**云计算实验报告**

**学号：2018522108**

**姓名：徐龙**

1. **实验环境搭建**

**1.1硬件环境**

雷神笔记本、16G内存、1TB+128GB外存、CPU酷睿i7八代、GPU-GTX1050。

**1.2软件环境**

Windows10操作系统、MySQL数据库8.0.19、Navicat Premium 12、JDK1.8、IDEA 2019.1、Tomcat8服务器。

**1.3环境搭建步骤**

1. 下载MySQL数据库在命令行中创建用户名和密码，创建数据库表进行测试是否安装成功。
2. 下载Navicat Premium 12连接上一步创建的数据库，使用该图形化界面对数据进行操作。
3. 安装JDK1.8，安装Tomcat8，安装IDEA javaIDE，之后在IDEA上配置JDK和Tomcat8使之能够编写java程序，并在tomcat服务器上运行。
4. 创建数据库，使用IDEA连接数据库，测试能否通过编写java代码对数据库进行操作。
5. **实验原理与步骤**

**2.1实验目的与原理**

**2.1.1目的**

通过小组实现网盘程序，熟悉Hadoop系统的部署与使用，体会云计算的实际应用，锻炼对B/S架构的理解以及javaEE的web程序的实践开发。

**2.1.2原理**

用户通过网页登陆or注册网盘账户，可以在网盘中进行下载、上传、删除文件。

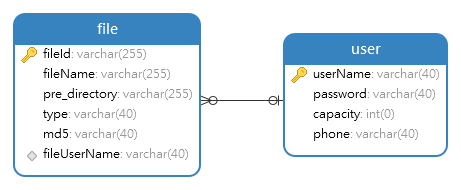
网盘View层采用css+html +JavaScript实现、Controller层采用Servlet实现、Dao层采用JPA实现、Model层采用javaBean实现，数据库使用Mysql数据库，文件存储采用hadoop技术文件分布系统。

**2.2实验具体步骤**

首先进行实验分工，在小组中本人负责后端数据库表的设计与网盘底层数据传输，因此只讲解数据库是如何实现并与Controller层进行数据传输。

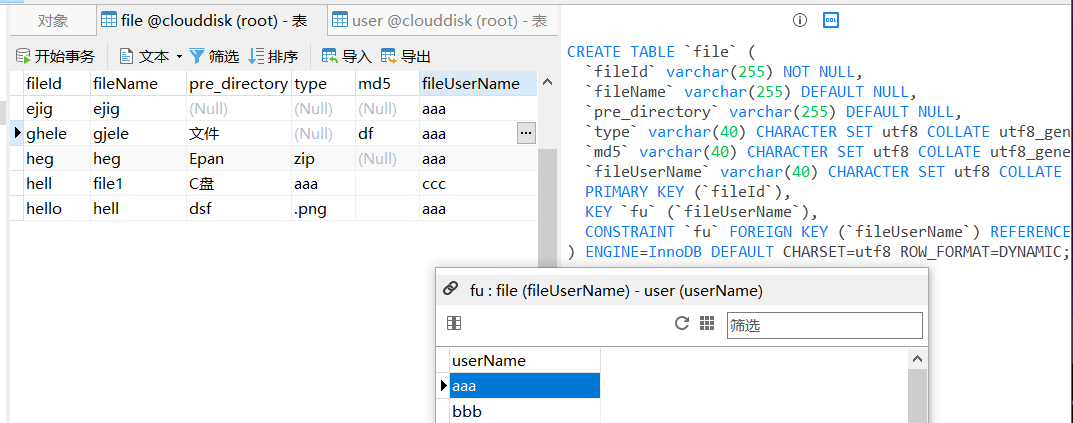
**2.2.1数据库建表设计**

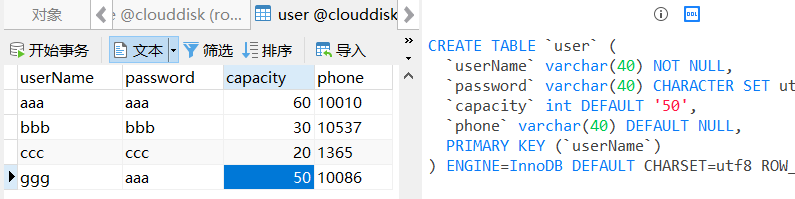
首先为了能够实现用户的登陆与注册功能，因此数据库必须要建立一个User表来存储UserName、Password等属性，之后由于用户要存储文件，因此必须还要创建一个File表来存储文件信息，由于用户可以存储多个文件，因此User表与File表为一对多关系。由此在Navicat创建如下模型：

****

其中file表中fileId为主键，fileUserName为外键与user表中的userName关联，user表中userName为主键。这样就要求用户在注册账户时，要求注册的用户名不能在数据中已经存在，进行用户名检测。这样做的目是在不影响整个系统的使用情况下简化数据库表的结构，方便对数据库进行操作。

**2.2.2数据库表生成**

 根据设计的模型，使用Navicat将其转换为数据库表，在表中添加数据测试多对对映射和其他属性，生成的表如下图所示：

**file表**

**user表**

**2.2.3 java代码实现类到数据库的映射**

在IDEA中编写两个实体类user和file，用于实现和数据库的映射，定义的类内主要变量如下：



**user类**

其中fileList存放用于存放该用户所拥有的文件集合，当从数据库查找到一个user后，将查找结果以user对象的方式返回，该user对象中的fileList包含该user的所拥有的文件集合，其他变量的值与数据库中属性值一一映射。

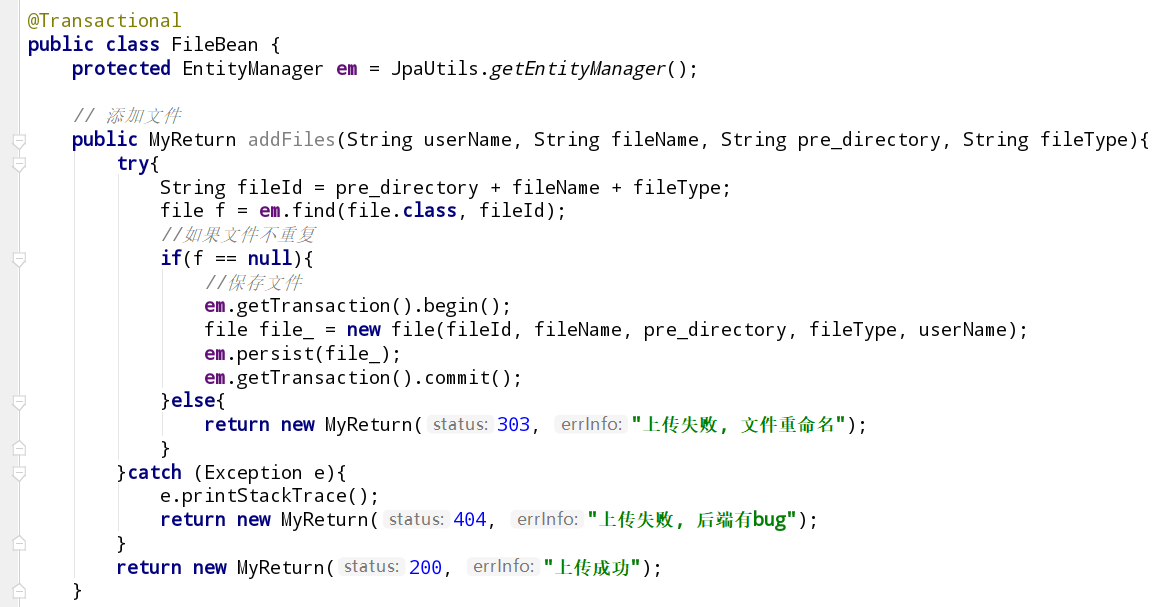


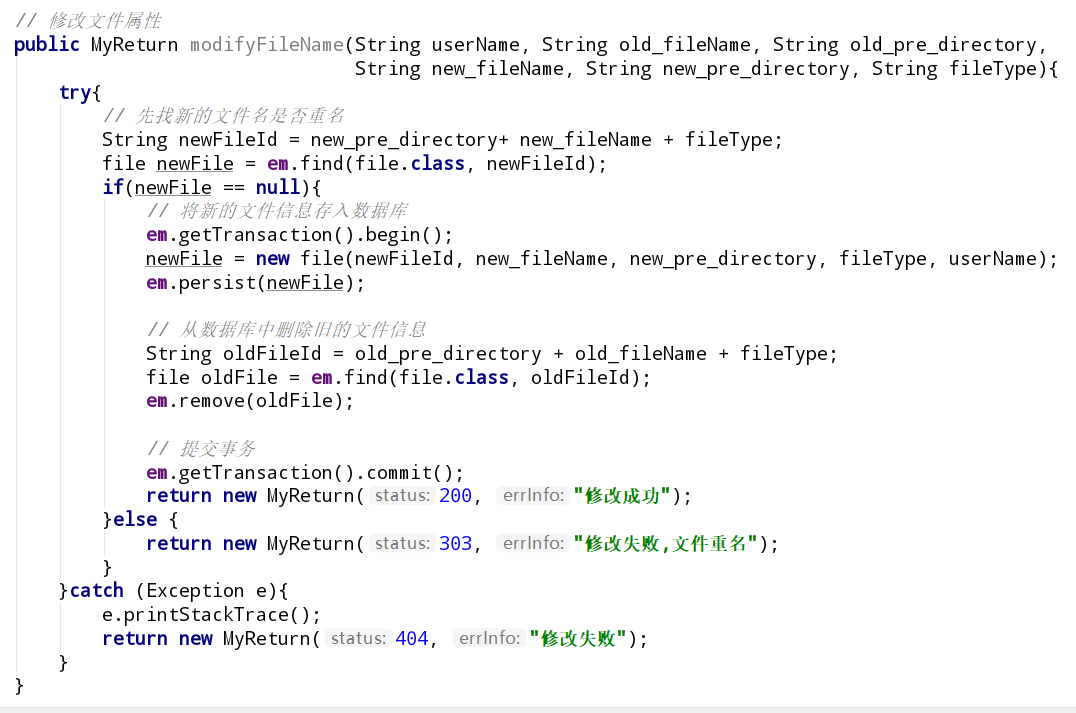
file类

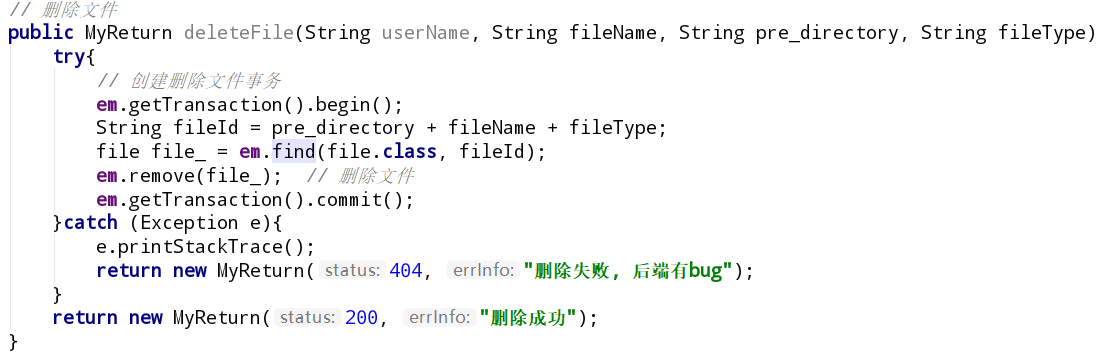
file类用于与file表建立映射关系，用于用户对文件的增删改查。

**2.2.4 创建实体bean对数据库进行操作**

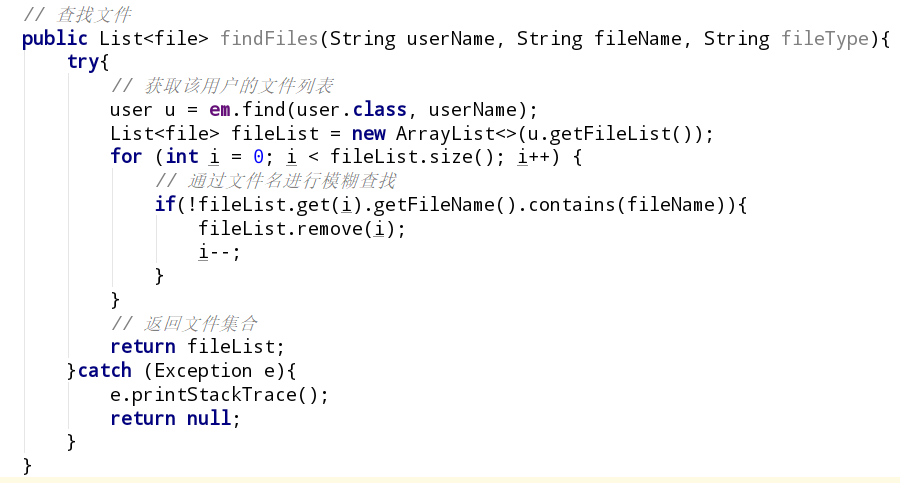
创建两个实体bean，fileBean和ServiceBean，fileBean用于对文件的操作，ServiceBean用于对用户的操作。

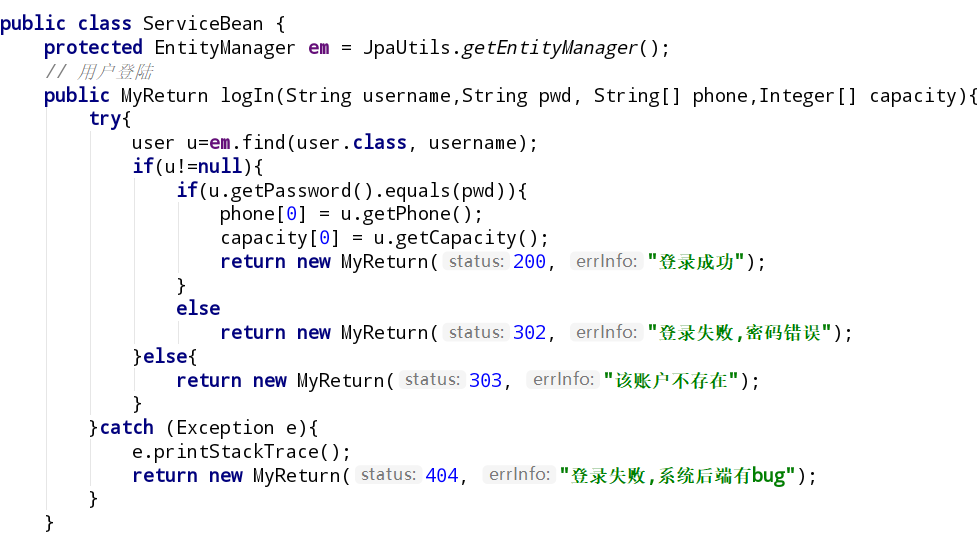
【fileBean主要函数】

addFIles()函数通过前端传来的用户名、文件名、文件目录、文件类型。将该文件信息存入数据库，并与该用户建立所属关系。返回值MyReturn包含了添加文件的结果，失败or成功。

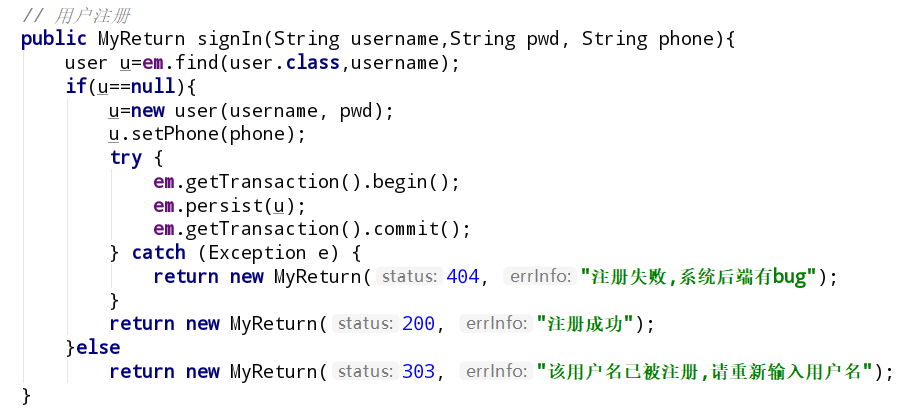
modifyFileName()函数用于修改用户所拥有的文件属性，修改文件所在文件目录，文件名。实现机制为添加新的文件信息，删除旧的文件信息。

deleteFile()函数用于删除用户所拥有的文件，传入参数为文件名和文件目录。

findFIles()函数用于查找文件，返回一个文件列表，

【ServiceBean主要函数】

LogIn()函数用于用户登陆网盘程序，通过前端传来的用户名和密码，来从数据库中查找该用户，之后返回查询结果



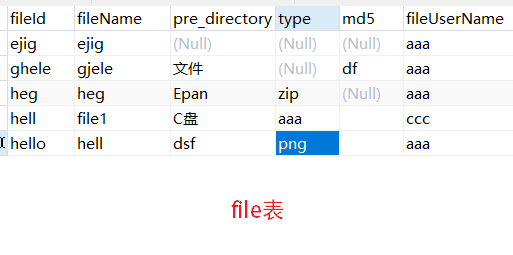
signIn()函数用于用户注册，该函数通过前端传来的用户名和密码，在数据库中添加一条用户信息。

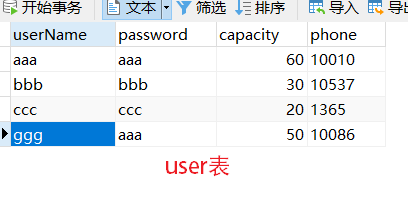
1. **实验结果与分析**

**3.1实验结果**

通过导入 org.junit.Test包对实现的各种函数进行测试，根据数据库中的内容测试结果如下：

数据库中已存储的数据





**fileBean中的函数测试**

addFiles()函数：

测试用例：

|  |  |  |  |
| --- | --- | --- | --- |
| userName | fileName | pre\_directory | fileType |
| aaa | 文件1 | F盘 | png |
| aaa | heg | Epan | zip |

结果：MyReturn属性值为：202、“上传成功”。

结果：MyReturn属性值为：303、“文件重命名”。

modifyFileName()函数：

测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| userName | old\_fileName | old\_pre\_directory | new\_fileName | new\_pre\_directory | fileType |
| aaa | 文件1 | F盘 | heg | Epan | png |
| aaa | 文件1 | F盘 | 文件one | E盘 | png |

结果：MyReturn属性值为：303、“修改成功”。

结果：MyReturn属性值为：303、“修改成功”。

deleteFile()函数：

测试用例：

|  |  |  |  |
| --- | --- | --- | --- |
| userName | fileName | pre\_directory | fileType |
| aaa | 文件one | E盘 | png |

结果：MyReturn属性值为：202、“删除成功”。

findFiles()函数：

测试用例：

|  |  |  |
| --- | --- | --- |
| userName | fileName | fileType |
| aaa | file1 | aaa |
| ccc | file1 | aaa |

结果：MyReturn属性值为：202、“上传成功”。

结果：MyReturn属性值为：303、“文件重命名”。

**ServiceBean中的函数测试**

logIn()函数：

测试用例：

|  |  |
| --- | --- |
| userName | password |
| zzz | 000 |
| aaa | aaa |
| ccc | 111 |

结果：MyReturn属性值为：303、“该用户名不存在”。

结果：MyReturn属性值为：200、“登陆成功”。

结果：MyReturn属性值为：302、“登陆失败，密码错误”。

signIn()函数：

测试用例：

|  |  |
| --- | --- |
| userName | password |
| zzz | 000 |
| aaa | aaa |
| ccc | 111 |

结果：MyReturn属性值为：303、“该用户名不存在”。

结果：MyReturn属性值为：200、“登陆成功”。

结果：MyReturn属性值为：302、“登陆失败，密码错误”。

**3.2结果分析**

同上结果展示

1. **遇到的问题和解决办法**
2. **小组团队合作问题：**

在小组协作编程时，常常因为变量名，和定义的数据结构不统一而导致在程序测试的时候出现问题。

**解决办法：**

在团队合作的时候的时候一定要商讨清楚变量名，对象，数据结构定义等信息，并在项目文档中给出范例，让团队小组成员都按照文档的格式编写，在需要修改文档时，一定到告知其他团队成员，并发布修改后的项目文件，以做到项目版本的统一。

1. **数据库设计问题**

* 一开始再设计数据库时，为了减少数据冗余，采用了用户和文件多对多关系，一个用户可以拥有多个文件，一个文件可以分享给多个用户，在实现的时候fileId采用自增长的方式生成。
* 但是这样的话就不能通过fileId查找文件，fileId没有被使用到，如果要想实现多对多关系，那么就必须建立文件信息映射表，一个文件可以通过哈希函数唯一的映射到一个fileid上，这样就能够通过获取fileid来查找文件了。
* 另外一个问题是由于一个文件可以被多个用户共享，因此要建立一个用户文件索引，当用户点击删除文件时，在数据库上操作时，尽是删除了该用户对该文件的索引，而不是文件本身，当该文件索引数为0时，则才真正的删除该文件。
* 但是构建映射表和文件索引需要更多的时间难度大，但是时间不允许，所以只能改变数据库表结构，采用一对多的方式实现。

1. **心得体会（或者对实验的建议）**
2. 希望老师能教一下hadoop如何使用，如何部署，我们大家之前都没学过hadopp和租用服务器导致做这个实验的时候，需要现学先用，这就很难受。
3. 个人感觉这个实验难度有些高，总体难度比我们JavaEE小学期的实现的项目难度还高，而javaEE所给的小组开发时间为两个星期，开发出一个javaweb程序，有前后端。而这个实验不仅要求要有前后端，而且要使用hadoop进行文件分布存储，需要租用服务器，但是开发时间仅有四次上课的时间，完全不够，需要课下再进行小组交流开发。
4. 通过此次实验我提高了对JavaWeb项目开发的经验，对团队交流与合作编程积累了经验，对云计算的使用有了更深刻的了解。