|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | JSP应用程序设计 | | | | | |
| 实验名称 | 综合实验 | | | | | |
| 实验时间 | 2023年5月25日 | | 实验地点 | 信息661-3 | | |
| 学 院 | 信息科学与工程学院 | | 专 业 | 计算机科学与技术 | | |
| 姓 名 | 方家琪 | 班 级 | 计科2103 | 学 号 | | 202131222118 |
| 同组实验者  姓名 | 无 | | | | | |
| 实验成绩 |  | | 指导教师  （签字） | 乔善平 | | |
| 指标点 | | | | | 成绩 | |
| 毕业要求指标点3-2（40%）  掌握JSP技术的基础知识以及网络程序设计的基本思想方法，能够运用所学知识与方法，分析和理解基于JSP的网络软件系统开发的工作原理，并能够针对复杂网络软件系统问题需求选取恰当的开发工具和技术、凝练并表达设计流程，形成设计方案并研究其可行性。 | | | | |  | |
| 毕业要求指标点4-3（60%）  能够运用JSP运行环境搭建、Web项目开发、调试和部署等知识，分析网络软件系统设计、开发和实施过程中的关键影响因素，并能够针对复杂网络软件系统问题搭建恰当的实验平台实施软件实验并验证。 | | | | |  | |
| 1. **角色与功能**（有哪几种角色、每种角色有哪些功能）   系统涉及两种角色，分别是普通用户和管理员。  （1）普通用户  普通用户可以登录、注册、退出系统、注销账号、更改个人信息（除了用户ID）、查询电影信息（包括电影名称、导演、发行商、售价）、购买电影、查看数据库中所有电影的部分信息（包括电影名称、导演、发行商、售价）。  （2）管理员  管理员可以登录、退出系统、更改个人信息（除了用户ID）、查询电影信息、更改电影信息、插入电影信息、删除电影信息、查看当前已注册用户的信息、查看数据库中所有电影的全部信息（包括电影名称、导演、发行商、原价、售价、余票）。  **二、数据库设计**（截图给出表的结构与表的内容）  数据库有两个表，分别为movie和user。  （1）movie表  movie表的各属性分别为电影ID、电影名称、导演、发行商、原价、售价、余票，主码为电影ID。    图1 movie表结构    图2 movie表内容  （2）user表  user表的各属性包括用户ID、用户名、密码、权限、手机号码、邮箱地址，主码用户ID。    图3 user表结构    图4 user表内容  **三、项目文件结构**（截图列出项目中的jsp、JavaBean、Servlet、其他文件）    图5 文件目录结构   |  |  | | --- | --- | |  | **JSP** | | 文件名 | 简介 | | movielist.jsp | 电影列表，显示部分电影信息（电影名称、导演、发行商、售价） | | movielistadmin.jsp | 电影列表（管理员），并显示电影全部信息 | | index.jsp | 系统主页 | | indexadmin.jsp | 系统主界面（管理员） | | login.jsp | 登录页面 | | profile.jsp | 个人中心页面 | | register.jsp | 注册页面 | | search.jsp | 电影搜索，通过电影名检索，展示部分电影信息 | | searchadmin.jsp | 电影搜索（管理员），通过电影名检索，展示全部电影信息，并可以完成编辑操作 | | movie.jsp | 电影购票表格显示 | | movieadmin.jsp | 电影购票（管理员）表格显示 | | regulateadmin.jsp | 用户管理页面 | |  | **DBM** | | 文件名 | 简介 | | DBM.java | 数据库工具类 | |  | **Bean** | | 文件名 | 简介 | | Movie.java | 电影类 | | User.java | 用户类 | |  | **Servlet** | | 文件名 | 简介 | | Code.java | 生成五位验证码实现 | | DeleteAccount.java | 用户注销账号实现 | | DeleteAccountAdmin.java | 管理员注销用户账号实现 | | DeleteMovie.java | 删除电影信息实现 | | GetAllMovies.java | 获取所有电影的全部信息实现 | | GetUserInfo.java | 获取所有用户的全部信息实现 | | InsertMovie.java | 插入电影信息实现 | | Check.java | 登录身份验证实现 | | PurchaseSearch.java | 购买搜索的电影操作实现 | | RegisterCheck.java | 注册验证实现 | | Search.java | 查询电影信息实现 | | UpdateMovie.java | 更新电影信息实现 | | UpdatePassword.java | 更改用户密码实现 | | UpdateUserInfo.java | 更改用户个人信息实现 | | GetAllUsers.java | 获取所有用户的全部信息实现 | | AllMovies.java | 展现所有电影信息实现 | | Buy.java | 购买电影实现 |   **四、运行效果截图**（尽可能充分地通过截图展示系统的运行情况）  **1、普通用户**    图6 登录界面  点击验证码可切换，验证码不正确时给出提示。    图7 验证码错误提示  用户名或密码不正确时给出提示。    图8 用户名或密码不正确  普通用户没用账号，可以注册。    图9 注册页面  用户名已存在时，给出提示。    图10 用户存在提示  成功注册后，给出成功注册提示。    图11 成功注册提示  登录或者注册成功后进入系统主页面。    图12 系统主页面  用户可以点击购买按钮购买对应的电影票，购买成功后，会给出成功购买提示（如图13）。    图13 购买成功提示  用户在右侧购票查询中输入待查询的电影名称，可以搜索到对应的电影信息（如图14），用户同样可以在查询结果中进行购买操作（如图15）。    图14 查询结果    图15 购买成功提示  当所购买的电影票剩余为零时，购买会提示票已售罄（如图16）    图16 票已售罄提示  点击导航栏上的“电影列表”按钮，跳转到电影列表页面，该页面展示数据库中电影的序号、电影名成、导演、发行商、售价等信息以及电影售票统计扇叶图（如图17）。    图17 电影列表页面  鼠标悬浮在右上角的“欢迎你 xxx”时，自动弹出“个人中心”、“退出登录”和“注销账号”（如图18），点击“个人中心”即可进入个人中心页面（如图19）。    图18 用户中心    图19 个人中心  用户可以在个人信息和修改密码中进行操作。  （1）个人信息的修改  用户可以更改用户名称、手机号码以及电子邮箱，但不可更改用户ID。  更改完后，点击保存，给出修改成功提示（图20）。    图20 修改成功提示  （2）修改密码  用户修改密码时，需要按照要求填写信息，当旧密码填写错误时，弹出提示按钮（如图21），两次密码不正确时（如图22），修改成功提示（如图23）。    图21 旧密码填写错误提示    图22 密码不一致提示    图23 成功更改提示  用户需要退出系统时，点击“退出登录”按钮，即可完成退出系统的操作（如图24）。    图24 用户中心  **2、管理员**  管理员登录系统后进入系统主页面（如图25）。    图25 系统主页面  管理员可以查询电影信息（如图26），在查询结果中可以修改（如图27）或删除（如图28）电影信息，并给出相应提示。    图26 查询结果    图27 修改成功提示    图28 删除成功提示  点击导航栏中的电影售票，会跳转到售票管理界面（如图29），管理员可以在此界面对电影信息进行插入操作，插入成功会给出提示（如图30），插入要求电影名称不可为空，否则给出提示（如图31）。    图29 电影售票界面    图30 成功插入提示    图31 插入失败提示  点击导航栏中的“用户管理”按钮，会跳转到用户管理界面（如图32），管理员可以对用户进行注销操作。    图32 用户管理界面  鼠标悬浮在右上角的“欢迎你 xxx”时，自动弹出“个人中心”和“退出登录”如图33），点击“个人中心”即可进入个人中心页面（如图34）。    图33 管理员中心    图34 个人中心  （1）个人信息的修改  管理员可以更改管理员名称、手机号码以及电子邮箱，但不可更改管理员ID。  更改完后，点击保存，给出修改成功提示（图35）。    图35 修改成功提示  （2）修改密码  管理员修改密码时，需要按照要求填写信息，当旧密码填写错误时，弹出提示按钮（如图36），两次密码不正确时（如图37），修改成功提示（如图38）。    图36 旧密码填写错误提示    图37 密码不一致提示    图38 成功更改提示  管理员需要退出系统时，点击“退出登录”按钮，即可完成退出系统的操作（如图39）。    图39 管理员中心  **五、实验分析及反馈**  1、主要问题及其解决方法  （1）中文乱码  问题：起初页面中文会显示乱码，经排查发现需要更改request的编码格式。  解决办法：在页面中添加request.setCharacterEncoding("utf-8")，即可解决问题。  （2）SQL语句报错  问题：在查询电影余票的时候，后台不断报错：java.sql.SQLException: Illegal operation on empty result set. 经排查发现原来是执行SQL查询并没有返回任何数据所导致的。  解决办法：购买电影票时用过a标签传递正确的参数，确保SQL语句可以查询到结果。修改后的代码如下：  <td style="text-align: center;"><a class="btn btn-success btn-xs" type="button" href="Buy?name=${movie.name}">购买</a> </td>  String name = request.getParameter("name");  String sql1 = "SELECT STOCK FROM movie WHERE NAME = '" + name + "'";  2、目标达成情况说明（结合毕业要求指标点的内容和实际实验情况进行描述）  （1）登录功能  当用户在登录页面输入用户名和密码并点击登录按钮后，系统会将用户输入的信息提交给Check.java进行验证。在验证流程中，系统会首先对用户输入的验证码进行比对，如果验证码正确，则系统会在数据库中查询用户名是否存在，如果存在，则系统会将输入的密码与数据库中该用户名对应的密码进行比对，如果密码也正确，则用户会被授权进入系统。同时，在确认用户身份后，如果用户权限为管理员，则系统还会将其授权为管理员身份登录系统。  （2）注册功能  当用户点击注册按钮时，系统会将用户输入的信息提交给RegisterCheck.java进行验证。在验证流程中，系统会首先对用户输入的验证码进行比对，如果验证码正确，则系统会在数据库中查询该用户名是否已存在，如果不存在，则系统会检查两次输入的密码是否一致。如果两次密码输入正确，则系统将认为用户注册成功，并授权其以普通身份登录系统。如果用户名已存在或两次密码输入不一致，则系统将拒绝用户的注册请求。  （3）验证码实现  为了增强认证的安全性，可以采用随机生成数字和字母的方式生成验证码，并将验证码存储在session中。在用户输入用户名和密码之后，服务器会从session中获取之前生成的验证码并进行比对，以确保该请求是合法的并不是自动化脚本发送的。  （4）个人主页实现  从数据库中获取用户的个人信息可以使用GetUserInfo.java方法，在获取到用户信息后将其存储在session中。在profile.jsp页面中，可以使用EL表达式来获取存储在session中的用户个人信息，并进行显示。  用户提交更新后的个人信息时，可以使用UpdateUserInfo.java方法来更新数据库中对应的记录，但在更新之前，需要在数据库中查询出除了原本的用户名以外是否还存在其他用户名和新用户名一致的记录。  用户提交修改密码的请求时，可以使用UpdatePassword.java来更新数据库中的对应记录。但在更新前，需要对用户输入的新旧密码进行检查，如新旧密码是否相同以及旧密码是否正确。  如果用户需要注销账户，则可以使用DeleteAccount.java方法来删除该用户的信息记录。删除后，将用户从session中注销，并在注销后跳转到登录界面。  （5）插入电影信息实现  使用InsertMovies.java可以将用户提交的电影信息插入到数据库中。该类中需要先获取表单中用户提交的信息，然后将这些信息封装成为一个电影对象，最后将该对象插入到数据库中。  （6）删除电影信息实现  使用DeleteMovie.java可以删除指定的电影信息。在实现该方法时，可以在数据库中查询出要删除的电影记录，然后执行删除操作。同时，对于删除操作，需要保证只有管理员才有权限进行该操作。  （7）修改电影信息实现  使用UpdateMovie.java可以更新指定的电影信息。在实现该方法时，可以在数据库中查询出指定的电影记录，然后执行更新操作。需要对用户提交的更新信息进行合法性验证，例如对电影名称、导演、演员、剧情介绍和海报等字段进行非空验证。  （8）查询电影信息实现  使用Search.java可以根据用户输入的关键字在数据库中查询符合条件的电影信息。在实现该方法时，需要调用数据库API来进行查询操作，并将查询结果存储在request中，以便JSP页面进行显示。  （9）用户管理功能实现  使用DeleteAccountAdmin.java进行删除用户信息，此操作只有管理员才能调用，实现细节与DeleteAccount.java类似。  3、意见和建议  可以与本学期的Web作业进行一个结合。 | | | | | | |