个人课程设计报告

团队编号：第十组

报告人员：周子渔

完成日期：2023/7/10

# 1．个人负责任务

## 1.1个人负责的活动

数据库的建立与实现

数据库类的实现

登录界面的实现

注册界面的实现

项目开发计划文档的撰写

## 1.2活动具体过程及步骤

**数据库部分：**

**需求分析：**在开始数据库设计之前，我对业务需求进行分析，了解存储哪些数据，以及这些数据之间的关系。根据给出的代码，我们的业务需求包括：用户注册和登录、用户信息管理、图书信息管理、订单管理、类别管理、出版社管理等。**设计数据表：**需求分析完成后，我设计了数据表。对于每个业务需求，我创建了一个或多个数据表。例如，我创建了一个users表来存储用户信息，一个books表来存储图书信息，一个orders表来存储订单信息，等等。在这个过程中，我还**决定了每个表的字段及其数据类型**。例如，users表可能需要包括Id、LoginId、LoginPwd、Name、Address、Phone、Mail、UserRoleId和UserStateId等字段。其中，Id是用户的唯一标识，LoginId和LoginPwd用于用户登录，Name、Address、Phone和Mail用于存储用户的个人信息，UserRoleId和UserStateId用于记录用户的角色和状态。**创建数据表：**设计好数据表后，我使用SQL语句创建这些表。例如，我使用CREATE TABLE语句创建users表，并使用AUTO\_INCREMENT、NOT NULL等约束来保证数据的完整性。**设置主外键：**在所有数据表创建完毕后，我设置了各表之间的关联。这一步需要使用到FOREIGN KEY和REFERENCES等SQL关键字，如users表中的UserRoleId字段是userroles表中的Id字段的外键。**插入数据：**创建并关联好数据表后，我开始插入一些初始数据。例如，我插入了一些测试用户、测试图书和测试订单。**测试：**最后，需要对数据库进行测试，以确保所有功能都正常工作。我尝试了包括添加、删除、修改和查询等操作的测试。

**登陆界面部分：**

**需求分析:**我认为，在开始任何代码编写之前，首先需要对应用的需求进行全面的分析。这个网页的主要功能是允许用户登录到一个在线书店系统。需要有登录表单，一个连接到注册页面的链接，以及一些附加功能，比如记住密码和忘记密码的选项。**页面布局设计:** 这个网页的布局是通过HTML和CSS来完成的。首先，我设计了一个包含图片和登录表单的页面布局。使用Bootstrap的栅格系统（class="col-xs-7" 和 class="col-xs-4"）来创建响应式布局。**样式设计:**为网页添加样式，包括内联样式和外部样式表（style.css 和 font-awesome.min.css）。对#container .imgs和.login\_input等元素进行样式定制，如大小、颜色等。**编写HTML和JSP代码:**在设计完成后，开始编写HTML和JSP代码来实现网页的基本结构。这包括头部(header.jsp)和尾部(footer.jsp)的导入，登录表单的创建，以及“记住密码”和“忘记密码”的功能实现。**创建表单提交功能:**在form元素中，action属性被设置为"UserServlet?opt=login"，这意味着当用户提交表单后，数据将被发送到UserServlet处理，opt=login是一个参数，它告诉Servlet这是一个登录操作。**客户端验证:**使用JavaScript函数check\_Login\_Submit()在表单提交之前进行客户端验证。如果验证失败，将返回false并取消提交。如果验证成功，将返回true并继续提交。在用户名和密码输入框失去焦点时，使用onblur事件触发check\_Login\_Id()和check\_Login\_Pwd()函数进行单个输入的验证。**添加动态元素:**使用JSP来动态地创建页面元素。例如，${ctxPath}是一个动态值，它代表应用的上下文路径。这在加载CSS样式表和JavaScript文件时非常有用。**测试:**在完成编码后，进行全面的测试，确保所有的功能都能正常工作，包括输入验证、表单提交等。同时要在不同的设备和浏览器上进行测试，确保响应式设计的正确性。**部署:**在完成所有测试后，将网页部署到服务器上，供用户访问。

**注册界面部分：**

**需求分析:** 这个页面的需求是让用户进行注册。根据需求，我收集了用户名、密码、确认密码、真实姓名、电子邮箱、手机号码和联系地址等信息。页面需要包括表单验证和表单提交到后端的功能。**设计页面布局:** 使用HTML和CSS来设计页面布局。这个页面包含一个注册表单，以及其他一些用户提示和链接到登录页面的功能。编写HTML和JSP代码: 在设计完成后，开始编写HTML和JSP代码来实现页面的基本结构。这包括头部(header.jsp)和尾部(footer.jsp)的导入，以及注册表单的创建。**创建表单提交功能:**在form元素中，action属性被设置为"${ctxPath }/UserServlet?opt=register"，这意味着当用户提交表单后，数据将被发送到UserServlet处理，opt=register是一个参数，它告诉Servlet这是一个注册操作。**表单验证:** 使用JavaScript函数check\_Rg\_Submit()在表单提交之前进行客户端验证。如果验证失败，将返回false并取消提交。如果验证成功，将返回true并继续提交。**处理服务器返回的信息:** 在注册表单中，有一个id为"msg"的div元素用于显示服务器返回的注册结果消息(${MSG\_USER\_REGISTER\_RESULT})。在页面加载时，使用JavaScript代码来处理这个消息：如果消息为空，就隐藏这个div元素；否则，就显示这个元素。**添加动态元素:** 使用JSP来动态地创建页面元素。例如，${ctxPath}是一个动态值，它代表应用的上下文路径。这在加载JavaScript文件时非常有用。另外，${MSG\_USER\_REGISTER\_RESULT}是从后端传来的注册结果消息。**测试:** 在完成编码后，进行全面的测试，确保所有的功能都能正常工作，包括输入验证、表单提交等。同时要在不同的设备和浏览器上进行测试，以确保页面的兼容性和响应式设计的正确性。**部署:** 在完成所有测试后，将页面部署到服务器上，供用户访问。

# 2．研发过程中的困难与收获

**困难：**

**兼容性问题：**在前端开发中，不同浏览器对于HTML、CSS和JavaScript的支持程度有所不同。在开发时需要确保代码在所有主流浏览器中都能正常工作，这无疑增加了开发的复杂性。**UI设计：**用户界面设计是一个具有挑战性的过程，需要平衡美观与易用性。在这个过程中，我尝试许多不同的设计并进行了大量的用户测试，以保证我们的页面既好看又好用。

**收获：**

**提升了我们的前端技术水平：**在开发过程中，我熟悉了HTML、CSS和JavaScript的使用，提升了前端技术水平。**对用户体验有了更深的理解：**在进行UI设计和用户测试过程中，我对用户体验有了更深的理解，这将对我未来的开发工作有很大的帮助。**提高了团队合作能力：**在开发过程中，我学会了如何在团队中协作，有效地分配工作并进行有效的沟通。