。

1. 使用DriverManager方式连接并访问数据库的基本方法；

Class.forName(String driver)加载驱动

DriverManager.getConnection(String url,String username,String passwroed)代表数据库的连接，传输数据库的url、数据库的账号和密码。

Statement接口可以执行SQL语句采用的字符串拼接

PrepardStatement接口可以通过设置参数和设置值来进行连接SQL语句

Resultset方法用来获取数据库查找后返回来的信息。

Close方法来关数据库连接

1. 数据源的概念，以及使用数据源方式连接并访问数据库的方法；

数据源就是连接数据库的一条路径，包括数据库连接池和连接池管理两部分。

数据源连接方式：

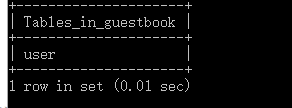
将数据库的信息写入properties文件当中，通过IO流读取文件，使用Properties对象中的load方法进行加载，通过获取properties文件来构建数据源，然后通过数据源来获取数据库连接对象Connection，同时数据库连接的资源释放也由数据库连接池来自动管理。

4、DAO设计模式的基本思想，VO类及DAO类的基本设计。

1. 实验步骤、数据记录及处理
2. 安装和启动MySql服务，创建GuestBook数据库及User表，表中含ID、用户名、密码、邮箱等字段；

创建数据库和设计的表如下：

**CREATE** DATABASE GuestBook;  
**CREATE TABLE User**(  
 **id INT** auto\_increment **PRIMARY KEY NOT NULL**,  
 **name VARCHAR** (11) **NOT NULL** ,  
 **password VARCHAR** (16) **NOT NULL** ,  
 **mail VARCHAR** (20) **NOT NULL**);

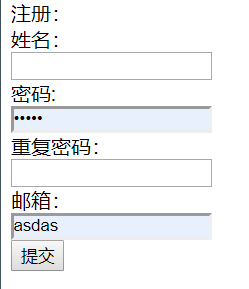


1. 创建用户注册页面及注册处理Servlet，能将注册用户信息插入User表；

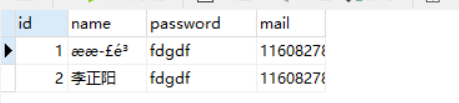
处理注册页面的Servlet：

**package** com.darksouls;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
**import** java.sql.\*;  
  
**public class** RegisterServlet **extends** HttpServlet {  
 @Override  
 **protected void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 String name = **"root"**;  
 String password = **"root"**;  
 String url = **"jdbc:mysql://localhost:3306/guestbook?useSSL=false&serverTimezone=UTC"**;  
 String driverClassName = **"com.mysql.cj.jdbc.Driver"**;  
 Connection conn = **null**;  
 PreparedStatement prst = **null**;  
 String username = req.getParameter(**"name"**);  
 String userpassword = req.getParameter(**"password"**);  
 String email = req.getParameter(**"mail"**);  
 **try** {  
 Class.*forName*(driverClassName);  
 conn= DriverManager.*getConnection*(url,name,password);  
 String sql = **"INSERT INTO user(name,password,mail) VALUES ( ?,?,?)"**;  
 prst = conn.prepareStatement(sql);  
 prst.setString(1,username);  
 prst.setString(2,userpassword);  
 prst.setString(3,email);  
 prst.execute();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **try** {  
 **if**(prst != **null**){  
 prst.close();  
 }  
 **if**(conn !=**null**){  
 conn.close();  
 }  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

注册界面如下：



点击完注册之后，数据库中出现如下信息：



1. 开发用户登录页面及登录验证Servlet，从User表中检索用户信息进行验证，验证通过则转入成功页面，否则提示重新登录；

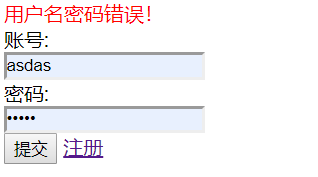
修改登陆的Servlet代码如下：

通过读取数据库中存储的密码和用户发送来的密码，进行验证。

**package** com.darksouls;  
  
**import** com.alibaba.druid.util.StringUtils;  
**import** sun.rmi.runtime.Log;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.http.Cookie;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.util.HashMap;  
  
*/\*\*  
 \** ***@author*** *李正阳 17060208112  
 \*/***public class** LoginServerlet **extends** HttpServlet {  
 **static** HashMap<String,Integer> *LoginPool* = **new** HashMap<String, Integer>();  
 @Override  
 **public void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
  
 req.setCharacterEncoding(**"utf-8"**);  
 resp.setCharacterEncoding(**"utf-8"**);  
 String userName = req.getParameter(**"name"**);  
 String userPassword = req.getParameter(**"password"**);  
 */\*\*  
 \* 记录数据库连接信息  
 \*/* String name = **"root"**;  
 String password = **"root"**;  
 String url = **"jdbc:mysql://localhost:3306/guestbook?useSSL=false&serverTimezone=UTC"**;  
 String driverClassName = **"com.mysql.cj.jdbc.Driver"**;  
 Connection conn = **null**;  
 PreparedStatement prst = **null**;  
 ResultSet rs = **null**;  
  
 **try** {  
 Class.*forName*(driverClassName);  
 conn = DriverManager.*getConnection*(url,name,password);  
 String sql = **"SELECT password FROM user WHERE name = ?"**;  
 prst = conn.prepareStatement(sql);  
 prst.setString(1,userName);  
 rs = prst.executeQuery();  
 String sqlpassword = **null**;  
 **if**(rs.next()){  
 sqlpassword = rs.getString(**"password"**);  
 }  
  
 */\*\*  
 \* 验证登陆  
 \*/* **if**(!StringUtils.*isEmpty*(userName)){  
 **if**(StringUtils.*equals*(sqlpassword,userPassword)){  
 */\*\*  
 \* 添加session，以及访问次数  
 \*/* **if**(*LoginPool*.containsKey(userName)){  
 *LoginPool*.put(userName,*LoginPool*.get(userName) + 1);  
 }**else** {  
 *LoginPool*.put(userName,1);  
 }  
 req.getSession().setAttribute(**"USER\_NAME"**,userName);  
 req.getSession().setAttribute(**"NUM"**,*LoginPool*.get(userName));  
  
 req.getRequestDispatcher(**"WEB-INF/view/success.jsp"**).forward(req,resp);  
 }**else** {  
 req.setAttribute(**"ERROR\_INFO"**,**"<span style='color:red;'>用户名密码错误！</span>"**);  
 req.getRequestDispatcher(**"index.jsp"**).forward(req,resp);  
 }  
 }**else** {  
 req.setAttribute(**"ERROR\_INFO"**,**"<span style='color:red;'>用户名密码错误！</span>"**);  
 req.getRequestDispatcher(**"index.jsp"**).forward(req,resp);  
 }  
  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **try** {  
 **if**(rs != **null**){  
 rs.close();  
 }  
 **if**(prst != **null**){  
 prst.close();  
 }  
 **if**(conn !=**null**){  
 conn.close();  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 req.getRequestDispatcher(**"WEB-INF/view/error.jsp"**).forward(req,resp);  
 }  
}

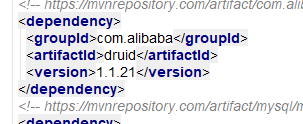


密码错误界面如下：



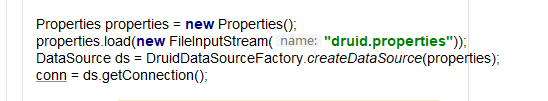
1. 使用DataSource方式重新实现用户注册和登录验证的Servlet，写出数据源配置方法，及在Servlet中检索和使用数据源的方法；

数据源有DBCP和C3P0，但是目前最流行还是阿里巴巴的druid数据源，所以我使用druid数据源来完成本步骤。



在MAVEN中引入druid的jar包。

然后将数据库连接的信息写入druid.properties文件中，然后将LoginServlet文件中的数据库连接换成数据源的连接方式，代码如下：



从druid.properties中获取Connection连接。

然后使用方法还和之前是一样的。

使用登陆界面，成功登陆，如下图：

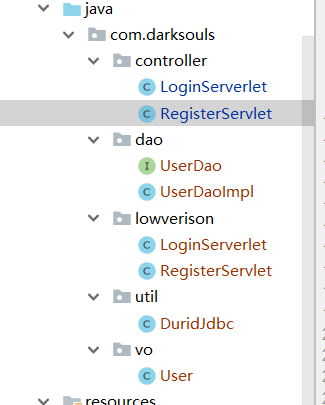


注册Servelet和登陆的Servelet是修改方法是一样的。

注册和登陆的Servelet的源码放在github上了。

1. 使用DAO模式设计数据访问层，写出VO类及DAO类的核心代码，以及在Servlet中调用数据访问层的方法。

创建结构如下：



**Vo层保存的是用户实体**

代码如下：

**package** com.darksouls.vo;  
  
**public class** User {  
 **private** String **name**;  
 **private** String **password**;  
 **private** String **mail**;  
 **public** User() {  
 }  
  
 **public** User(String name, String password, String mail) {  
 **this**.**name** = name;  
 **this**.**password** = password;  
 **this**.**mail** = mail;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getPassword() {  
 **return password**;  
 }  
  
 **public void** setPassword(String password) {  
 **this**.**password** = password;  
 }  
  
 **public** String getMail() {  
 **return mail**;  
 }  
  
 **public void** setMail(String mail) {  
 **this**.**mail** = mail;  
 }  
}

Dao层将对数据的连接和操作都封装起来

UserDao.java接口文件

**package** com.darksouls.dao;  
  
**import** com.darksouls.vo.User;  
  
**public interface** UserDao {  
  
 String SelectUserPassword(String name);  
  
 **void** InsertUser(User user);  
   
}

**实现UseDao接口的UserDaoImpl.java**

**package** com.darksouls.dao;  
  
**import** com.darksouls.util.DuridJdbc;  
**import** com.darksouls.vo.User;  
  
**import** java.sql.Connection;  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
  
**public class** UserDaoImpl **implements** UserDao {  
 */\*\*  
 \* 查询userName用户密码  
 \** ***@param userName*** *被查询用户的用户名  
 \** ***@return*** *该用户的密码  
 \*/* **public** String SelectUserPassword(String userName) {  
  
 Connection conn = DuridJdbc.*getConnection*();  
 String sql = **"SELECT password FROM user WHERE name = ?"**;  
 PreparedStatement prst = **null**;  
 ResultSet rs = **null**;  
 **try** {  
 prst = conn.prepareStatement(sql);  
 prst.setString(1,userName);  
 rs = prst.executeQuery();  
 String sqlpassword = **null**;  
 **if**(rs.next()){  
 sqlpassword = rs.getString(**"password"**);  
 }  
 **return** sqlpassword;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }**finally** {  
 **try** {  
 **if**(rs != **null**) {  
 rs.close();  
 }  
 **if**(prst != **null**){  
 prst.close();  
 }  
 **if**(conn !=**null**){  
 conn.close();  
 }  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 **return null**;  
 }  
  
 */\*\*  
 \* '注册时插入用户信息  
 \** ***@param user*** *用户的实体类  
 \*/* **public void** InsertUser(User user) {  
 Connection conn = DuridJdbc.*getConnection*();  
 PreparedStatement prst = **null**;  
 ResultSet rs = **null**;  
 **try** {  
 String sql = **"INSERT INTO user(name,password,mail) VALUES ( ?,?,?)"**;  
 prst = conn.prepareStatement(sql);  
 prst.setString(1,user.getName());  
 prst.setString(2,user.getPassword());  
 prst.setString(3,user.getMail());  
 prst.execute();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
}

6、假如在数据库中验证用户名和密码的工作封装在存储过程中，该存储过程接收用户名和密码2个参数，返回一个整数（代表满足条件的记录条数），试设计该存储过程，并在登录验证Servlet中调用该存储过程。（选作）

在UserDao里面添加**int** SelectUser(String name,String password);方法：

在UserDaoImpl中实现这个方法：

**public int** SelectUser(String name, String password) {  
 PreparedStatement prst ;  
 ResultSet rs ;  
 **try** {  
 String sql = **"SELECT** *COUNT***(***\****) AS TOTALUSER FROM user WHERE name = ? AND password = ?"**;  
 prst = **conn**.prepareStatement(sql);  
 prst.setString(1,name);  
 prst.setString(2,password);  
 rs = prst.executeQuery();  
 **if**(rs.next()) {  
 n = rs.getInt(**"TOTALUSER"**);  
 }  
 **return** n;  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
 **return** -1;  
}

然后修改LoginServlet中的部分代码。

由验证密码改为验证数据库中是否有这一条记录。

**int** n = userDao.SelectUser(userName,userPassword);

**if**(n == 1){

.....

}