# 实验三 基于Hibernate的CRUD

一、实验目的：

熟悉并掌握运用Servlet开发模式(HTML+JSP+JavaBean+Servlet)和Hibernate实现数据库CRUD基本编程。

二、实验内容：

在MyEclipse环境下运用HTML+JSP+JavaBean+Servlet以及Hibernate技术完成相应的数据库CRUD功能，调试运行程序。

三、实验要求：

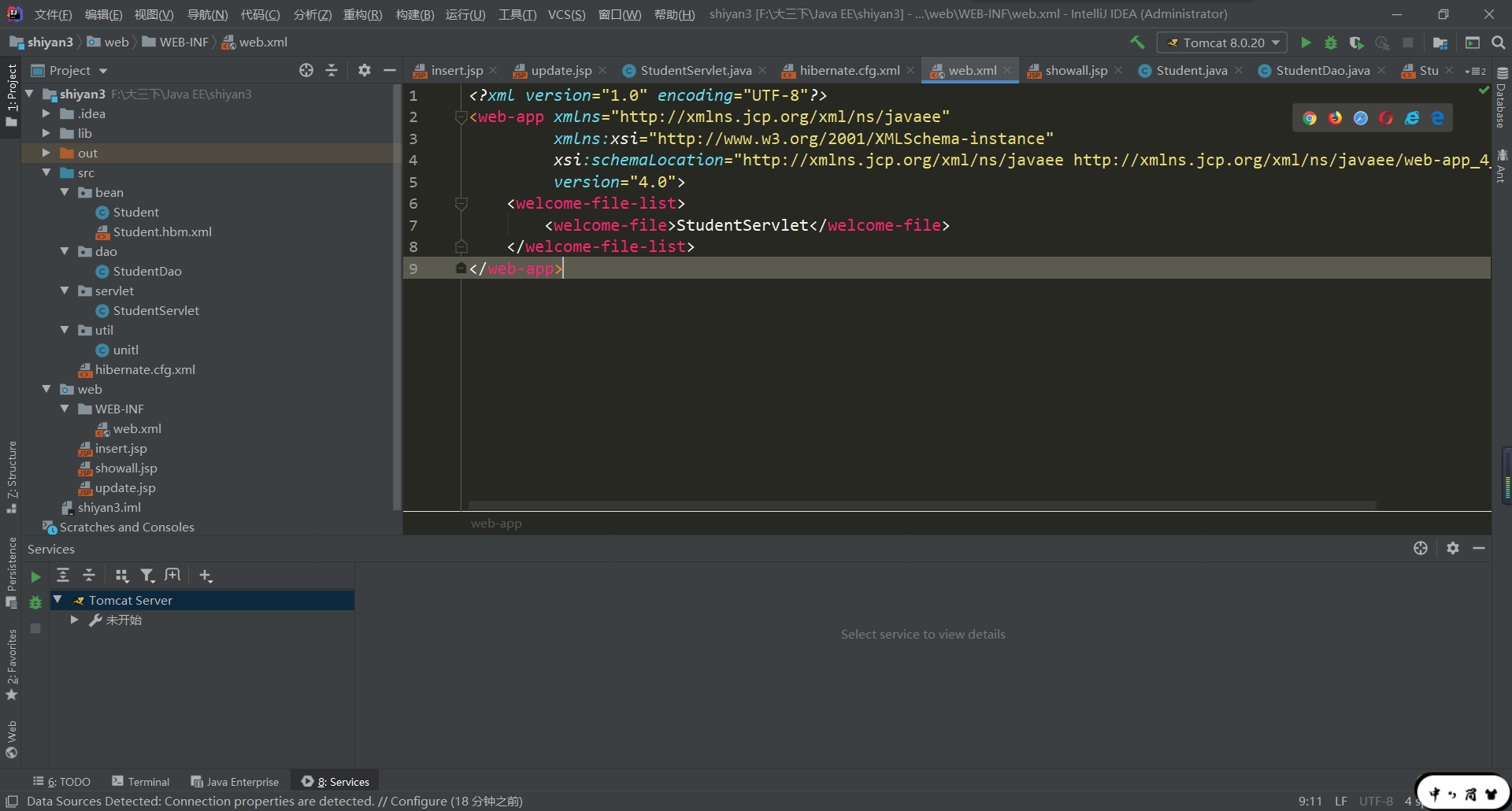
1. 熟悉并掌握运用MVC技术和Hibernate技术开发功能模块的基本步骤；

2. 运用相关技术（HTML+JSP+JavaBean+Servlet以及Hibernate）完成规定功能；

3. 写出实验报告。

四、实验步骤：

1．进入MyEclipse环境，新建一个Web Project。



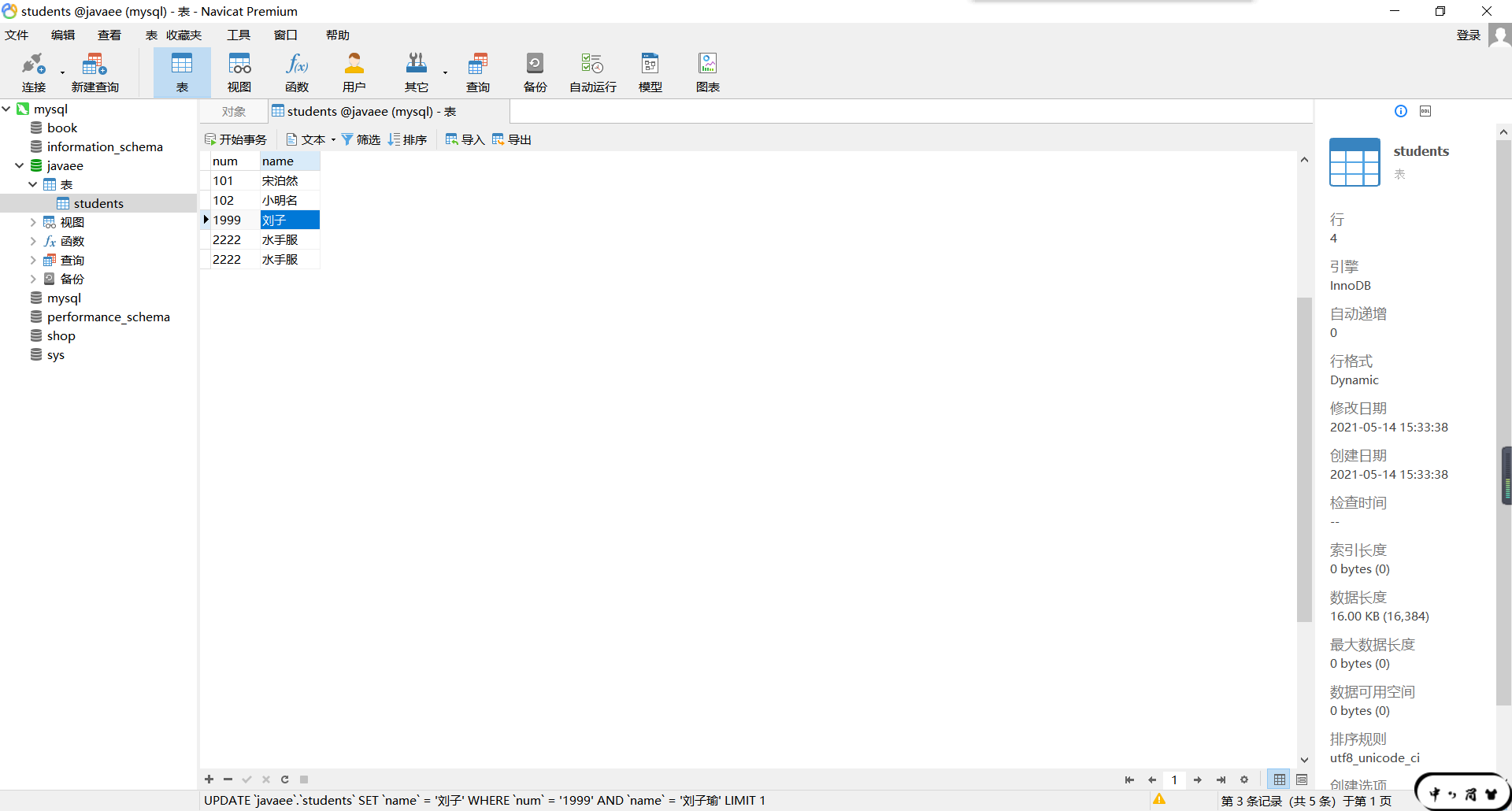
1. 设计数据库表结构。

CREATE TABLE `students` (

`num` varchar(255) NOT NULL,

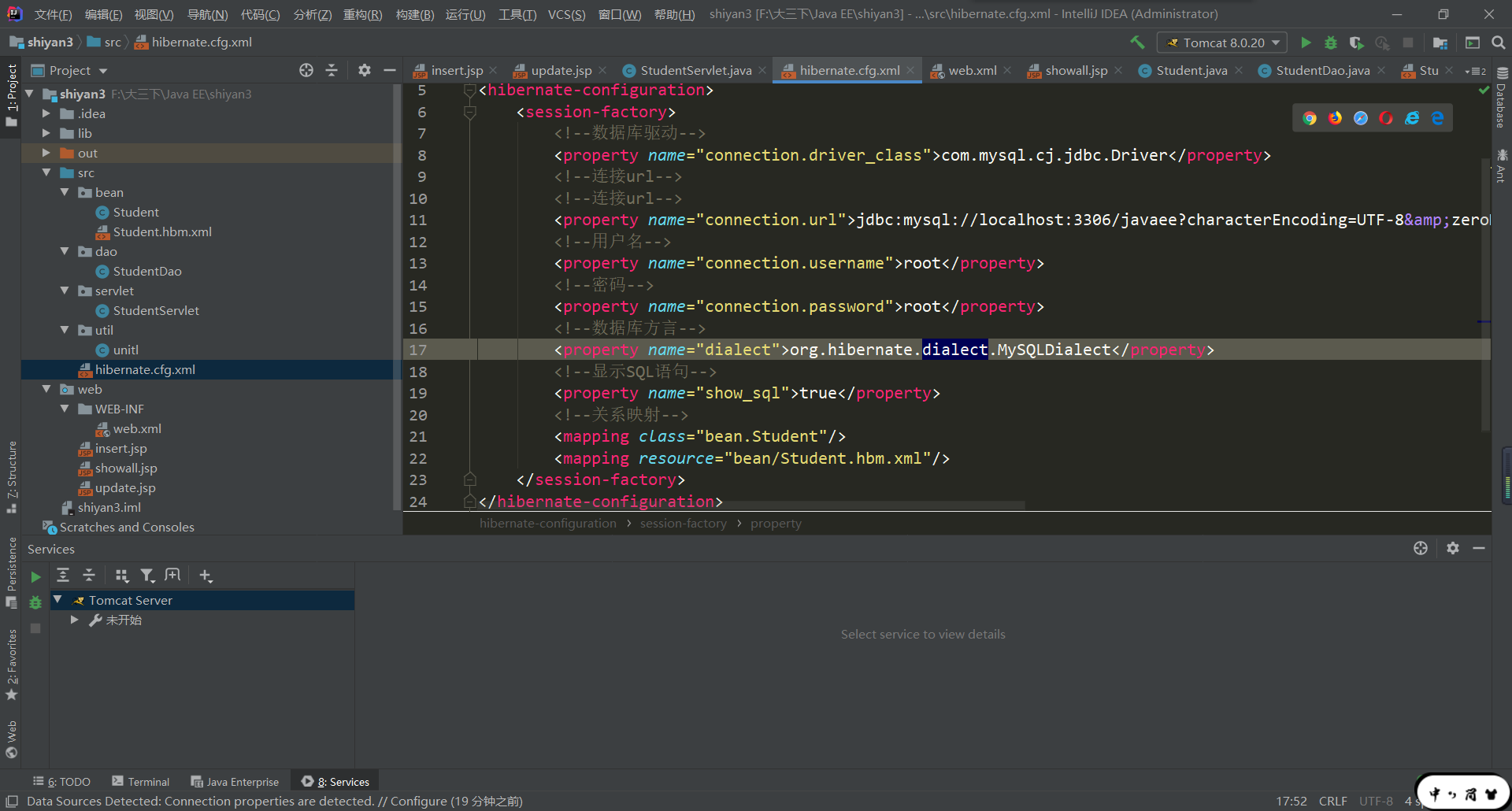
`name` varchar(255) NOT NULL

) ;

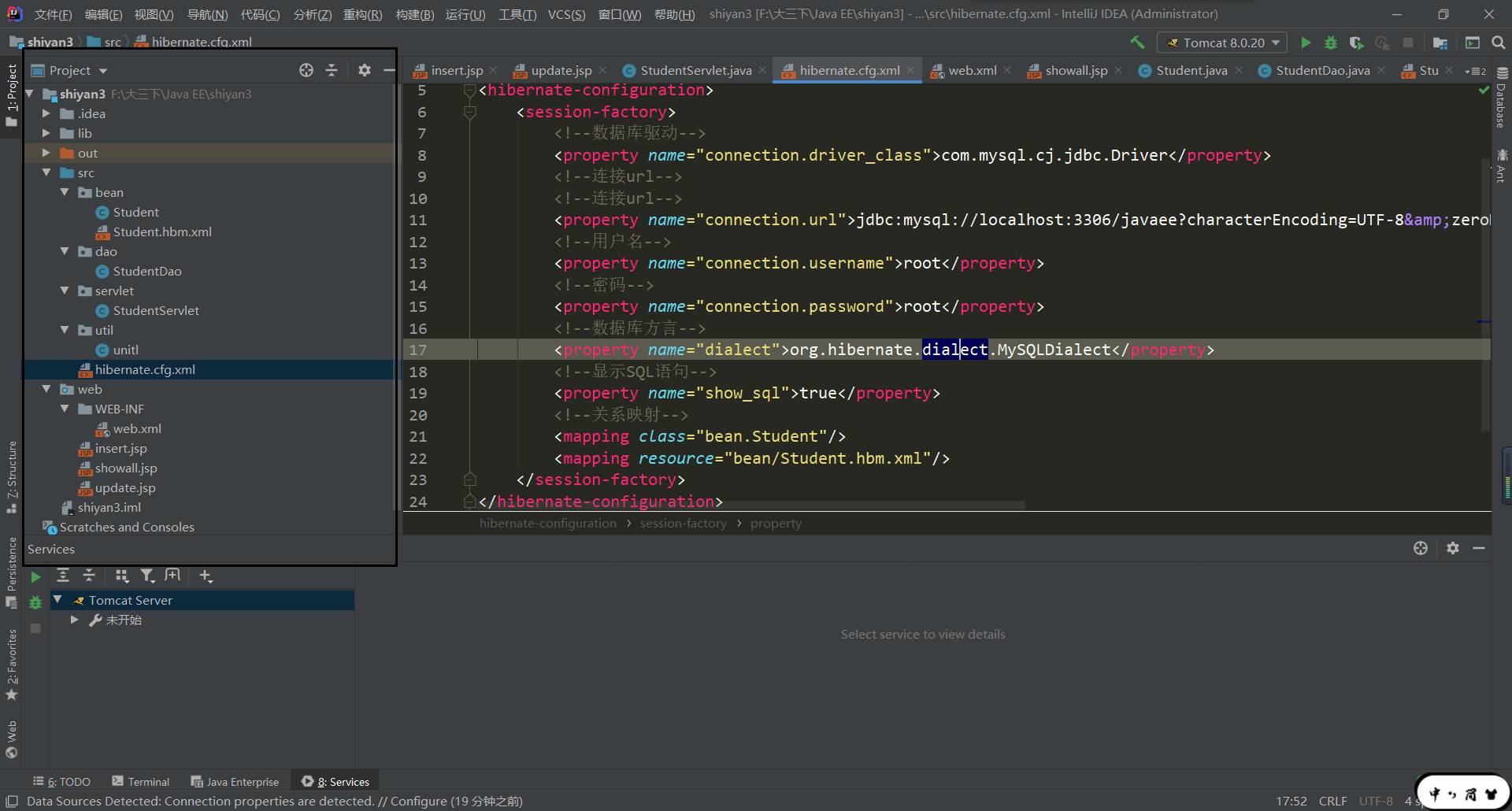


1. 采用MVC技术和Hibernate完成数据库的CRUD。

配置文件

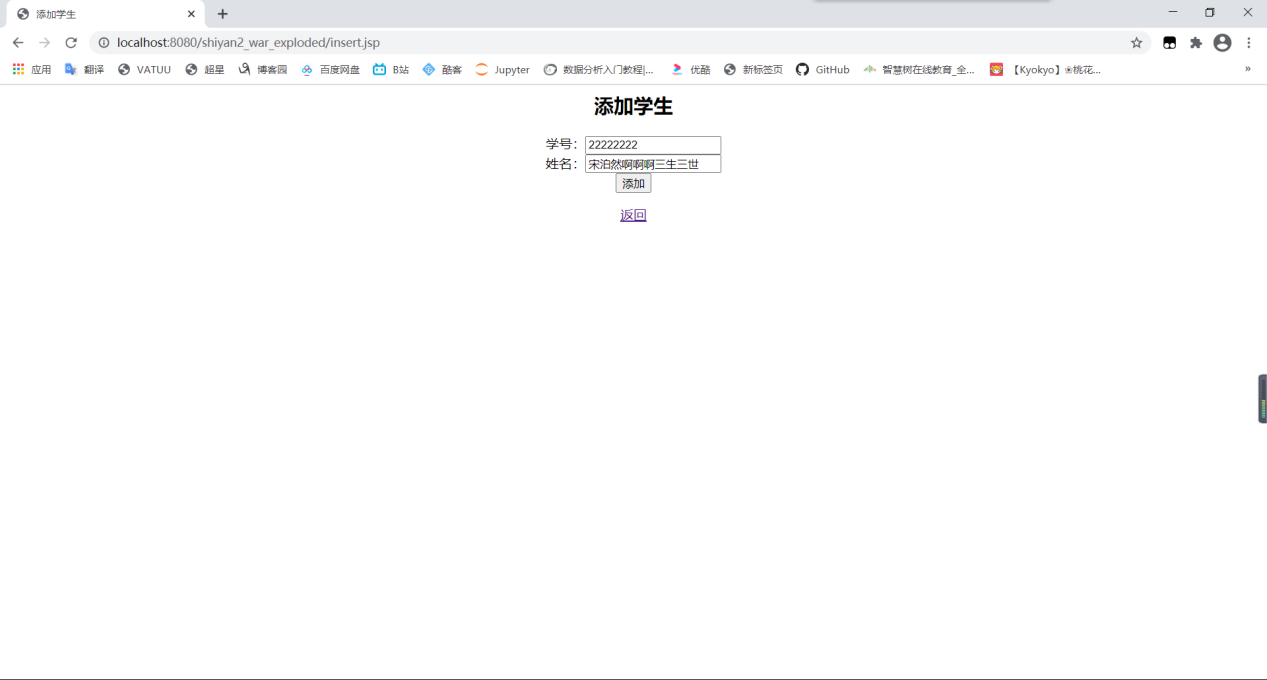
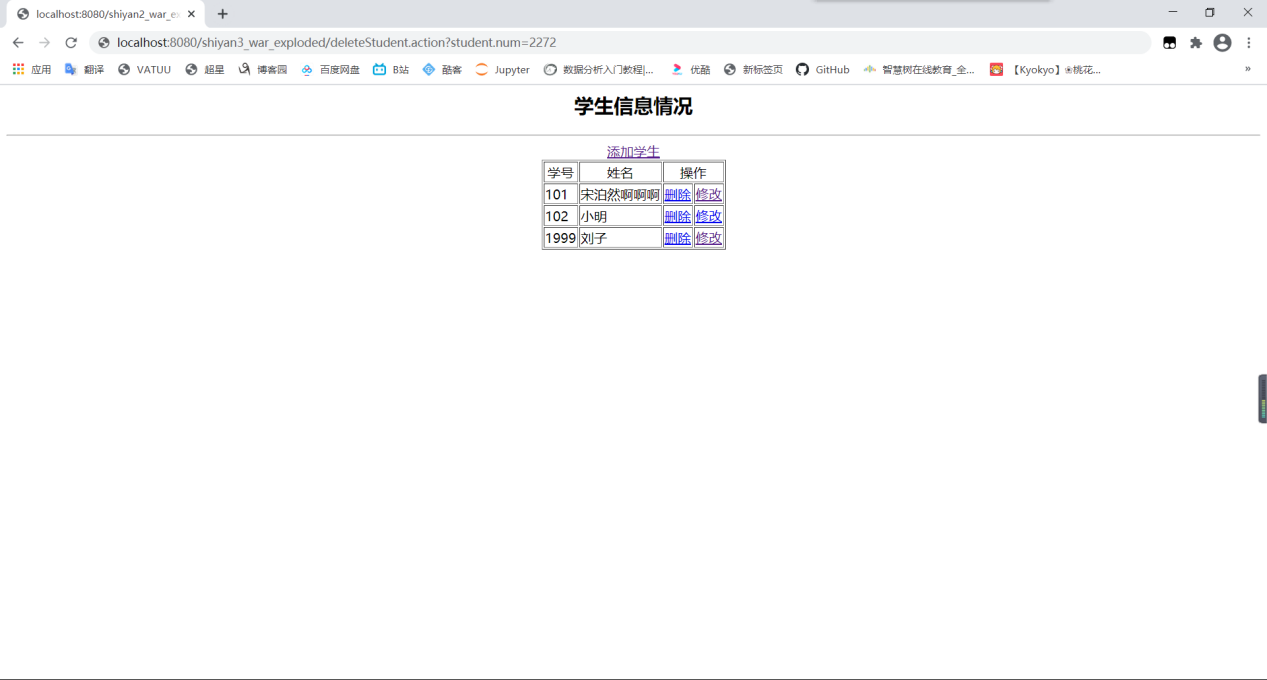
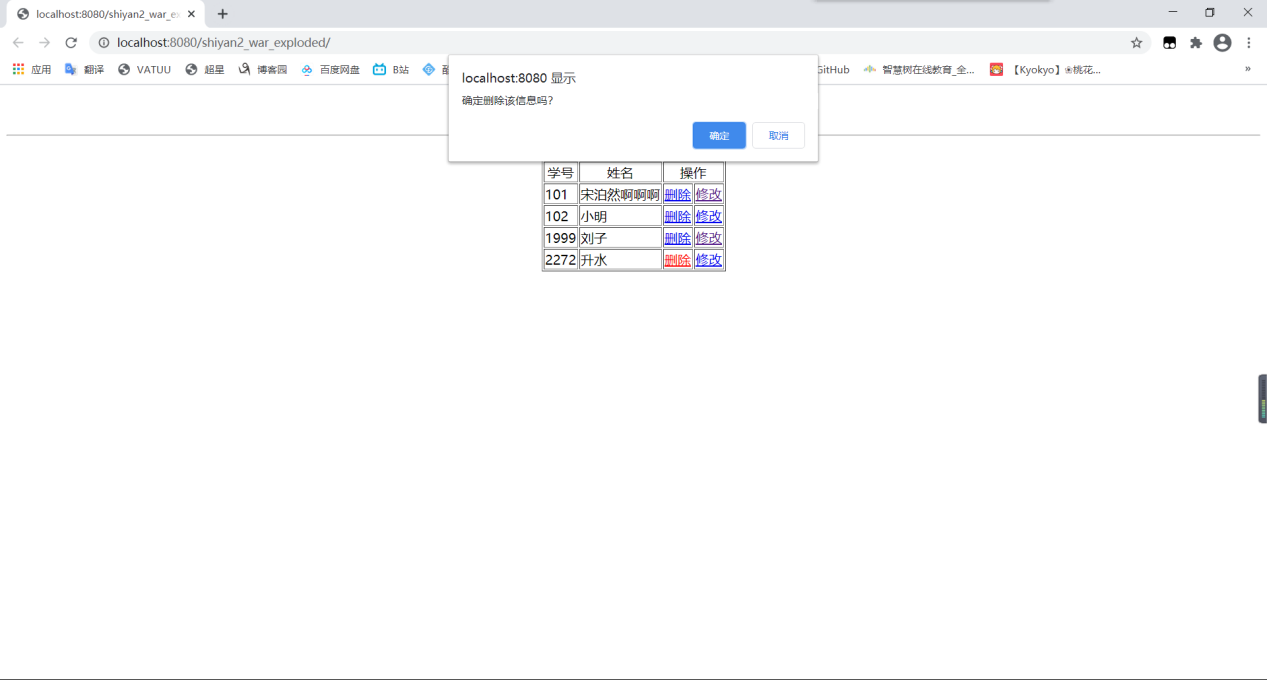


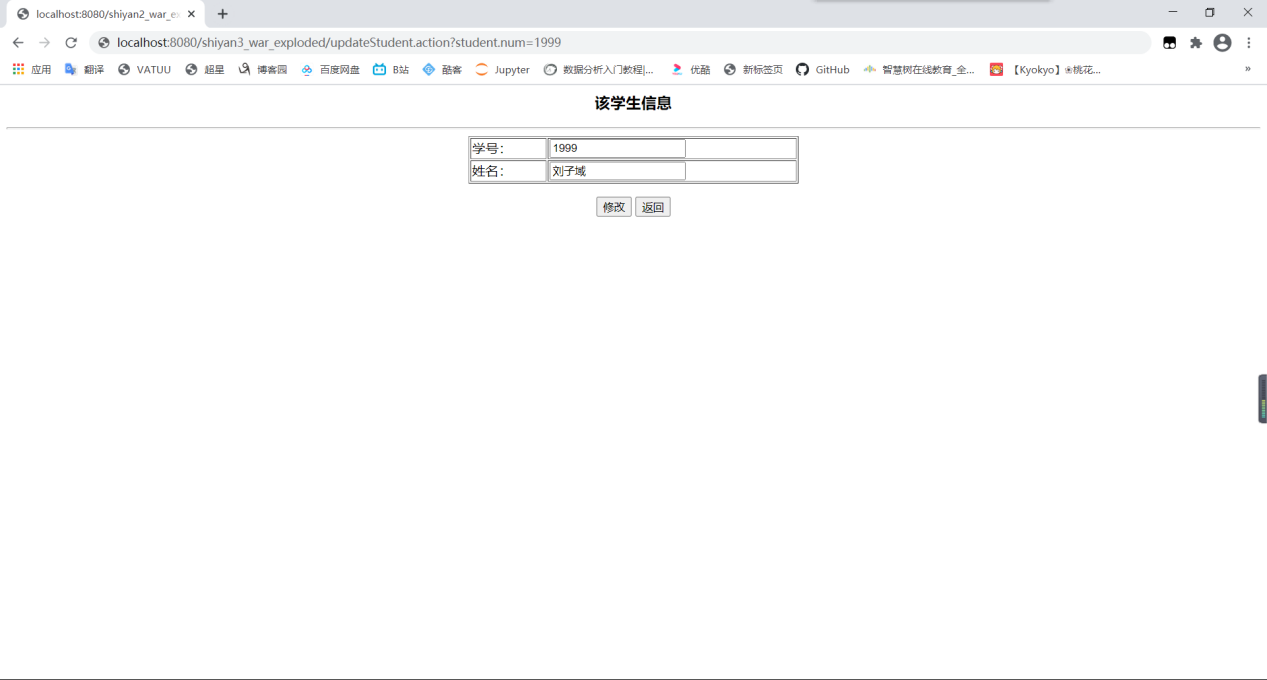
目录结构



1. 程序运行截图。







源代码：

Studentdao

package *dao*;  
  
import *bean.Student*;  
import *org.hibernate.Session*;  
import *org.hibernate.Transaction*;  
import *util.unitl*;  
  
import *java.util.List*;  
  
public class *StudentDao* {  
 //调用工具包  
 static *util.unitl* unitl=new unitl();  
 static *Session* session=unitl.lj();  
 int a=0;  
  
 /\*\*\*\*\*\*\*\*\*\*\*添加学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 public int addstudent(*String num*,*String name*){  
 int a=0;  
 //3、开启事务  
 *Transaction* ts=session.beginTransaction();  
 //执行数据库操作  
 //添加操作  
 *Student* student=new Student();  
 student.setNum(*num*);  
 student.setName(*name*);  
 session.save(student);  
 //提交  
 ts.commit();  
 //关闭session  
 //session.close();  
 a=1;  
 return a;  
 }  
  
 /\*\*\*\*\*\*\*\*\*\*\*查看所有学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 @SuppressWarnings("unchecked")  
 public *List*<*Student*> selectall(){  
 //执行数据库操作  
 //编写hql  
 *String* hql="from *Student* ";  
 *List*<*Student*> list=session.createQuery(hql).list();  
 return list;  
 }  
 /\*\*\*\*\*\*\*\*\*\*\*根据学号删除学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 public int delete(*String num*){  
 int a=0;  
 //3、开启事务  
 *Transaction* ts=session.beginTransaction();  
 //执行数据库操作  
 //删除操作  
 session.createQuery("delete from *Student* where num = *?0*")  
 .setString(0, *num*).executeUpdate() ;  
 //提交  
 ts.commit();  
 //关闭session  
 //session.close();  
 a=1;  
 return a;  
 }  
 /\*\*\*\*\*\*\*\*\*\*\*根据ID查询一个部门\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 public *List*<*Student*> selectone(*String num*){  
 //执行数据库操作  
 //编写hql  
 *String* hql="select new Student (num,name) from *Student* where num=*?0*";  
 *List*<*Student*> list=session.createQuery(hql)  
 .setString(0, *num*)  
 .list();  
 return list;  
 }  
 /\*\*\*\*\*\*\*\*\*\*\*修改部门\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 public void update(*String num*, *String name*){  
 //执行数据库操作  
 //编写hql  
 *Transaction* ts=session.beginTransaction();  
 session.createQuery("update *Student* s set s.name = *?1* where num =*?0*")  
 .setString(0, *num*).setString(1,*name*).executeUpdate() ;  
 //提交  
 ts.commit();  
 }  
  
}

Hibernate.cfg.xml

*<?xml version*='1.0' *encoding*='utf-8'*?>*<!DOCTYPE *hibernate-configuration* PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd"*>*<hibernate-configuration>  
 <session-factory>  
 <!--数据库驱动-->  
 <property *name*="connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <!--连接url-->  
 <!--连接url-->  
 <property *name*="connection.url">jdbc:mysql://localhost:3306/javaee?characterEncoding=UTF-8&amp;zeroDateTimeBehavior=convertToNull&amp;allowMultiQueries=true&amp;serverTimezone=UTC</property>  
 <!--用户名-->  
 <property *name*="connection.username">root</property>  
 <!--密码-->  
 <property *name*="connection.password">root</property>  
 <!--数据库方言-->  
 <property *name*="dialect">org.hibernate.dialect.MySQLDialect</property>  
 <!--显示SQL语句-->  
 <property *name*="show\_sql">true</property>  
 <!--关系映射-->  
 <mapping *class*="bean.Student"/>  
 <mapping *resource*="bean/Student.hbm.xml"/>  
 </session-factory>  
</hibernate-configuration>

Showall.jsp

<%@ page *contentType*="text/html;charset=UTF-8" *language*="java"%>  
<%@taglib *uri*="http://java.sun.com/jsp/jstl/core" *prefix*="c" %>  
<html>  
<head>  
 <title>学生信息</title>  
</head>  
<body>  
<center>  
<h1>学生信息</h1>  
<a *href*="insert.jsp">添加</a>  
<table *border*="1">  
 <tr>  
 <td>学号</td>  
 <td>姓名</td>  
 <td>操作</td>  
 </tr>  
 <c:forEach *items*="${list}" *var*="stu">  
 <tr>  
 <td>${stu.num}</td><td>${stu.name}</td>  
 <td><a *href*="StudentServlet?type=5&&num=${stu.num}">修改</a><!-- 修改前先查一个 -->  
     
 <a *href*="StudentServlet?type=3&&num=${stu.num}">删除</a></td>  
  
 </tr>  
 </c:forEach>  
</table>  
</center>  
</body>  
</html>

Insert.jsp

*<%--  
 Created by IntelliJ IDEA.  
 User: ASUS  
 Date: 2021/4/28  
 Time: 19:57  
 To change this template use File | Settings | File Templates.  
--%>*<%@ page *contentType*="text/html;charset=UTF-8" *language*="java" %>  
<html>  
<head>  
 <title>添加学生</title>  
</head>  
<body>  
<center>  
<h1>添加学生</h1>  
<form *action*="StudentServlet" *method*="post">  
 <input *type*="hidden" *value*="1" *name*="type">  
 学号：<input *type*="text" *name*="num"/><br/>  
 姓名：<input *type*="text" *name*="name"/><br/>  
 <input *type*="submit" *value*="提交">  
</form>  
<a *href*="StudentServlet">返回</a>  
</center>  
</body>  
  
</html>

五、心得体会：

现在虽然比较流行ssm，但是ssh也依然很好，hibernate的dao层也要比mybatis更简单，mybatis需要维护sql和结果映射，hibernate对对象的维护和缓存要比mybatis好，对增删改查的对象维护要方便，hibernate数据库移植性好，mybatis不同的数据库要写不同的sql，移植性差，但是mybatis比hibernate更容易掌握，可以进行sql优化。