**上海电力学院**

**Java EE课程设计**



**题 目: 云盘客户端(PC端)应用系统**

**学 号： 20131809 20131805**

**姓 名： 黄海琨 顾珏**

**院 系： 计算机科学与技术学院**

**专业年级： 计算机科学与技术2013级**

**2016年7月5日**

**目录**

[1 设计题目 1](#_Toc455524385)

[1.1 实验题目 1](#_Toc455524386)

[1.2 问题描述 1](#_Toc455524387)

[1.3 基本要求 1](#_Toc455524388)

[1.3.1 系统主要功能 1](#_Toc455524389)

[2 需求分析 2](#_Toc455524390)

[2.1 功能描述 2](#_Toc455524391)

[2.1.1 用户 2](#_Toc455524392)

[2.2 组织结构 2](#_Toc455524393)

[2.3 功能详情 2](#_Toc455524394)

[2.3.1 用户 2](#_Toc455524395)

[2.4 数据库设计 4](#_Toc455524396)

[2.4.1 数据库E-R图 4](#_Toc455524397)

[2.4.2 数据库描述 4](#_Toc455524398)

[2.4.3 关系模型 4](#_Toc455524399)

[2.4.4 数据库表 4](#_Toc455524400)

[3 详细设计 6](#_Toc455524401)

[3.1 文件上传（含秒传功能） 6](#_Toc455524402)

[3.2 文件下载 8](#_Toc455524403)

[3.3 在线浏览播放 11](#_Toc455524404)

[3.4 文件复制模块 13](#_Toc455524405)

[3.5 文件删除模块 15](#_Toc455524406)

[3.6 文件共享模块 17](#_Toc455524407)

[3.7 文件搜索模块 18](#_Toc455524408)

[4 关键处和创新点 20](#_Toc455524409)

[4.1 音乐播放（创建线程） 20](#_Toc455524410)

[4.2 拖拽上传 21](#_Toc455524411)

[5 课程总结 22](#_Toc455524412)

[5.1 主要错误 22](#_Toc455524413)

[5.1.1 文件中文乱码 22](#_Toc455524414)

[5.1.2 Hibernate 主键自增问题 22](#_Toc455524415)

[5.1.3 MP3文件无法播放 23](#_Toc455524416)

[5.2 小结 23](#_Toc455524417)

# 设计题目

## 实验题目

云盘客户端(PC 端)应用系统

## 问题描述

云盘现在都是拥有浏览器访问和客户端访问，在客户端上可以进行文件和文件夹的基本操作，可以方便用户进行云盘的使用和管理。本地客户端还可以在线同步文件，可以实时管理文件。

## 基本要求

### 系统主要功能

1. 通过云盘客户端上传文件至服务器
2. 客户端浏览管理云盘服务器文件（复制、粘贴、删除、移动、重命名、新建文件夹）
3. 文件查找
4. 云盘服务器文件下载
5. 图片在线预览
6. 视频音频播放功能
7. 文件共享
8. 文件秒传

# 需求分析

## 功能描述

### 用户

用户注册登录界面后进行云盘文件管理

## 组织结构

本云盘客户端系统拥有文件上传、文件下载、文件管理和用户模块。

* 文件上传：普通上传；文件秒传；拖拽上传；
* 文件下载：个人文件下载；分享文件下载；
* 文件管理：复制粘贴；文件删除；文件查找；在线浏览；文件重命名；文件夹新建；文件分类查看；
* 用户：用户登录；



图2-1 云盘客户端系统组织结构图

## 功能详情

### 用户

用户首先登录系统，登录成功后进入，首先进入自己的主页，在客户端里可以对文件进行上传、下载、新建文件夹、重命名、删除操作。文件分类浏览，多媒体文件在线浏览，文件拖拽上传等功能。



图2-2 用户用况图

## 数据库设计

### 数据库E-R图



图2-3 云盘客户端系统ER图

### 数据库描述

本系统采用MySQL数据库，MySQL数据库采用结构化的数据库查询语言，是一种开源的关系型数据库。

### 关系模型

* 用户（用户编号、用户账号、用户密码、状态）
* 文件（编号、文件名、拥有者、文件格式、格式、MD5、上级路径、路径、下载路径、创建时间、状态、共享情况）

### 数据库表

表2-1 文件表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 属性名 | 属性描述 | 类型 | 长度 | 主键 | 备注 |
| file | FileNo | 编号 | int |  | Y |  |
| FileName | 文件名 | nvarchar | 100 | N |  |
| UserNo | 拥有者 | int |  | N | 用户编号 |
| FileFormat | 文件格式 | nvarchar | 10 | N | 对应文件后缀名 |
| FileType | 格式 | int |  | N | 0：文件夹 1：文件 |
| FileMD5 | MD5 | nvarchar | 40 | N |  |
| SupFolder | 上级路径 | int |  | N | 上级文件夹编号 |
| FilePath | 路径 | nvarchar | 150 | N | 虚拟路径 |
| DownloadPath | 下载路径 | nvarchar | 150 | N | 绝对路径 |
| CreatTime | 创建时间 | datetime |  | N |  |
| FileStatus | 状态 | int |  | N | 0：未启用 1：启用 |
| ShareStatus | 共享情况 | int |  | N | 0：未分享 1：分享 |

表2-2 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 属性名 | 属性描述 | 类型 | 长度 | 主键 |
| User | UserNo | 用户编号 | int |  | Y |
| UserName | 用户账号 | nvarchar | 20 | N |
| UserPwd | 用户密码 | nvarchar | 20 | N |
| UserStatus | 状态 | int |  | N |

# 详细设计

## 文件上传（含秒传功能）

用户点击上传文件按钮，在系统打开时候就创建了一个HttpClient，选择上传的文件，查找该文件在当前目录中是否存在，如果存在结束，如果不存在，计算文件MD5，上传文件MD5查看文件是否已在云端拥有，如果拥有，上传其他参数，不需要上传文件（秒传），如果没有，上传文件再上传其他参数，返回函数值，提醒上传成功与否。



图3-1 文件上传功能流程图

**【代码】**

**public** String Upload(String fileRealName,String filePath,**int** userNo,**int** supfloder,String path) **throws** IOException, NoSuchAlgorithmException {  
 String hash = Md5CaculateUtil.*getHash*(filePath,**"MD5"**);  
 **if**(**this**.CkeckMd5(**"checkmd5"**,hash)){  
 System.***out***.println(**"Exist"**);  
 **return this**.AddFile(**"addfile"**,fileRealName,filePath,userNo,supfloder,path,hash);  
 }**else**{  
 System.***out***.println(**"NotExist"**);  
 String rvalue = **this**.FileUpload(hash,filePath);  
 rvalue = rvalue.substring(0, rvalue.length() - 2);  
 **if**(**"success"**.equals(rvalue)){  
 System.***out***.println(**"CopyFileSuccess"**);  
 **return this**.AddFile(**"addfile"**,fileRealName,filePath,userNo,supfloder,path,hash);  
 }  
 **return "error"**;  
 }  
}

**public** String FileUpload(String hash,String filePath)**throws** IOException, NoSuchAlgorithmException {  
 String rvalue=**""**;  
 **try** {HttpPost httpPost = **new** HttpPost(**"http://localhost:8080/SkyDrive/servlet/UploadServlet"**);  
 *// 把文件转换成流对象FileBody*  
 File file = **new** File(filePath);  
 FileBody bin = **new** FileBody(file);  
 StringBody md5 = **new** StringBody(hash, ContentType.*create*(**"text/plain"**, Consts.***UTF\_8***));HttpEntity reqEntity = MultipartEntityBuilder.*create*()  
 .setCharset(Charset.*forName*(HTTP.***UTF\_8***))  
 .setMode(HttpMultipartMode.***BROWSER\_COMPATIBLE***)  
 .addPart(**"md5"**, md5)  
 .addPart(**"file1"**, bin).build();  
 httpPost.setEntity(reqEntity);CloseableHttpResponse response = **httpclient**.execute(httpPost);  
 **try** {HttpEntity resEntity = response.getEntity();  
 **if** (resEntity != **null**) {  
 rvalue= EntityUtils.*toString*(resEntity, **"UTF-8"**);  
 }EntityUtils.*consume*(resEntity);  
 } **finally** {  
 response.close();  
 }  
 }**return** rvalue;  
 }

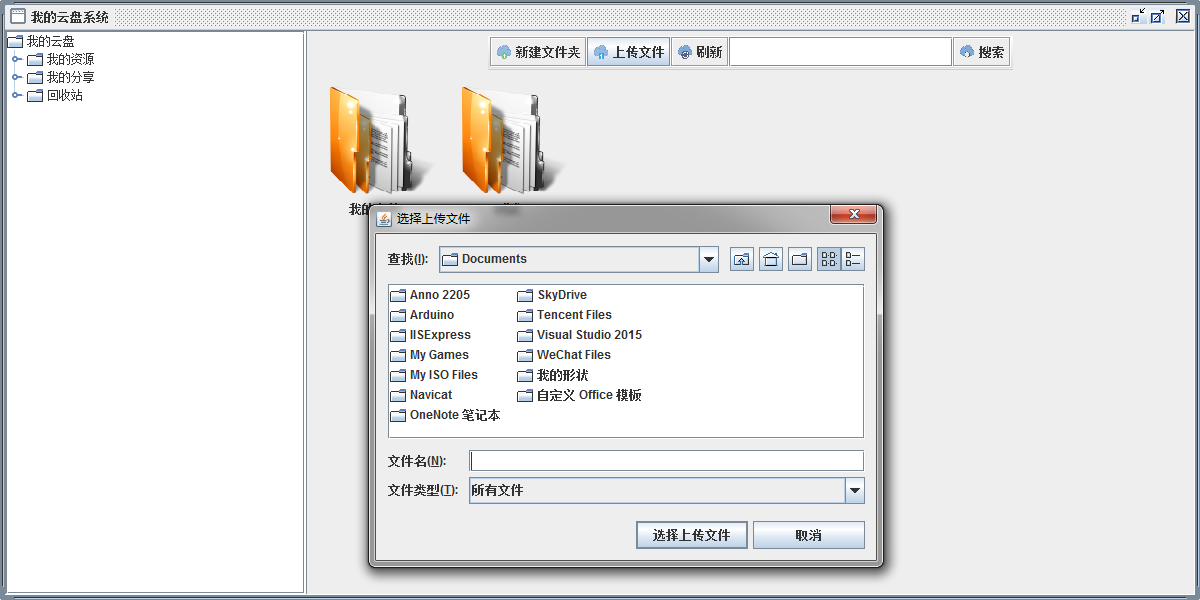


图3-2 文件上传功能实现

## 文件下载

用户右键选择文件，选择文件下载目的地目录，上传文件编号，servlet返回文件在服务器中的路径，客户端接收到路径，请求一个流文件形式下载，从服务器下载文件，然后保存到本地指定目录中。



图3-3 文件下载功能流程图

**【代码】**

**public** String Download(String opttype, **int** fileno, String fileName, String savePath) **throws** IOException {  
 *// 创建httppost* HttpPost httppost = **new** HttpPost(**"http://localhost:8080/SkyDrive/servlet/DownloadServlet"**);  
 *// 创建参数队列* List<BasicNameValuePair> formparams = **new** ArrayList<BasicNameValuePair>();  
 formparams.add(**new** BasicNameValuePair(**"fileNo"**, String.*valueOf*(fileno)));  
 formparams.add(**new** BasicNameValuePair(**"opttype"**, opttype));  
 String downLoadPath = **this**.GetEntity(httppost, formparams);  
 downLoadPath = downLoadPath.substring(0, downLoadPath.length() - 2);  
 System.***out***.println(**"DownLoadPath:"** + downLoadPath);  
 **if** (**this**.DownMethod(downLoadPath, savePath, fileName) == **true**) {  
 **return "success"**;  
 } **else** {**return "error"**;}  
}

**public** Boolean DownMethod(String downpath,String savepath,String fileName) **throws** IOException{  
 System.***out***.println(savepath);  
 **try**{HttpPost post = **new** HttpPost(downpath);  
 HttpResponse response = **httpclient**.execute(post);  
 **if**(HttpStatus.***SC\_OK***==response.getStatusLine().getStatusCode()){  
 HttpEntity entity = response.getEntity();  
 **if** (entity != **null**) {  
 File storeFile = **new** File(savepath,fileName);  
 FileOutputStream output = **new** FileOutputStream(storeFile);  
 InputStream input = entity.getContent();  
 **byte** b[] = **new byte**[1024];  
 **int** j = 0;  
 **while**( (j = input.read(b))!=-1){  
 output.write(b,0,j);  
 }output.flush();output.close();  
 }}**return true**;  
 }**catch** (Exception e){  
 e.printStackTrace();  
 **return false**; }  
}



图3-4 文件下载功能实现

## 在线浏览播放

系统在打开的时候就在制定路径中创建了一个临时文件夹，用户选择文件右击，选择浏览播放，判断文件类型，如果不是媒体文件类型，不给予操作，如果是操作文件的类型，下载文件到临时文件夹，打开播放。



图3-5 在线浏览播放功能流程图

**【代码】**

**public** String MediaDownload(String opttype, **int** fileno,String fileName,String savePath,**int** otptype) **throws** IOException {  
 *// 创建httppost* HttpPost httppost = **new** HttpPost(**"http://localhost:8080/SkyDrive/servlet/DownloadServlet"**);  
 *// 创建参数队列* List<BasicNameValuePair> formparams = **new** ArrayList<BasicNameValuePair>();  
 formparams.add(**new** BasicNameValuePair(**"fileNo"**, String.*valueOf*(fileno)));  
 formparams.add(**new** BasicNameValuePair(**"opttype"**, opttype));  
 String downLoadPath = **this**.GetEntity(httppost, formparams);  
 downLoadPath = downLoadPath.substring(0, downLoadPath.length() - 2);  
 System.***out***.println(**"DownLoadPath:"**+downLoadPath);  
 Properties prop = System.*getProperties*();  
 savePath = **"C:\\Users\\"**+prop.getProperty(**"user.name"**)+**"\\Downloads"**+**"\\temp"**;  
 **if**(**this**.DownMethod(downLoadPath,savePath,fileName) == **true**){  
 **return** savePath+**"\\"**+fileName;  
 }**else** {**return null**;  
 }  
}

**if** (fileformat.equals(**"mp3"**)) {  
 String filePath = **dao**.MediaDownload(**"download"**, **fileno**, **filename**, **null**, 1);  
 **if** (filePath != **null**) {  
 JFrame.*setDefaultLookAndFeelDecorated*(**true**);  
 MediaFrm mediaFrm = **new** MediaFrm(filePath, **filename**);  
 mediaFrm.setSize(280, 180);  
 mediaFrm.setVisible(**true**);  
}  
**if** (fileformat.equals(**"jpg"**) || fileformat.equals(**"png"**)) {  
 String filePath = **dao**.MediaDownload(**"download"**, **fileno**, **filename**, **null**, 1);

**if** (filePath != **null**) {  
 JFrame.*setDefaultLookAndFeelDecorated*(**true**);  
 ImageViewer imageViewer = **new** ImageViewer(filePath);  
 imageViewer.setSize(1024, 768);  
 imageViewer.setVisible(**true**);  
}

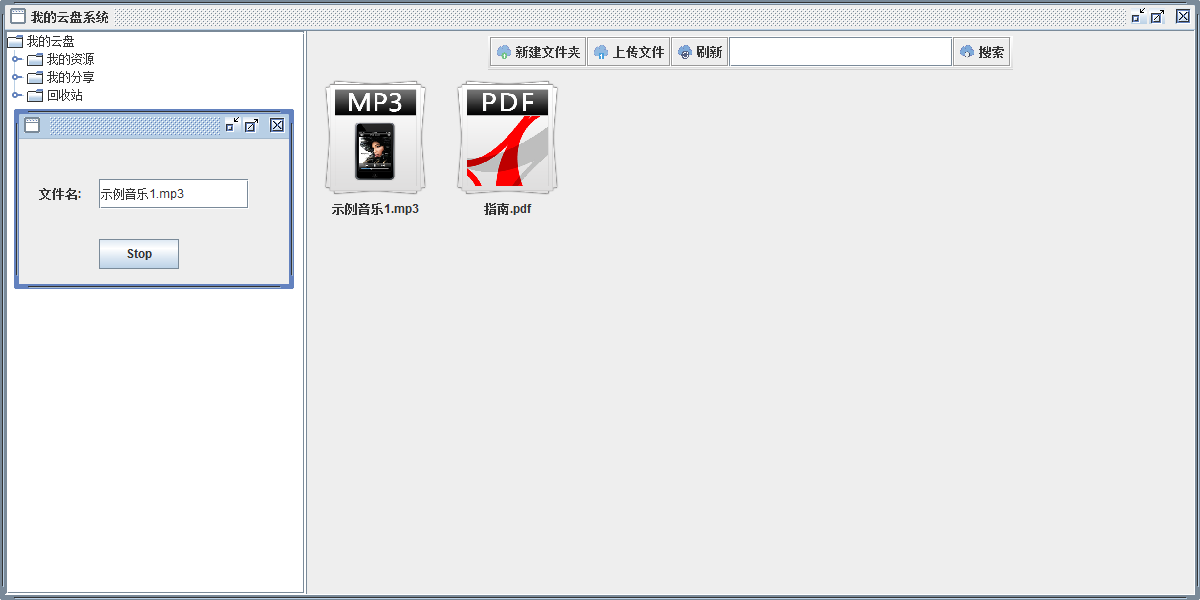


图3-6 音乐在线播放功能实现



图3-7 图片在线浏览功能实现

## 文件复制模块

文件复制模块是云盘客户端系统中文件管理中的内容，文件复制由用户进行。用户进入个人云盘页面之后可以选择文件进行复制，再选择想要复制文件的位置进行粘贴。选择复制的文件后判断所选是否为文件夹，若是文件夹则清除选择，否则进行粘贴。在所选位置进行粘贴时，查询当前路径下是否有同名文件，如果有则提示错误，否则粘贴并刷新面板。至此，流程结束。



图3-8 文件复制模块流程图

**【代码】**

Files files = **filesDao**.findFile(fileno);  
List<Files> filesList = **filesDao**.queryFileNoInPathByFileName(files.getUserNo(), path, files.getFileName());  
**if** (filesList.size() != 0)  
 out.print(**"error"**);  
**else**{  
 Files paste = **new** Files();  
 **if** (path != 0) {  
 paste.setFilePath(dir.getFilePath() + **"/"** + dir.getFileNo());  
 paste.setSupFolder(dir.getFileNo());  
 } **else** {  
 paste.setFilePath(**"../"** + files.getUserNo());  
 paste.setUserNo(files.getUserNo());  
 }  
 **filesDao**.addFile(paste);

## 文件删除模块

文件删除模块是云盘客户端系统中文件管理中的内容，文件删除由用户进行。用户进入个人云盘页面之后可以选择文件进行删除。文件在服务器端进行查找，如果文件是文件夹，则查找文件夹路径下的所有文件，对文件进行依次删除，等所有文件删除完毕再删除文件夹；若不是文件夹则直接删除。删除文件本质上是数据库信息的修改，即将文件的状态修改为删除，而非真实删除文件。直至文件删除结束，重新查找文件并刷新面板。至此，文件删除模块结束。



图3-9 文件删除模块流程图

**【代码】**

Files files=**filesDao**.findFile(fileno);  
**if**(files.getFileType()==1){  
 **if** (!**filesDao**.disableFile(fileno))  
 out.print(**"error"**);  
}  
**if**(files.getFileType()==0){  
 List<Files> filesList=**filesDao**.queryFileInPath(files.getFilePath()+**"/"**+files.getFileNo());  
 **for**(Files files1:filesList){  
 **filesDao**.disableFile(files1.getFileNo());  
 }  
 **filesDao**.disableFile(fileno);  
}

## 文件共享模块

文件共享模块是云盘客户端系统中文件管理中的内容，文件共享由用户进行。用户进入个人云盘页面之后可以选择文件进行共享。选择文件共享后，服务器进行文件的查询，查询文件的分享值，若为未共享则设置为共享，并在面板上提示文件共享的路径；若文件已开启共享，则将共享关闭，并进行提示。至此，文件共享模块结束。



图3-10 文件共享模块流程图

**【代码】**

**int** fileno = Integer.*valueOf*(request.getParameter(**"FileNo"**));  
Files files=**filesDao**.findFile(fileno);  
**if**(files.getShareStatus()==0)  
 **filesDao**.enableShareFile(fileno);  
**else**{  
 **filesDao**.disableShareFile(fileno);  
 out.print(**"disShare"**);  
}

## 文件搜索模块

文件搜索模块是云盘客户端系统中文件管理中的内容，文件搜索由用户进行。用户进入个人云盘页面之后可以在搜索框填写自己需要搜索的文件名。填写完后，发送至服务器进行模糊查询，并更新面板。至此，文件搜索模块结束。



图3-11 文件搜索模块流程图

**【代码】**

**if** (!**searchText**.getText().equals(**""**)) {  
 String result = **dao**.Search(**"search"**, **userNo**, **searchText**.getText());  
 dispFiles(result);  
 **mainpane**.repaint();  
}

**int** user = Integer.*valueOf*(request.getParameter(**"UserNo"**));  
String filename=request.getParameter(**"FileName"**);  
List<Files>filesList=**filesDao**.queryLikeFileInAllPath(user,filename);

# 关键处和创新点

## 音乐播放（创建线程）

音乐播放采用Java线程编程方法，可以实现同步操作和播放音乐。音乐的播放导入了mp3plugin.jar包可以支持mp3音频的播放，采用javax.sound类库进行播放音乐。如要播放视频就要采用JMF（Java Media Framework）来实现了，这里没有视频的在线播放。

**【代码】**

**private** String **filePath** = **null**;  
Runnable1 **r** = **new** Runnable1();  
Thread **t** = **new** Thread(**r**);  
**public void** StartPlayer(String filePath) **throws** Exception {  
 **this**.**filePath** = filePath;  
 **t**.start();  
}  
**public void** StopThread(){  
 **t**.stop();  
}  
**class** Runnable1 **implements** Runnable{  
 **public void** run() {  
 **byte**[] buffer = **new byte**[***BUFFER\_SIZE***];  
 File soundFile = **new** File(**filePath**);  
 **try** {  
 AudioInputStream in = AudioSystem.*getAudioInputStream*(AudioFormat.Encoding.***PCM\_SIGNED***, AudioSystem.*getAudioInputStream*(soundFile));  
 AudioFormat audioFormat = in.getFormat();  
 SourceDataLine line = (SourceDataLine) AudioSystem.*getLine*(**new** DataLine.Info(SourceDataLine.**class**, audioFormat));  
 line.open(audioFormat);  
 line.start();  
 **while** (**true**) {  
 **int** n = in.read(buffer, 0, buffer.**length**);  
 **if** (n < 0) {**break**; }  
 line.write(buffer, 0, n);  
 }  
 line.drain();  
 line.close();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
}



图4-1 音乐播放示例

## 拖拽上传

文件的拖拽上传是将文件从桌面进行拖拽至面板上，进行上传。文件拖拽至面板时，获取文件的信息再调用文件上传的方法，进行上传。为完成拖拽上传需要定义拖拽方法，并为面板设置接受拖拽。

**【代码】**  
**public void** drag()*//定义的拖拽方法*{ *//panel表示要接受拖拽的控件* **new** DropTarget(**mainpane**, DnDConstants.***ACTION\_COPY\_OR\_MOVE***, **new** DropTargetAdapter() {  
 @Override  
 **public void** drop(DropTargetDropEvent dtde)*//重写适配器的drop方法* {**try** { **if** (dtde.isDataFlavorSupported(DataFlavor.***javaFileListFlavor***))*//如果拖入的文件格式受支持* {dtde.acceptDrop(DnDConstants.***ACTION\_COPY\_OR\_MOVE***);*//接收拖拽来的数据* List<File> list = (List<File>) (dtde.getTransferable().getTransferData(DataFlavor.***javaFileListFlavor***));  
 dtde.dropComplete(**true**);*//指示拖拽操作已完成* } **else** {  
 dtde.rejectDrop();*//否则拒绝拖拽来的数据* }  
 } **catch** (Exception e) {  
 e.printStackTrace();

}

}  
 });  
}

# 课程总结

## 主要错误

### 文件中文乱码

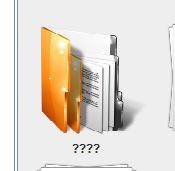


图5-1 文件中文乱码错误

**【解决方案】**

将Servlet的DoPost方法头部加入：

request.setCharacterEncoding(**"UTF-8"**);  
response.setCharacterEncoding(**"UTF-8"**);

将编码格式进行转换为UTF-8，消除中文乱码

### Hibernate 主键自增问题

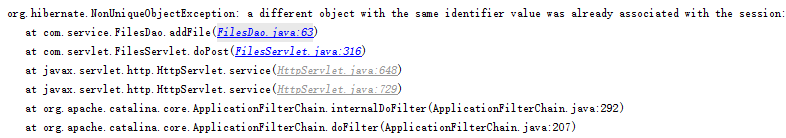


图5-2主键自增问题错误

**【解决方案】**

**为主键添加自增的注解**

@Id  
@Column(name = **"FileNo"**, nullable = **false**)  
@GeneratedValue(strategy = GenerationType.***AUTO***)  
**public int** getFileNo() {  
 **return fileNo**;  
}

### MP3文件无法播放

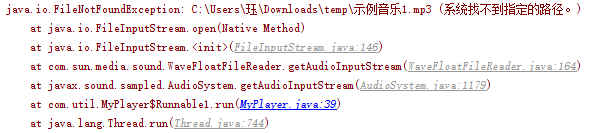


图5-3 MP3文件播放错误

**【解决方案】**

在Java\jdk1.7.0\_51\jre\lib\ext目录下添加mp3plugin.jar

## 小结

在本次课程设计中，采用HttpClient来模仿网页的Http请求操作，所有的操作都基于HttpClient的HttpPost，通过HttpPost来选择访问的Servlet来实现不同的操作。在实际编程过程中，带参数的文件上传过程花了一点时间来理解，最后成功理解整个流对象的传输过程。

客户端的Http的编程这是第一次接触，现在很多客户端程序都是采用这种方式来访问服务器，再由服务器来访问数据库，这样既可以完成网页的服务器访问，也可以完成客户端的服务器访问，节约了服务器的资源和开发人员的工作量，数据访问更加安全。