院 系 计算机学院

学 号 　　18340224 姓 名 钟赫明

学 号 　　18340222  姓 名 郑立锋

学 号 　　18340226 姓 名 周伯霖

学 号 　　18340217 姓 名 张翊健

学 号 　　18340216 姓 名 张烨禧

专业工程项目实践

# 大作业实验报告

【实验目的】

通过建立一个网站来综合应用本课程学习的知识。

【大作业分工】

组员分工可以按照模块（JSP程序）分工，也可以按照软件开发过程分工，例如：系统设计、前台(html+CSS)、后台(JSP)、数据库管理、测试、安装、写文档。

【大作业评分标准】

（1）有无数据库增改删操作

（2）有无文件上传功能

（3）核心功能是否实现以及功能是否完整性

（4）项目的整体难度

（5）画面设计的美观

（6）实验报告编写

（7）提交时间

同一个组所有成员统一打分。所有2~5人的组，打分时不看人数，同等对待。对于单人1组的要有充足的理由（在实验体会中进行说明），否则要扣分。

【实验环境】

tomcat，JDK和mysql。注意：要允许外部访问应用程序需要[设置Windows防火墙](http://www.xitongzhijia.net/xtjc/20150226/39028.html)。

【实验要求】

（1）最好有数据库增改删操作，上传文件操作。例如：博客上传照片，增加修改删除博客。

（2）最好有管理员后台的操作功能，例如，浏览所有用户的博客内容，删除用户非法博客，某用户禁言。

（3）前后台都直接采用底层编程，不要使用任何框架。

（4）确定需求后，画出用例图（顶层）、类图、E-R图，然后再编码。采用快速迭代的方法，先做一个小型完整的系统，有时间再不断增加功能或改进设计。

（5）整个项目完成后要在教学服务器上安装，安装说明由另一个文件给出。

（6）不要做用户增改删功能，用户数据直接通过数据库管理软件输入到数据库中。

（7）如果需要用户，可以做用户登录功能。预先增加好测试用的用户，使TA或其他同学可以直接登录进入系统，普通用户名采用user，管理用户名（如果需要）采用admin，密码都是123。

【安全防护】

要注意以下安全性问题：

（1）MySQL注入：

如果某个网站的登录验证的SQL查询代码为：

strSQL="SELECT \* FROM users WHERE (name = '" + userName + "') and (pw = '"+ passWord +"');"

则可以填入

用户名:1' OR '1'='1

密码:1' OR '1'='1

进行登录时，SQL语句变为：

SELECT \* FROM users WHERE (name = '1' or '1' = '1') and (pw = '1' or '1' = '1');

（2）脚本攻击

在博客中输入<script>window.location.href="http://www.abc.com"</script>

会导致显示该网页时转至http://www.abc.com。

**[下面各项为要填写的内容]**

【组长】

学号：18340224 姓名：钟赫明 手机号：13117983818 QQ：3216769194

【小组号】

18340216

【项目名称】

MDBlog

【项目简称】

Markdown博客系统

【项目描述】

本项目是一个简单的基于Markdown文件的博客系统，支持在线的markdown格式博客编辑，以及将存储在博客系统上的markdown格式博客展示出来；

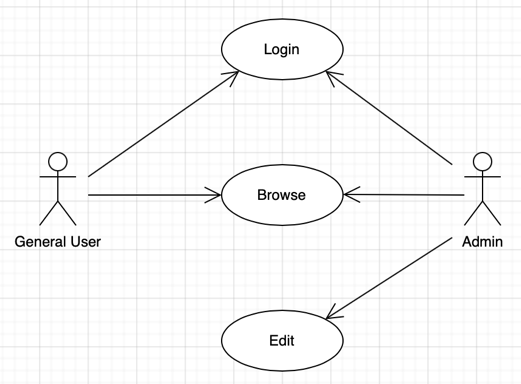
【系统设计】

1、顶层用例图

根据要求，在开始实现之前我们需要先确定好项目的顶层用例。通过查阅相关资料我们得知，项目开发中的用例图通常是用来描述项目中用户的需求的，即从用户的角度描述系统的功能，并指出各功能的执行者。在具体实现上，用例图应该要强调谁在使用系统，以及系统为执行者完成了哪些功能。

所以根据我们的讨论的结果，我们可以大概确定该系统的功能及满足的需求：

1. 首先是浏览(Browse)功能。我们要确保该博客系统能够像其他的博客网站一样，允许博客的拥有者以及其他获得访问许可的用户来浏览记录在该系统上的博客内容；
2. 然后是编辑(Edit)功能。对于有该权限的用户，我们需要让他能够通过该系统进行博客内容的实时编辑，以及对系统上所有博文的修改、增加、删除、查找功能，以此来实现对于系统上的博文内容的管理；
3. 最后是登录(Login)功能。因为系统不可能允许任意的访问者修改系统上的内容，因此需要通过登录/用户管理的功能来实现对于访问者权限的控制。只有哪些拥有管理权限的用户才能在登录之后修改博文内容，而其他的普通用户/游客用户只能浏览系统上的内容而不能编辑。

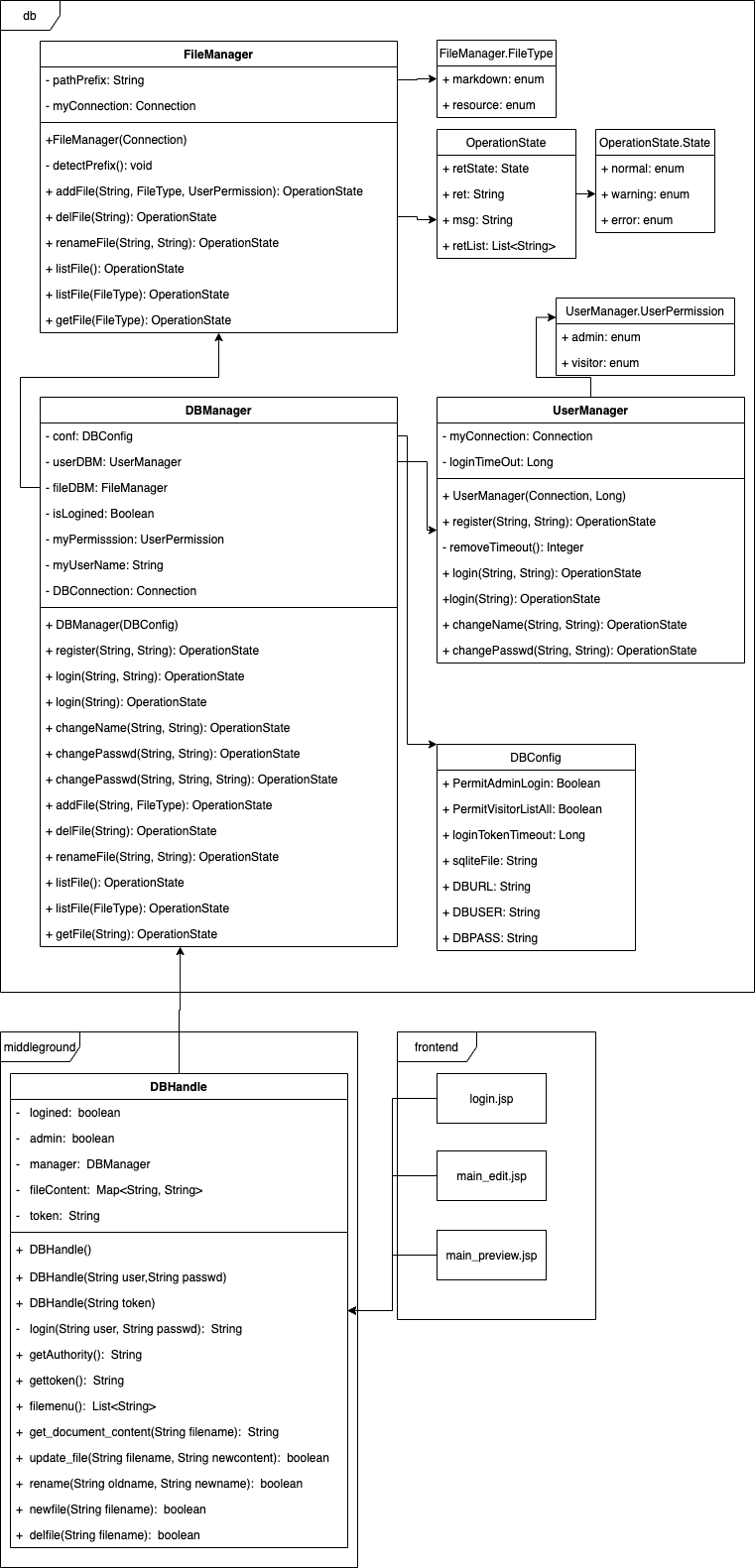
确定了上面的功能需求之后，我们便可以对应地将所有的用户分为两种：

1. 普通用户(General User)：只能查看和浏览系统上的内容，没有管理和修改的权限；
2. 管理员用户(Admin)：能够对博客系统上托管的内容进行查看和编辑，拥有对博客系统的管理权限；

有了上面提到的所有内容，我们就可以确定如右图所示的顶层用例图了。

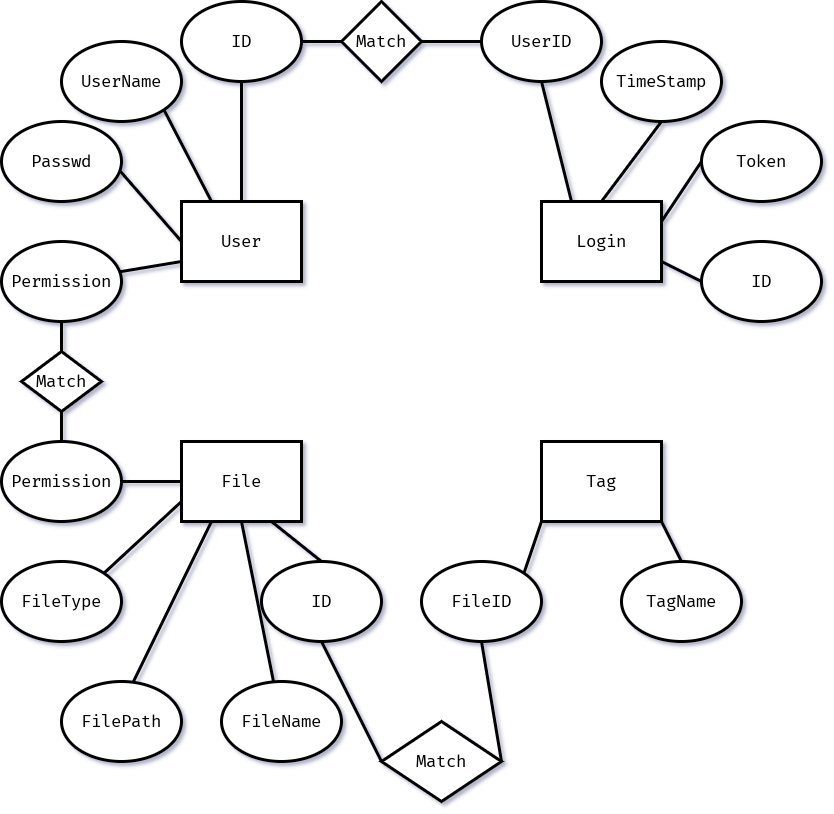
2、类图  
（可以把每个主要的jsp文件当成一个类）

然后根据要求我们还需要确定整个项目结构的类图。这里类图中主要就是将我们在中间件、后端部分实现的主要的java类以及它们的成员数据、方法、关系给表示出来。具体的内容如下图所示：



3、E-R图  
（表之间的关系图）

然后，因为项目的后端要使用到数据库，所以我们还需要确定如何在数据库中对我们系统用到的数据进行管理。因此我们需要确定好我们数据库的E-R图。



4、数据库设计

| 表名：User | | | |
| --- | --- | --- | --- |
| 字段名 | 中文名 | 类型 | 说明 |
| ID | 用户ID | INTEGER | 系统生成，自增 |
| UserName | 用户名 | VARCHAR | 用户名，不允许重复 |
| Passwd | 密码 | VARCHAR | 密码，未加密 |
| Permission | 权限 | INTEGER | 权限码，越低权限越高 |

| 表名：Login | | | |
| --- | --- | --- | --- |
| 字段名 | 中文名 | 类型 | 说明 |
| ID | 登录会话ID | INTEGER | 系统生成，自增 |
| UserID | 用户ID | INTEGER | 外键，关联到用户ID |
| Token | 登录会话Token | VARCHAR | 登陆Token，UUID |
| TimeStamp | 登陆时间戳 | BIGINT | 登陆时的毫秒时间戳 |

| 表名：File | | | |
| --- | --- | --- | --- |
| 字段名 | 中文名 | 类型 | 说明 |
| ID | 文件ID | INTEGER | 系统生成，自增 |
| FileName | 文件名 | VARCHAR | 文件名 |
| FilePath | 文件实际路径 | VARCHAR | 文件路径 |
| FileType | 文件类型 | INTEGER | 文件类型码 |
| Permission | 权限 | INTEGER | 文件权限码，与用户表中的权限对应 |

【运行截图】

这里我们给出项目运行的结果。

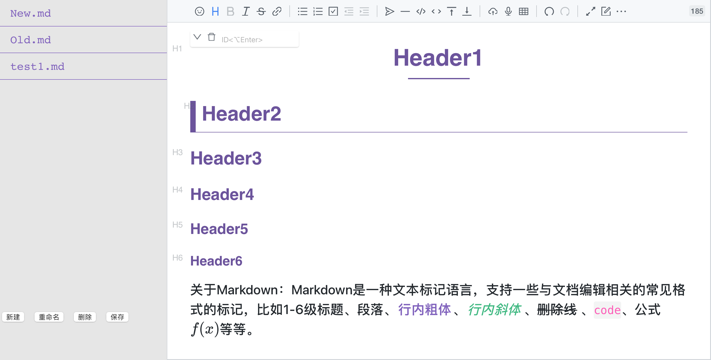
首先是初始页面，即登录界面：



然后是管理员账户登录，用户名admin、密码123：



登录之后跳转到编辑管理页面main\_edit.jsp，在这里可以查看每个博客文档的具体内容并对其进行编辑、管理：



上图中，左侧为所有文档博客构成的列表、下方有对文档进行管理、选择的工具栏。右侧为一个Markdown编辑器，可以在其中对文档内容进行查看和编辑。

Main\_preview的结果与main\_edit类似，除了少了编辑和修改功能。这里就不再贴出相关结果了。

以上就是项目的实验结果。

【实验体会】

写出实验过程中遇到的问题，解决方法和实验体会，每位同学单独写。

[同学1钟赫明]

主要遇到的问题：

如何完成jsp中的java代码和js的交互；如何进行jsp文件之间的跳转、如何保证从login.jsp到main\_xxx.jsp过程中对登录状态的检查和保持。如何与后端数据进行交互。这些问题都是和实现相关的，所以它们的解决方法即为相关源文件的实现内容。

还有一些和环境、部署相关的问题。比如如何在tomcat日志中获取后端程序的debug打印内容、jsp中如何debug与jsp变量相关的内容。前者的话，对于Apache-tomcat-10.0.6，可以在日志文件catalina.out中找到System.out.println打印的内容；后者，可以将jsp变量赋值成js变量，然后用alert打印出来。

实验体会：

本来以为做一个简单的博客系统会花不了多长时间，但是在项目实现过程中我们还是遇到了许多问题。比如：要如何正确地整合前端、中间件、数据库这三个模块？jsp项目要如何与其他用到的开源前端项目整合？项目小组中的不同成员之间要如何磨合？而且在一些细节实现上我们也遇到了许多的挑战。总之，这次项目我们的实现过程并不是很顺利，遇到了许多技术上、合作上的问题，但是我们最终还是成功完成这个项目。

[同学2郑立锋]

在本次实验中，我负责完成部分JSP文件的JAVA代码编写，主要功能是调用中端实现的函数接口对输入进行判断，并给JS提供url参数以方便跳转。所以在我的视角看来，底层函数是已经实现并且封装好的，作为前端，只需要调用已经实现的函数，并利用返回值进行逻辑判断，再跳转到对应JSP文件即可，对于工具栏的实现，也是调用已经实现的函数，再将所需要的参数传递给JS。在实现过程中，我也参考了作业jsp1，因此完成这一部分时难度不是很大。

而在实现完每个人的部分后，将各个部分组合在一起DeBug时，却遇到了许多难题，诸如：改一个函数名或者返回值，许多部分都要一起修改，增大工作量；多个文件组织在一起出bug时，有时会出现tomcat崩溃现象，此时DeBug就只能进入日志文件逐个查找，较为麻烦。此外，由于小组内分工问题，DeBug的部分基本是由组长一人完成，工作量很大，十分辛苦，这也导致实验耗时延长。

[同学3周伯霖]

遇到的问题：

我们的任务是实现中间件，为前端提供可靠的接口。

我们遇到的第一个问题是文件内容的缓存。由于在设计该类的时候就已经考虑到会对文件内容进行缓存以减少磁盘读写次数，类的每个实例都会保存一个自己的缓存，这是非常占用服务器内存的，而且会发生脏读等错误。

我们想到的第一个方法是DBHandle类使用单例模式，这样我们就获得了一个全局上的缓存。然而这个方法很快被我们自己否决了，因为每个实例是对应一个用户的登陆状态的，这部分内容肯定不能共享。

我们的第二个办法是仅对文件内容的缓存使用静态变量，但是在后面的调试过程中，我们发现静态变量文件缓存并不能真正发挥作用。由于每次跳转到新的网页之后，都会开启新的进程，其中并没有保存着我们先前的静态变量。

于是我们取消了静态变量的使用。

实验体会：

实验过程中由于对JAVA的使用并不熟悉导致出现很多语法上的bug，并且由于我们的分工决定了前后端以及中间件需要共同合作才能执行并debug，出现在后端和中间件的bug往往比较难发现。

[同学4张翊健]

遇到的问题：

最初的设计中，中间件需要实现两个类，一个用来进行文件管理，一个用来进行登陆管理，

登陆管理和文件管理都需要数据库管理类的一个实例，由于数据库管理类保存了登陆信息，我们这两个类必须使用同一个数据库管理类的实例。

此外，文件管理需要登陆管理中保存的登陆状态和用户权限等信息，因此两个类之间需要通信。

我们的第一个办法是使用继承，但由于这两个类不具有逻辑上的继承关系，使用继承则不具有可解释性。

我们的第二个办法就是朴素地合并了两个类，并将类名修改为DBHandle。

第二个问题是文件管理的具体实现，我们从后台获取URL，并直接对其进行读写。

[同学5张烨禧]

本次实验中我负责实现的是数据库接口的部分。这方面主要的难点是要理解前面其他人的工作，并且设计出适合他们使用的接口，若果设计得当，将能减少其他人的工作量。这里面的一个比较重要的点就是登录状态表和登录Token；在原本的设计中，这这方面应该交给中台和前端来实现，数据库不应该存储这些时效性比较强的信息，但是在经过与其他同学交流之后，发现由数据库来存储可以大幅减轻他们的工作，因此我们也在数据库实现中完成了这个功能。

另外，实验中我也比较了MySql和SQLite这两个SQL数据库的实现和JDBC接口，最后发现以我们的需求而言，这两个数据库都能完美完成任务；因此，考虑到Debug难度和环境的一致性，权衡之下我们最后选用了SQLite实现我们的数据持久化。

【提交报告】

每个小组交一份实验报告。

截止时间：2021年6月26日 23:00 (周六)

上传网站：<http://172.18.187.251/netdisk/default.aspx?vm=18web>

实验上交/大作业/实验报告

上传文件：  
 (1) 小组号\_项目简称\_实验报告.doc

(2) 小组号\_项目简称\_组员名单.xlsx

(3) 小组号\_项目简称.rar (所有源程序文件)

(4) 小组号\_项目简称.mp4 (操作录像)