浙江工业大学

Web 应用开发上机作业

2022 级

实验题目： MVC 设计模式的应用

学生姓名：李飞飞

学生学号：2022

任课教师：赵　小　敏

专业班级：2022 软件工程（移动应用开发 1）

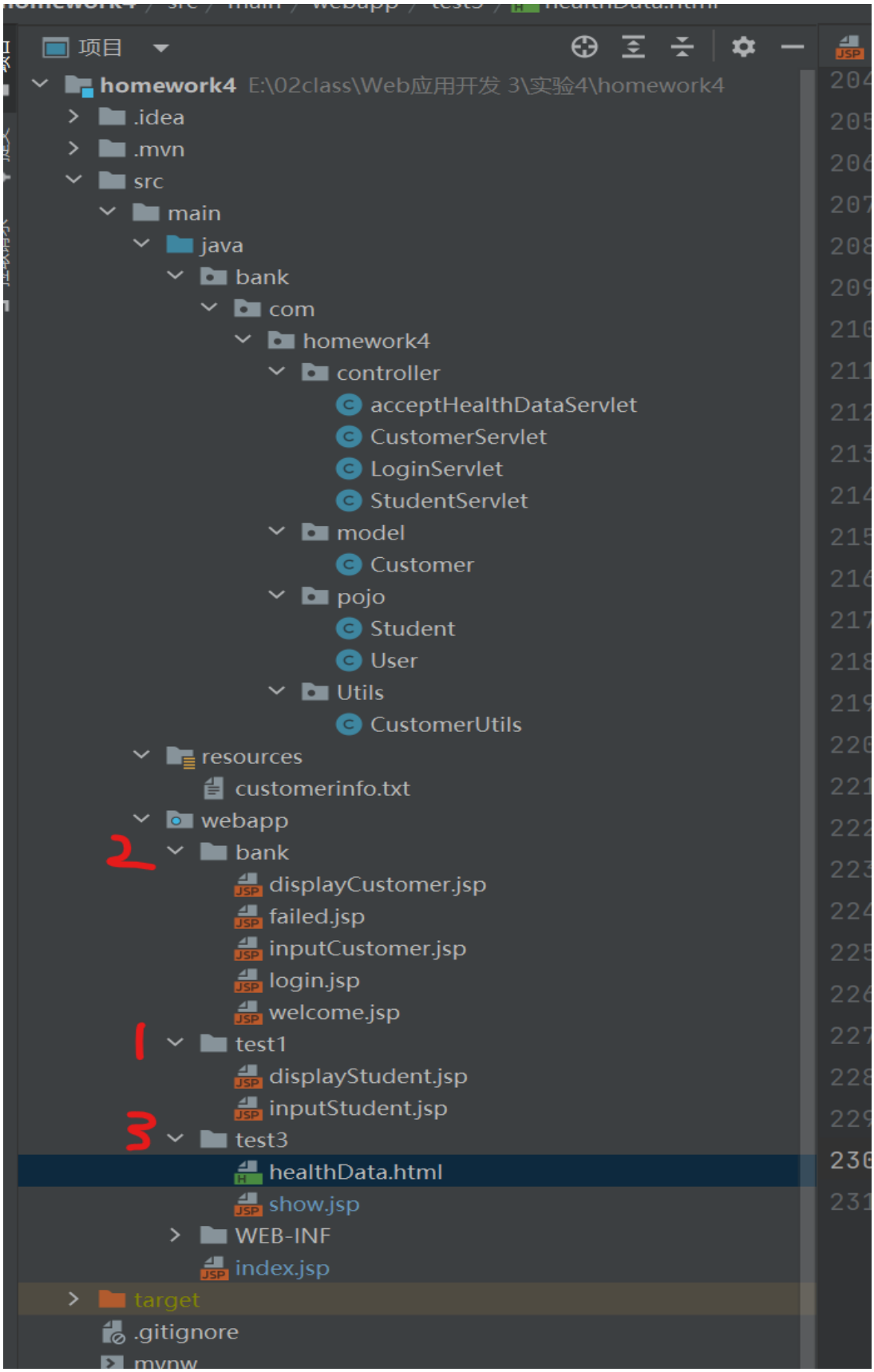
学　　院：计算机科学与技术学院

2024 年 4 月 4 日

目　　录

写在一个 maven 项目里，点击主页链接跳转对应实验

下图是项目目录结构，前端页面分离三个模块，后端 controller 放在一起了，实体类放 pojo 里（实验 2 根据要求放 model 里了） 第三题引入了生成二维码的依赖，前端动态引入 vue.Js 所以要等他加载一下才能正常显示


项目
homework4 E:\02class\Web应用开发 3\实验4\homework4
.idea
.mvn
src
main
java
bank
com
homework4
controller
acceptHealthDataServlet
CustomerServlet
LoginServlet
StudentServlet
model
Customer
pojo
Student
User
Utils
CustomerUtils
resources
customerinfo.txt
webapp
2
bank
displayCustomer.jsp
failed.jsp
inputCustomer.jsp
login.jsp
welcome.jsp
1
test1
displayStudent.jsp
inputStudent.jsp
3
test3
healthData.html
show.jsp
WEB-INF
index.jsp
target
.gitignore
mvnw

1、学生信息管理系统

1.1 题目要求：

某学校的学生信息管理系统有一个录入学生信息的功能，请按要求编写程序。

（1）编写一个名为 Student 的 JavaBeans，包括 3 个属性：stuid 表示学号、name 表示学生姓名和 major 表示专业；

（2）编写输入学生信息页面 inputStudent.jsp，通过表单输入学生信息，将请求转发到 StudentServlet；

（3）StudentServlet 从 JSP 页面得到学生信息，并将学生信息通过作用域共享后转发至学生信息显示页面 displayStudent.jsp；

（4）displayStudent.jsp 显示学生的信息。

1.2 需求分析

需要开发一个学生信息管理系统的学生信息录入功能，该功能包括以下步骤：

- 用户在输入学生信息页面填写学生的学号、姓名和专业信息。
- 用户提交表单后，将学生信息传递给后端的 Servlet 处理。
- 后端 Servlet 接收学生信息，并将其存储在数据库中。
- 学生信息存储成功后，跳转到学生信息显示页面，展示刚录入的学生信息。

1.2.1 功能描述

1. 输入学生信息功能

1.1 访问输入学生信息页面：

用户通过浏览器访问系统的输入学生信息页面。

1.2 填写学生信息：

在输入学生信息页面，用户填写学生的学号、姓名和专业信息。

1.3 提交学生信息：

用户填写完毕后，点击提交按钮将表单数据发送至后端 Servlet 进行处理。

2. 后端处理功能

2.1 接收学生信息：

后端 Servlet 接收从输入学生信息页面提交的学生信息数据。

2.2 存储学生信息：

后端 Servlet 将接收到的学生信息存储到数据库中。

3. 学生信息显示功能

3.1 显示学生信息页面：

学生信息录入成功后，用户被重定向到学生信息显示页面。

3.2 展示学生信息：

在学生信息显示页面，系统展示刚录入的学生信息，包括学号、姓名和专业信息。

1.2.2 业务处理流程及要求

用户访问输入学生信息页面（inputStudent.jsp）。

用户在页面上填写学生的学号、姓名和专业信息。

用户点击提交按钮，将表单数据发送到后端的 Servlet（StudentServlet）。

Servlet 接收到学生信息后，将其存储到数据库中。

Servlet 将存储成功的学生信息传递到学生信息显示页面（displayStudent.jsp）。

学生信息显示页面展示刚录入的学生信息。

1.2.3 输入信息

学生信息

1.2.4 输出信息

学生信息展示

1.3 开发步骤

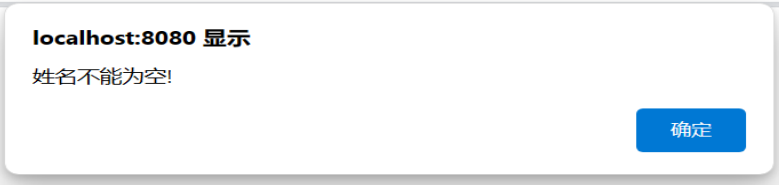
先创建 student.java 再编写 inputStudent.jsp 和 StudentServlet，最后写 displayStudent.jsp

1.4 运行结果

- 信息录入

- 信息校验

其它格式校验也有，不再赘述



- 信息展示

1.5 问题及解决方案

无

1.6 实验收获

1. **JavaBeans 的运用**： 通过编写名为 Student 的 JavaBeans，我学会了如何创建一个简单的 Java 类来表示业务实体。这让我意识到 JavaBeans 的重要性，它可以使数据的封装更加简洁和方便管理。
2. **JSP 页面的开发**： 编写输入学生信息页面 inputStudent.jsp 和学生信息显示页面 displayStudent.jsp，我熟悉了如何在 JSP 中编写 HTML 和嵌入 Java 代码。这让我更加了解了 JSP 的使用和原理。

1.7 源代码

信息录入没什么好看的都放了几百遍了，这里就不放了，完整代码见附件

- 学生类

```java
package bank.com.homework4.pojo;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 17:56 2024/4/3
 * @ Description: 实验 1 学生类
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Student {
    private String stuid;
    private String name;
    private String major;
}
```

- StudentServlet

```java
package bank.com.homework4.controller;
/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 17:58 2024/4/3
 * @ Description: ${description}
 */

import bank.com.homework4.pojo.Student;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "StudentServlet", value = "/StudentServlet")
public class StudentServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }
```

```
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        Student student=(Student)
getServletContext().getAttribute("Student");
        if(student==null){
            student=new Student();
        }
        student.setStuid(request.getParameter("stuid"));
        student.setName(request.getParameter("name"));
        student.setMajor(request.getParameter("major"));
        request.getSession().setAttribute("student",student);
        //转发

request.getRequestDispatcher("./test1/displayStudent.jsp").forward(re
quest,response);
    }
}
```

- displayStudent

```
<%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/4/3
  Time: 17:59
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>展示学生</title>
</head>
<body>
<jsp:useBean id="student" scope="session"
class="bank.com.homework4.pojo.Student"></jsp:useBean>
<h1>学生信息展示页面</h1>
<h1>学号: <jsp:getProperty name="student" property="stuid"/></h1>
<h1>姓名: <jsp:getProperty name="student" property="name"/></h1>
<h1>专业: <jsp:getProperty name="student" property="major"/></h1>
</body>
</html>
```

2. MVC 银行

2.1 题目要求

某银行客户管理系统采用 MVC 模式设计实现，其中有一个注册客户信息和登陆的功能，请按如下要求编写程序。

（1）编写一个类为 bank.com.model.Customer 的 JavaBeans，包括 4 个属性：email 表示客户邮箱，password 表示密码， custName 表示客户姓名，phone 表示客户手机号；

（2）编写一个输入客户信息页面 bank/inputCustomer.jsp，通过表单输入客户信息，将请求转发到映射地址为 CustomerServlet.do 的 bank.com.control.CustomerServlet 类进行处理；

（3）编写一个 bank.com.control.CustomerServlet 类，从输入客户信息页面得到客户信息存入文件 WEB-INF/customerinfo.txt 文件，并将客户信息通过作用域共享后转发至客户显示页面 bank/displayCustomer.jsp；

（4）bank/displayCustomer.jsp 显示某客户的邮箱、姓名和手机号；

（5）bank/login.jsp 为登陆页面，可输入邮箱和密码

（6）编写一个 bank.com.control.LoginServlet 类，获取登陆页面的邮箱和密码并进行判断，登陆成功后跳转至 welcome.jsp 页面，否则跳转至失败页面。

2.2 需求分析

1. 注册客户信息功能

1.1 输入客户信息页面（inputCustomer.jsp)

- 用户访问输入客户信息页面。
- 用户通过表单填写客户的邮箱、密码、姓名和手机号。
- 用户提交表单数据，请求将被发送到 CustomerServlet.do。

1.2 处理客户信息（CustomerServlet.do）

- CustomerServlet.do 接收来自输入客户信息页面的请求。
- 从请求中获取客户信息，包括邮箱、密码、姓名和手机号。
- 将客户信息存储到名为 WEB-INF/customerinfo.txt 的文件中。
- 将客户信息通过作用域共享，转发至客户显示页面。

2. 客户信息显示功能

2.1 显示客户信息页面（displayCustomer.jsp）

- 客户信息显示页面展示某客户的邮箱、姓名和手机号。
- 从作用域获取客户信息，包括邮箱、姓名和手机号。
- 将客户信息显示在页面上。

3. 登陆功能

3.1 登陆页面（login.jsp）

- 用户访问登陆页面。
- 用户可以输入邮箱和密码进行登陆。

3.2 处理登陆请求（LoginServlet）

- LoginServlet 接收来自登陆页面的请求。
- 从请求中获取用户输入的邮箱和密码。
- 验证邮箱和密码是否与注册信息匹配。
- 如果验证成功，将用户重定向到 welcome.jsp 页面。
- 如果验证失败，将用户重定向到失败页面。

2.2.1 功能描述

注册客户信息功能

1. 输入客户信息页面 (inputCustomer.jsp)

1.1 访问页面：

- 用户通过浏览器访问输入客户信息页面。

1.2 填写客户信息：

- 在输入客户信息页面，用户填写客户的邮箱、密码、姓名和手机号。

1.3 提交客户信息：

- 用户填写完毕后，点击提交按钮将表单数据发送至 CustomerServlet 进行处理。

2. 处理客户信息 (CustomerServlet)

2.1 接收请求：

- CustomerServlet 接收来自输入客户信息页面的请求。

2.2 获取客户信息：

- 从请求中获取客户的邮箱、密码、姓名和手机号。

2.3 存储客户信息：

- 将客户信息存储到名为 WEB-INF/customerinfo.txt 的文件中。

2.4 转发客户信息：

- 将客户信息通过作用域共享，转发至客户显示页面。

客户信息显示功能

3. 显示客户信息页面（displayCustomer.jsp）

3.1 **访问页面**：

- 用户被重定向到客户信息显示页面。

3.2 **展示客户信息**：

- 在客户信息显示页面，系统展示某客户的邮箱、姓名和手机号。

登陆功能

4. 登陆页面（login.jsp）

4.1 **访问页面**：

- 用户通过浏览器访问登陆页面。

4.2 **输入登陆信息**：

- 用户在登陆页面输入邮箱和密码。

4.3 **提交登陆信息**：

- 用户填写完毕后，点击提交按钮将表单数据发送至 LoginServlet 进行处理。

5. 处理登陆请求（LoginServlet）

5.1 **接收请求**：

- LoginServlet 接收来自登陆页面的请求。

5.2 **验证登陆信息**：

- 从请求中获取用户输入的邮箱和密码。
- 验证邮箱和密码是否与注册信息匹配。

5.3 **处理登陆结果**：

- 如果验证成功，将用户重定向到 welcome.jsp 页面。
- 如果验证失败，将用户重定向到失败页面。

2.2.2 业务处理流程及要求

业务处理流程及要求

1. 注册客户信息功能流程

用户访问输入客户信息页面 bank/inputCustomer.jsp。

用户填写客户信息并提交表单。

请求被转发到 CustomerServlet 处理。

CustomerServlet 获取客户信息并存入文件 WEB-INF/customerinfo.txt。

客户信息通过作用域共享后转发至客户显示页面 bank/displayCustomer.jsp。

displayCustomer.jsp 页面显示某客户的邮箱、姓名和手机号。

2. 登陆功能流程

用户访问登陆页面 bank/login.jsp。

用户输入邮箱和密码并提交表单。

请求被转发到 LoginServlet 处理。

LoginServlet 获取登陆页面的邮箱和密码并进行验证。

如果验证成功，跳转至欢迎页面 welcome.jsp；否则跳转至失败页面。

业务处理要求

数据持久化要求：

客户信息应当被存储到名为 customerinfo.txt 的文件中，以便之后的访问和操作。

页面跳转要求：

在注册和登陆过程中，页面跳转应当符合用户的预期，提供友好的交互体验。

用户信息验证要求：

在注册过程中，应当对用户输入的邮箱、密码、姓名和手机号进行验证，确保数据的准确性和完整性。

在登陆过程中，应当验证用户输入的邮箱和密码是否与已注册的信息匹配，以确保登陆的安全性。

错误处理要求：

在用户输入不合法信息或发生其他错误时，应当给予相应的提示，引导用户正确操作。

2.2.3 输入信息

注册：邮箱，密码，客户姓名，客户手机号

登录：邮箱，密码

2.2.4 输出信息

注册：返回用户信息
登录：成功返回欢迎页面消息，失败返回失败页面消息

2.3 开发步骤

分别编写用户类，用户工具类，登录，注册模块

注册模块包括注册页面，请求跳转 controller CustomerServlet.do，用户信息页面

登录模块包括登录页面，登录请求跳转 controller LoginServlet,成功与失败页面

用户工具类用于 1.加载所有现存用户 2.保存一个用户

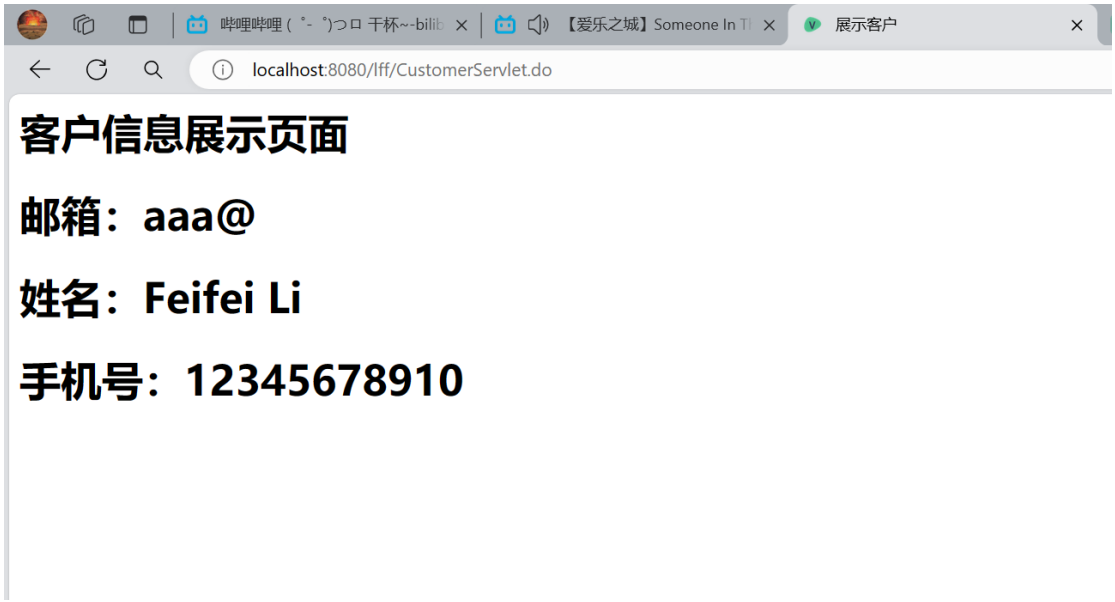
2.4 运行结果

- 注册页面

- 信息校验

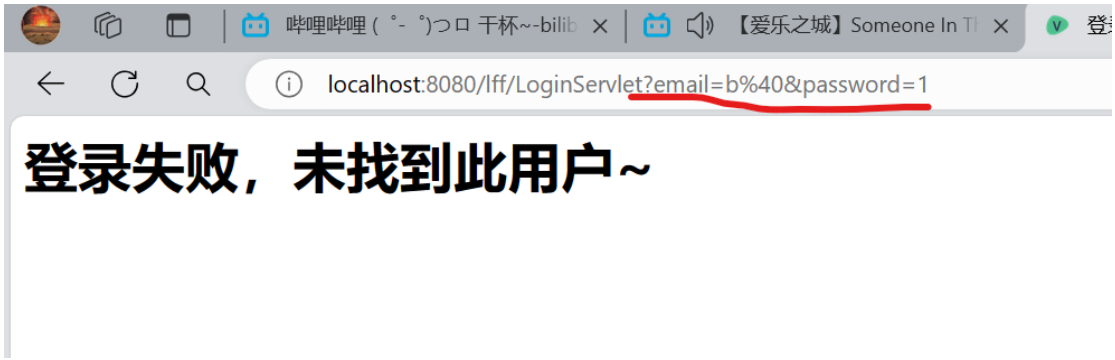
其它的格式校验也有，不再赘述

- 注册成功

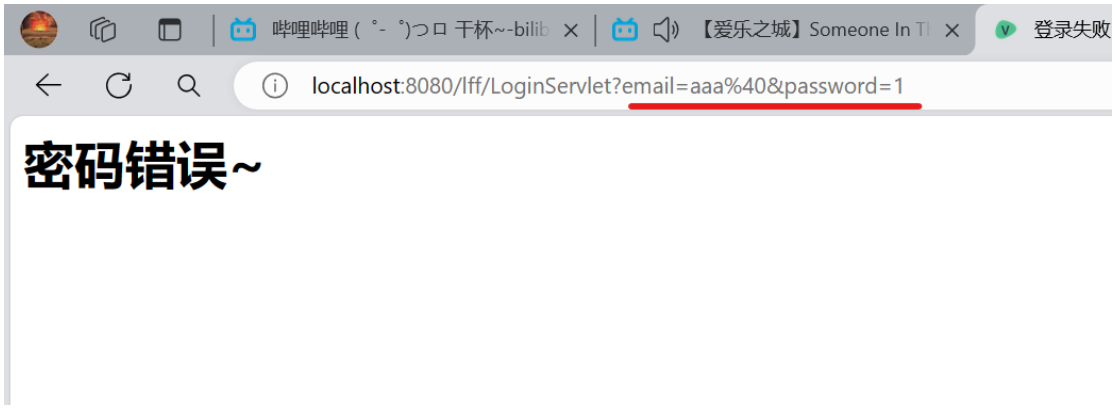


- 登录页面

- 登录失败——找不到用户



- 登录失败——密码错误



- 登录成功

2.5 问题及解决方案

无

2.6 实验收获

1. **理解 MVC 模式：** 通过这个项目，我深入理解了 MVC（Model-View-Controller）模式的概念和工作原理。在这个项目中，我们将业务逻辑（控制器）、数据（模型）和用户界面（视图）分离开来，使得系统更易于维护和扩展。
2. **JavaBeans 的运用：** 在项目中，我学会了如何编写 JavaBeans 来表示业务实体，以及如何在 Servlet 中使用 JavaBeans 来处理数据。这使得代码更加模块化和可重用。

2.7 源代码

只展示部分，其它无关紧要的放附件

- Custumer 类

```
package bank.com.homework4.model;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.Vector;

/*
 * @ Author      : Li Feifei
 * @ Date        : Created in 1:33 2024/4/4
 * @ Description: 实验 2Customer 类
 */
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Customer {
    public Vector<Customer> customers;
    private String email;
    private String password;
    private String custName;
    private String phone;
}
```

- CustumerUtils 工具类

```
package bank.com.homework4.Utils;

import bank.com.homework4.model.Customer;

import java.io.*;
import java.util.Objects;
import java.util.Vector;

/*
 * @ Author      : Li Feifei
 * @ Date        : Created in 1:54 2024/4/4
 * @ Description: 实验 2 工具类
 */
public class CustomerUtils {
```

```
  //刷新 CustomerBean 的 Vector
  public static void refreshCustomers(Customer customer) {
    Vector<Customer> now = new Vector<>();

    try (BufferedReader reader = new BufferedReader(
            new FileReader("customerinfo.txt"))
    ) {
      String line;
      while ((line = reader.readLine()) != null) {
        // 按 "|" 分隔行数据
        String[] fields = line.split("\\|");
        if (fields.length == 4) // 确保每行数据都包含了所有字段
          now.add(new Customer(null, fields[0], fields[1], fields[2],
fields[3]));
      }
    } catch (IOException e) {
      e.printStackTrace();
    }

    customer.setCustomers(now);
  }
  //保存一个 Customer
  public static void saveCustomer(Customer customer) {
    try (BufferedWriter writer = new BufferedWriter(
            new FileWriter("customerinfo.txt", true))) {

      String userInfo = customer.getEmail() + "|" +
customer.getPassword() + "|" +
              customer.getCustName() + "|" + customer.getPhone();

      // 将用户信息写入文件
      writer.write(userInfo);
      writer.newLine(); // 换行

    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

- LoginServlet

```
package bank.com.homework4.controller;
/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 1:37 2024/4/4
```

```
 * @ Description: ${description}
 */

import bank.com.homework4.Utils.CustomerUtils;
import bank.com.homework4.model.Customer;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "LoginServlet", value = "/LoginServlet")
public class LoginServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        Customer customer=(Customer)
getServletContext().getAttribute("Customer");
        if(customer==null){
            customer=new Customer();
        }
        String email=request.getParameter("email");
        String password=request.getParameter("password");
        String msg;
        //刷新用户列表
        CustomerUtils.refreshCustomers(customer);
        System.out.println("当前用户:
"+customer.getCustomers().toString());
        for(Customer now:customer.getCustomers()){
            if(now.getEmail().equals(email)){
                if(now.getPassword().equals(password)) {
                    msg = "欢迎" + now.getCustName() + "登录成功~";
                    request.getSession().setAttribute("msg",msg);

request.getRequestDispatcher("./bank/welcome.jsp").forward(request,re
sponse);
                    return;
                }
                else {
                    msg = "密码错误~";
                    request.getSession().setAttribute("msg",msg);

request.getRequestDispatcher("./bank/failed.jsp").forward(request,res
ponse);
```

```
                    return;
                }
            }
        }
        msg="登录失败，未找到此用户~";
        request.getSession().setAttribute("msg",msg);

request.getRequestDispatcher("./bank/failed.jsp").forward(request,res
ponse);
    }

    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }
}
```

- CustomerServlet.do

```
package bank.com.homework4.controller;
/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 1:36 2024/4/4
 * @ Description: ${description}
 */

import bank.com.homework4.Utils.CustomerUtils;
import bank.com.homework4.model.Customer;
import bank.com.homework4.pojo.Student;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "CustomerServlet", value = "/CustomerServlet.do")
public class CustomerServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }

    @Override
    protected void doPost(HttpServletRequest request,
```

```
HttpServletResponse response) throws ServletException, IOException {
        Customer customer=(Customer)
getServletContext().getAttribute("Customer");
        if(customer==null){
            customer=new Customer();
        }
        customer.setEmail(request.getParameter("email"));
        customer.setPassword(request.getParameter("password"));
        customer.setCustName(request.getParameter("name"));
        customer.setPhone(request.getParameter("phone"));
        CustomerUtils.saveCustomer(customer);
        request.getSession().setAttribute("customer",customer);


request.getRequestDispatcher("./bank/displayCustomer.jsp").forward(re
quest,response);
    }
}
```

- displayCustomer.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/4/4
  Time: 1:36
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>展示客户</title>
</head>
<body>
<jsp:useBean id="customer" scope="session"
class="bank.com.homework4.model.Customer"></jsp:useBean>
<h1>客户信息展示页面</h1>
<h1>邮箱：<jsp:getProperty name="customer" property="email"/></h1>
<h1>姓名：<jsp:getProperty name="customer" property="custName"/></h1>
<h1>手机号：<jsp:getProperty name="customer" property="phone"/></h1>
</body>
</html>
```

3. 申请健康码演示程序

3.1 题目要求

编写一个申请健康码演示程序。

申请健康码页面要求填写以下信息：

① 个人信息，包括姓名、身份证号、工号或学号、手机号。

② 本人近期（14 天内）是否去过重点疫区？

③ 本人近期（14 天内）是否去过国外？

④ 本人近期（14 天内）是否接触过新冠确诊病人或疑似病人？

⑤ 本人是否被卫生部门确认为新冠肺炎确诊病例或疑似病例？

⑥ 当前健康状况？无异常、发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻等。

⑦ 本人郑重承诺：填报信息真实，愿意承担相应的法律责任。

- 如果有下列任意一种情况的为黄色健康码
    - 本人近期（14 天内）去过疫区
    - 本人近期（14 天内）去过国外
    - 当前健康状况有且仅有发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻中的一种
- 如果有下列任意一种情况的为红色健康码
    - 本人近期（14 天内）接触过新冠确诊病人或疑似病人
    - 本人被卫生部门确认为新冠肺炎确诊病例或疑似病例
    - 当前健康状况有发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻等中两种症状及以上
- 其他为绿色健康码
- 健康码（二维码）信息包括姓名、工号或学号、生成时间等

3.2 需求分析

1. 申请健康码页面设计

1.1 **个人信息填写**：

- 页面包含输入框，用于填写个人信息，包括姓名、身份证号、工号或学号、手机号。

1.2 **是否去过重点疫区、国外**：

- 提供单选框或复选框，让用户选择是否在近期（14 天内）去过重点疫区或国外。

1.3 **是否接触过新冠确诊病人或疑似病人：**

- 提供单选框或复选框，让用户选择是否在近期（14 天内）接触过新冠确诊病人或疑似病人。

1.4 **当前健康状况选择：**

- 提供单选框或下拉菜单，让用户选择当前健康状况，包括无异常、发烧、乏力、干咳、鼻塞、流涕、咽痛、腹泻等。

1.5 **承诺填报信息真实：**

- 提供复选框，让用户勾选承诺填报信息真实，并愿意承担相应的法律责任。

2. 健康码生成规则

2.1 **生成健康码颜色判断：**

- 根据用户填写的信息进行判断，生成相应颜色的健康码。

2.2 **健康码颜色判断规则：**

- 如果满足黄色健康码条件，则生成黄色健康码。
- 如果满足红色健康码条件，则生成红色健康码。
- 其余情况生成绿色健康码。

3. 健康码信息展示

3.1 **健康码信息包括：**

- 健康码（二维码）包括姓名、工号或学号、生成时间等信息。

4. 页面跳转

4.1 **页面跳转规则：**

- 提交申请后，根据生成的健康码颜色，跳转至相应的页面，展示健康码信息。

3.2.1 功能描述

1. 申请健康码页面设计

1.1 个人信息填写：

用户访问申请健康码页面，填写个人信息，包括姓名、身份证号、工号或学号、手机号。

1.2 是否去过重点疫区、国外：

提供选择框，用户选择是否在近期（14 天内）去过重点疫区或国外。

1.3 是否接触过新冠确诊病人或疑似病人：

提供选择框，用户选择是否在近期（14 天内）接触过新冠确诊病人或疑似病人。

1.4 当前健康状况选择：

提供选择框，用户选择当前健康状况，包括无异常、发烧、乏力、干咳、鼻塞、流涕、咽痛、腹泻等。

1.5 承诺填报信息真实：

提供勾选框，用户勾选承诺填报信息真实，并愿意承担相应的法律责任。

2. 健康码生成规则

2.1 生成健康码颜色判断：

根据用户填写的信息进行判断，生成相应颜色的健康码。

2.2 健康码颜色判断规则：

如果满足黄色健康码条件，则生成黄色健康码。

如果满足红色健康码条件，则生成红色健康码。

其余情况生成绿色健康码。

3. 健康码信息展示

3.1 展示健康码：

根据生成的健康码颜色，展示相应颜色的健康码。

3.2 健康码信息包括：

健康码（二维码）信息包括姓名、工号或学号、生成时间等。

4. 页面跳转

4.1 页面跳转规则：

提交申请后，根据生成的健康码颜色，跳转至相应的页面，展示健康码信息。

3.2.2 业务处理流程及要求

1. 业务处理流程

用户访问申请健康码页面，填写个人信息和健康情况。

用户提交申请后，系统根据填写信息生成相应颜色的健康码。

系统展示生成的健康码信息给用户。

根据生成的健康码颜色，系统决定跳转至不同的页面，展示健康码信息。

2. 业务处理要求

2.1 安全性要求：

所有用户提交的个人信息需要进行合法性验证，防止恶意提交和注入攻击。

健康码生成过程中需要确保用户信息的隐私安全，不得泄露用户敏感信息。

2.2 用户体验要求：

申请健康码页面的设计应简洁明了，用户易于理解和操作。

健康码信息展示页面应清晰明了，让用户一目了然地了解自己的健康状态。

2.3 健康码颜色判断要求：

系统应根据用户填写的信息，准确判断健康码的颜色，并生成相应的健康码。

判断条件需要严格按照规定执行，确保生成的健康码准确反映用户的健康状况。

2.4 页面跳转要求：

页面跳转应流畅自然，用户操作后能够清晰地了解到自己的健康码信息。

跳转页面的设计应符合用户习惯，页面内容清晰易懂，用户能够快速获取所需信息。

2.5 用户承诺要求：

用户提交申请时需要认真阅读并勾选承诺填报信息真实的选项，确保所填信息真实有效。

系统应提示用户勾选承诺选项的重要性，并告知相关法律责任。

3.2.3 输入信息

个人健康信息

3.2.4 输出信息

健康码

3.3 开发步骤

编写信息填写页面 healthData→求管理 controller acceptHealthDataServlet→编写相应结果页面 show.jsp

3.4 运行结果

- 登记页面

姓名

飞飞

身份证号

1

工号或学号

1

手机号

12345678910

* 本人近期（14天内）是否去过重点疫区? 单选

是

否

* 本人近期（14天内）是否去过国外? 单选

是

否

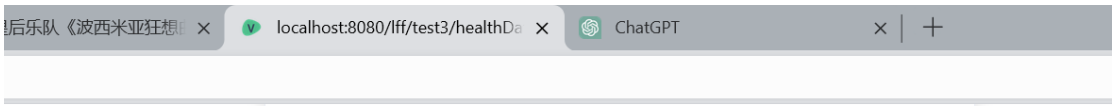
* 您最近14天是否接触过新冠确诊病人或疑似病人? 单选

是

否

* 本人是否被卫生部门确认为新冠肺炎确诊病例或疑似病例? 单选

是



否

* 您最近14天是否接触过新冠确诊病人或疑似病人? 单选

是

否

* 本人是否被卫生部门确认为新冠肺炎确诊病例或疑似病例? 单选

是

否

* 当前健康状况（若有一下状况请在方框内勾选） 多选

- [x] 无异常
- [ ] 发烧(>=37℃)
- [ ] 乏力
- [ ] 干咳
- [ ] 鼻塞
- [ ] 流涕
- [ ] 咽痛
- [ ] 腹泻

本人郑重承诺:

- [x] 为疫情防控，本人同意以上信息依法提交所在辖区疫情防控部门统筹管理。
- [x] 上述信息是我本人填写，本人对信息的真实性和完整性负责。如果信息有误或确实，本人愿意承担相应的法律责任。 同时本人保证遵守疫情管控的各项规定，配合并听从各项措施和要求。

提交

- 绿码

上面时间是会实时变的

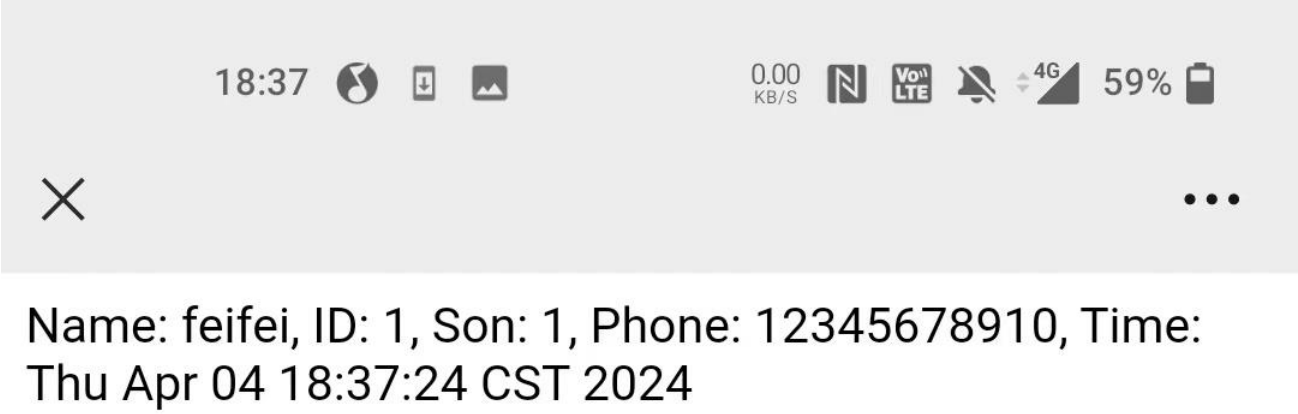
- 红码


哔哩哔哩 (゜- ゜)つロ 干杯~-bilib
【4K】皇后乐队《波
localhost:8080/lff/acceptHealthDataServlet
4月4日
18:35:47
飞飞
红码：。。。

- 黄码


哔哩哔哩 (゜- ゜)つロ 干杯~-bili
【4K】皇后乐队《波西米亚
Chat
localhost:8080/lff/acceptHealthDataServlet
4月4日
18:36:7
飞飞
黄码:。。。。

- 扫码结果


18:37
0.00
KB/S
4G
59%
Name: feifei, ID: 1, Son: 1, Phone: 12345678910, Time: Thu Apr 04 18:37:24 CST 2024

3.5 问题及解决方案

- 图片显示

  转成 base64 显示了

3.6 实验收获

1. **深入理解健康码生成流程：** 通过完成这个项目，我深入了解了健康码生成的流程和规则。了解了用户填写信息后，系统如何根据规则生成相应颜色的健康码，这对于理解疫情防控的重要性有很大帮助。
2. **加强了 Java Web 开发技能：** 在实现申请健康码功能的过程中，我加强了对 Java Web 开发技术的掌握，包括 JSP 页面的设计、Servlet 的编写和请求处理等方面。这让我对 Web 开发有了更深入的了解。
3. **学会了处理用户提交的表单数据：** 通过编写申请健康码页面，我学会了如何在 Java Web 应用中处理用户提交的表单数据，包括获取和验证用户输入，确保数据的合法性和完整性。
4. **提高了系统设计能力：** 在项目中，我需要设计一个合理的健康码生成流程，考虑到用户的填写习惯、信息的完整性和安全性等因素。这让我提高了系统设计的能力。

3.7 源代码

- 引入生成二维码依赖

```
<!--        二维码-->
        <dependency>
            <groupId>com.google.zxing</groupId>
            <artifactId>core</artifactId>
            <version>3.4.1</version> <!-- 或者是最新版本 -->
        </dependency>
        <dependency>
            <groupId>com.google.zxing</groupId>
            <artifactId>javase</artifactId>
            <version>3.4.1</version> <!-- 或者是最新版本 -->
        </dependency>
```

- acceptHealthDataServlet

```java
package bank.com.homework4.controller;
/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 4:13 2024/4/4
 * @ Description: ${description}
 */

import bank.com.homework4.model.Customer;
import bank.com.homework4.pojo.User;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;
import java.util.Date;

@WebServlet(name = "acceptHealthDataServlet", value =
"/acceptHealthDataServlet")
public class acceptHealthDataServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }

    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        //获取参数
        String name = request.getParameter("name");
        String id = request.getParameter("id");
        String son = request.getParameter("son");
        String phone = request.getParameter("phone");
        String radio1 = request.getParameter("radio1");
        String radio2 = request.getParameter("radio2");
        String radio3 = request.getParameter("radio3");
        String radio4 = request.getParameter("radio4");
        String checked2 = request.getParameter("checked2");
        String checked3 = request.getParameter("checked3");
        String checked4 = request.getParameter("checked4");
        String checked5 = request.getParameter("checked5");
        String checked6 = request.getParameter("checked6");
        String checked7 = request.getParameter("checked7");
        String checked10 = request.getParameter("checked10");
```

```
        //设置二维码颜色
        String color;
        while (true){
            //直接红码
            if(radio3.equals("1")||radio4.equals("1")){
                color="red";
                break;
            }
            int t=0;
            //条件红码
            if(checked2!=null&&checked2.equals("true"))t++;
            if(checked3!=null&&checked3.equals("true"))t++;
            if(checked4!=null&&checked4.equals("true"))t++;
            if(checked5!=null&&checked5.equals("true"))t++;
            if(checked6!=null&&checked6.equals("true"))t++;
            if(checked7!=null&&checked7.equals("true"))t++;
            if(checked10!=null&&checked10.equals("true"))t++;
            if(t>=2){
                color="red";
                break;
            }
            //黄码
            if(t!=0||radio1.equals("1")||radio2.equals("1"))
                color="yellow";
            else color="green";
            break;
        }

        //创建 bean
        User user=(User) getServletContext().getAttribute("User");
        if(user==null){
            user=new User(name,id,son,phone,color,new Date());
        }

        request.getSession().setAttribute("user",user);


request.getRequestDispatcher("./test3/show.jsp").forward(request,resp
onse);
    }
}
```

- show.jsp 展示二维码

```
<%@ page import="com.google.zxing.client.j2se.MatrixToImageConfig" %>
<%@ page import="java.io.ByteArrayOutputStream" %>
<%@ page import="com.google.zxing.common.BitMatrix" %>
<%@ page import="com.google.zxing.client.j2se.MatrixToImageWriter" %>
<%@ page import="com.google.zxing.MultiFormatWriter" %>
<%@ page import="com.google.zxing.BarcodeFormat" %>
<%@ page import="java.io.OutputStream" %>
<%@ page import="java.io.File" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="java.util.Base64" %><%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/4/4
  Time: 16:05
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>实验三展示</title>
</head>
<body>
<jsp:useBean id="user" scope="session"
class="bank.com.homework4.pojo.User"></jsp:useBean>
<%-- 使用 JavaScript 定时器实时更新当前时间 --%>
<h1 id="currentTime"></h1>
<hr>
<h5><jsp:getProperty name="user" property="name"/></h5>
<%-- 生成二维码 --%>

<%
    String qrCodeData = "Name: " + user.getName() + ", ID: " + user.getId()
+
            ", Son: " + user.getSon() + ", Phone: " + user.getPhone() +
            ", Time: " + user.getDate();
    int width = 300;
    int height = 300;
    String imageFormat = "png";

    // 设置二维码颜色
    int blackColor; // 黑色
    int whiteColor = 0xFFFFFFFF; // 白色
    if (user.getColor().equals("red"))
        blackColor = 0xFFFF0000; // 红色
    else if (user.getColor().equals("yellow"))
```

```
        blackColor = 0xFFFFFF00; // 黄色
    else blackColor = 0xFF00FF00; // 绿色

    // 创建颜色配置
    MatrixToImageConfig colorConfig = new
MatrixToImageConfig(blackColor, whiteColor);

    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    try {
        BitMatrix bitMatrix = new MultiFormatWriter().encode(qrCodeData,
BarcodeFormat.QR_CODE, width, height);
        MatrixToImageWriter.writeToStream(bitMatrix, imageFormat,
outputStream, colorConfig); // 使用颜色配置
    } catch (Exception e) {
        e.printStackTrace();
    }

    // 将二维码数据编码为 Base64 字符串
    String base64ImageData =
Base64.getEncoder().encodeToString(outputStream.toByteArray());
    // 构建 Data URI
    String dataUri = "" + base64ImageData;
%>
    <%-- 显示二维码图像 --%>
    <img src="<%= dataUri %>" alt="QR Code">

    <%-- 获取用户对象的颜色属性 --%>
    <% String color = user.getColor(); %>
    <%-- 根据用户颜色进行条件判断 --%>
    <%
        if ("red".equals(color)) {
    %>
    <h1>红码：。。。</h1>
    <%
    } else if ("green".equals(color)) {
    %>
    <h1>绿码：凭此码可。。。</h1>
    <%
    } else if ("yellow".equals(color)) {
    %>
    <h1>黄码:。。。。</h1>
    <%
    } else {
    %>
```

```
    <h1>其他颜色</h1>
    <%
        }
    %>


<script>
    function updateTime() {
        var currentTimeElement =
document.getElementById("currentTime");
        var currentTime = new Date();
        var formattedTime = currentTime.getMonth() + 1 + "月" +
currentTime.getDate() + "日 <br>" +
            currentTime.getHours() + ":" + currentTime.getMinutes() +
":" + currentTime.getSeconds();

        currentTimeElement.innerHTML = formattedTime;
    }
    // 每秒更新一次时间
    setInterval(updateTime, 1000);
</script>


</body>
</html>
```

- healthData.html 用户输入

这个文件用了 vue.js

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <style>
    .myTable{
      border: 2px solid #0000FF;
      margin-top: 15%;
    }
    .inputs{
      width: 95%;
      height: 25px;
      color: #999;
    }
    .btn{
```

```
      background-color:#4a8bd6;
      width: 90%;
      height: 40px;
      border: none;
      color: aliceblue;
      text-decoration: underline; /* 添加白色下划线 */
      margin: 10px;
    }
    .box-card{
      margin-top: 1%;
      width: 25%;
      margin-left: 37.5%;
    }
    .headImg{
      margin:0%;
      background-repeat: no-repeat;
      background-size: cover;
    }
    .btn{
      width: 100%;
      height: 100%;
    }
    .borders{
      width: 100%;
      color: black; font-weight: bold;
      height: 100%;
      /* border-top: 10px solid lightgrey; */
    }
    .flag{
      width: 65px;
      height: 35px;
      cursor:default;
    }
  </style>
</head>

<body>
<div id="app">
  <el-card class="box-card" style="width: 30%;">

    <!-- 内容 -->
    <el-form :model="healthInformation" labelPosition="top" id="from"
act>
      <!-- 信息 -->
```

```
      <!-- 姓名 -->
      <el-form-item label="姓名">
        <el-input v-model="name" placeholder="请输入姓名"></el-input>
      </el-form-item>
      <!-- 身份证号 -->
      <el-form-item label="身份证号">
        <el-input v-model="id" placeholder="请输入身份证号
"></el-input>
      </el-form-item>
      <!-- 工号或学号 -->
      <el-form-item label="工号或学号">
        <el-input v-model="son" placeholder="请输入工号或学号
"></el-input>
      </el-form-item>
      <!-- 手机号 -->
      <el-form-item label="手机号">
        <el-input v-model="phone" placeholder="请输入手机号
"></el-input>
      </el-form-item>
      <!-- 重点疫区 -->
      <el-form-item label="本人近期（14 天内）是否去过重点疫区？"
required class="borders" style="color: black; font-weight: bold;">
        <el-button type="primary" plain round class="flag" disabled
style="cursor: default">
          <font style="font-size: 12px; text-align: center; height:
100%;">单选</font>
        </el-button>
        <br>
        <el-radio-group v-model="radio1" style="width: 100%;">
          <el-radio :label="1">是</el-radio>
          <el-divider ></el-divider>
          <br>
          <el-radio :label="0">否</el-radio>
        </el-radio-group>
      </el-form-item>
      <!-- 国外 -->
      <el-form-item label="本人近期（14 天内）是否去过国外？" required
class="borders" style="color: black; font-weight: bold;">
        <el-button type="primary" plain round class="flag" disabled
style="cursor: default">
          <font style="font-size: 12px; text-align: center; height:
100%;">单选</font>
        </el-button>
        <br>
```

```
        <el-radio-group v-model="radio2" style="width: 100%;">
          <el-radio :label="1">是</el-radio>
          <el-divider ></el-divider>
          <br>
          <el-radio :label="0">否</el-radio>
        </el-radio-group>
      </el-form-item>
      <!-- 接触 -->
      <el-form-item label="您最近 14 天是否接触过新冠确诊病人或疑似病
人?" required class="borders">
        <el-button type="primary" plain round class="flag" disabled
style="cursor: default">
          <font style="font-size: 12px; text-align: center; height:
100%;">单选</font>
        </el-button>
        <br>
        <el-radio-group v-model="radio3" style="width: 100%;">
          <el-radio :label="1">是</el-radio>
          <el-divider ></el-divider>
          <br>
          <el-radio :label="0">否</el-radio>
        </el-radio-group>
      </el-form-item>
      <!-- 确诊 -->
      <el-form-item label="本人是否被卫生部门确认为新冠肺炎确诊病例或
疑似病例？" required class="borders">
        <el-button type="primary" plain round class="flag" disabled
style="cursor: default">
          <font style="font-size: 12px; text-align: center; height:
100%;">单选</font>
        </el-button>
        <br>
        <el-radio-group v-model="radio4" style="width: 100%;">
          <el-radio :label="1">是</el-radio>
          <el-divider ></el-divider>
          <br>
          <el-radio :label="0">否</el-radio>
        </el-radio-group>
      </el-form-item>
      <!-- 症状 -->
      <el-form-item label="当前健康状况 （ 若有一下状况请在方框内勾
选 )" required class="borders">
        <el-button type="warning" plain round class="flag" disabled
style="cursor: default">
```

```
          <font style="font-size: 12px; text-align: center; height:
100%;">多选</font>
        </el-button>
        <br>
        <el-checkbox v-model="checked1" style="border-bottom: 1px solid
lightgray; width: 100%;">无异常</el-checkbox>
        <br>
        <el-checkbox v-model="checked10" style="border-bottom: 1px
solid lightgray; width: 100%;">发烧(>=37℃)</el-checkbox><br>
        <el-checkbox v-model="checked2" style="border-bottom: 1px solid
lightgray; width: 100%;">乏力</el-checkbox><br>
        <el-checkbox v-model="checked3" style="border-bottom: 1px solid
lightgray; width: 100%;">干咳</el-checkbox><br>
        <el-checkbox v-model="checked4" style="border-bottom: 1px solid
lightgray; width: 100%;">鼻塞</el-checkbox><br>
        <el-checkbox v-model="checked5" style="border-bottom: 1px solid
lightgray; width: 100%;">流涕</el-checkbox><br>
        <el-checkbox v-model="checked6" style="border-bottom: 1px solid
lightgray; width: 100%;">咽痛</el-checkbox><br>
        <el-checkbox v-model="checked7">腹泻</el-checkbox><br>
      </el-form-item>
      <!-- 承诺 -->
      <el-form-item label="本人郑重承诺：" class="borders">
        <br>
        <el-checkbox v-model="checked8">为疫情防控，本人同意以上信息依
法提交所在辖区疫情防控部门统筹<br>管理。</el-checkbox><br>
        <el-checkbox v-model="checked9">上述信息是我本人填写，本人对信
息的真实性和完整性负责。如果信<br>息有误或确实，本人愿意承担相应的法律
责任。
          同时本人保证遵守疫<br>情管控的各项规定，配合并听从各项措施和
要求。
        </el-checkbox><br>

      </el-form-item>
      <!-- 确认 -->
      <el-button type="primary" class="btn" class="borders"
@click="custCheck">提交</el-button>
    </el-form>
    <!-- 隐藏的表单 -->
    <form ref="hiddenForm" action="../acceptHealthDataServlet"
method="post" style="display: none;">
      <input type="text" name="name" :value="name">
      <input type="text" name="id" :value="id">
      <input type="text" name="son" :value="son">
```

```
      <input type="text" name="phone" :value="phone">
      <input type="text" name="radio1" :value="radio1">
      <input type="text" name="radio2" :value="radio2">
      <input type="text" name="radio3" :value="radio3">
      <input type="text" name="radio4" :value="radio4">
      <input type="text" name="checked1" :value="checked1">
      <input type="text" name="checked2" :value="checked2">
      <input type="text" name="checked3" :value="checked3">
      <input type="text" name="checked4" :value="checked4">
      <input type="text" name="checked5" :value="checked5">
      <input type="text" name="checked6" :value="checked6">
      <input type="text" name="checked7" :value="checked7">
      <input type="text" name="checked8" :value="checked8">
      <input type="text" name="checked9" :value="checked9">
      <input type="text" name="checked10" :value="checked10">
    </form>


  </el-card>
</div>
</body>
<!-- import Vue before Element -->
<script
src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>
<!-- 引入样式 -->
<link rel="stylesheet"
href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">
<!-- 引入组件库 -->
<script src="https://unpkg.com/element-ui/lib/index.js"></script>
<script>
  new Vue({
    el: '#app',
    data: function () {
      return {
        name:'',
        id:'',
        son:'',
        phone:'',
        radio1:'',
        radio2:'',
        radio3:'',
        radio4:'',
        checked1:'',
        checked2:'',
```

```
        checked3:'',
        checked4:'',
        checked5:'',
        checked6:'',
        checked7:'',
        checked8:'',
        checked9:'',
        checked10:'',
      }
    },
    methods: {
      custCheck() {
        if(!this.name||!this.id||!this.son||!this.phone){
          alert("请补全个人信息!");
        }else if
(this.radio1===''||this.radio2===''||this.radio3===''||this.radio4===
'') {
          alert("请完整填写信息!");
          return false;
        } else if (!this.checked8 || !this.checked9) {
          alert("请确认承诺!");
          return false;
        } else {
          this.$refs.hiddenForm.submit();
        }
      }
    }
  })
</script>

</html>
```