

大连东软信息学院

本科毕业设计（论文）

论文题目：MyDisk 云盘管理系统的设计与实现

**The Design and Implementation of MyDisk Cloud Disk
Management System**

系 所： 软件工程系

专 业： 软件工程

学生姓名： 宋铮星

学生学号： 14110600109

指导教师： 彭志豪

导师职称： 副教授

完成日期： 2018 年 04 月 15 日

大连东软信息学院
Dalian Neusoft Institute of Information

MyDisk 云盘管理系统的设计与实现

摘 要

随着时代的发展，网络时代的步伐也在逐步地靠近，人们已经逐渐跨入科技时代，人们的東西越来越多，但行囊却越来越轻，就是因为网络和云盘的存在，使得人们只需要带上手机，或是笔记本电脑，在有网络的情况下就可以随意下载自己的文件，视频，查看自己的东西。

本系统使用的开发技术是 Spring+SpringMVC+Mybatis 框架，Java 语言进行开发，数据库方面主要使用了 Mysql 关系型数据库，前台采用 JSP 动态页面，采用 B/S 架构，使用 IntelliJ IDEA 2016.3(64)完成的一个云盘管理系统。本系统主要涉及用户操作，关于用户操作的页面有登录/注册页，主页，个人信息设置页，本系统实现的主要功能有文件的上传，下载和删除操作，文件与文件夹的拖拽，文件夹的加密，个人信息的修改等功能。

云盘的出现方便了文件的传输，不用再使用 U 盘，硬盘这样存储文件的东西，大大释放了行囊和包裹。而且相对于一些比较传统的文件存储方式来说，云盘的安全性也比较出众，只要记得用户名和密码，就可以再有互联网的地方轻松取得自己想要的文件。本系统也可以对文件进行排版，使得文件堆放有序。

关键词：云盘操作，Java，Mysql，MVC，MyBaits，Spring

The Design and Implementation of MyDisk Cloud Disk Management System

Abstract

With the development of the era, the pace of the Internet era is also gradually approaching. People have gradually stepped into the era of science and technology. There are more and more people's things, but their bags are getting lighter and lighter because of the existence of networks and cloud disks. People just need to bring a mobile phone or a laptop, in the case of a network, you can freely download your own files, videos, and view your own stuff.

The development technologies used in this system is the Spring+Spring MVC+Mybatis framework. The Java language is used for development. The database mainly uses the Mysql relational database. The front desk uses JSP dynamic pages, adopts the B/S architecture. The cloud management system completed using IntelliJ IDEA 2016.3 (64). This system mainly involves user operations. There are login/registration pages, homepages, and personal information settings pages for user operations. The main functions implemented in this system include uploading, downloading and deleting files, dragging and dropping files and folders, and files. Folder encryption, personal information modification and other functions.

The appearance of the cloud disk facilitates the transfer of files, we don't need to use U disk, hard disk to store files, greatly released bags and packages. And compared to some of the more traditional file storage methods, cloud disk security is also superior, as long as you remember the user name and password, you can easily get the files you want with the Internet. The system can also typeset the files so that the files are stacked in order.

Key words: cloud disk operation, Java, Mysql, MVC, MyBaits, Spring

目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 项目概述.....	1
1.1 问题描述.....	1
1.2 目的及意义.....	1
1.3 项目适用范围.....	1
1.4 涉众.....	2
第 2 章 需求分析.....	3
2.1 业务需求.....	3
2.2 功能性需求.....	3
2.2.1 用例概述.....	3
2.2.2 用例描述.....	5
2.3 非功能性需求.....	9
(1) 技术性.....	9
(2) 经济性.....	9
(3) 合法性.....	10
2.4 运行环境.....	10
第 3 章 系统设计.....	11
3.1 技术路线.....	11
3.2 软件体系结构设计.....	11
3.2.1 功能结构.....	11
3.2.2 系统架构.....	11
3.3 外部接口.....	12

3.4 数据库设计.....	12
3.4.1 概念结构设计.....	12
3.4.2 表的详细设计.....	13
3.5 用例实现.....	14
3.5.1 文件上传用例实现.....	14
3.5.2 文件下载用例实现.....	15
3.5.3 文件删除用例实现.....	16
3.5.4 文件或文件夹拖拽用例实现.....	16
3.5.5 文件加密功能用例实现.....	17
3.6 对象设计.....	17
3.6.1 模块设计.....	17
3.6.2 业务实体设计类图.....	18
3.6.3 业务对象设计类图.....	19
第 4 章 系统实现.....	20
4.1 核心功能实现.....	20
4.1.1 文件夹加密/解密模块.....	20
4.1.2 头像上传功能模块.....	24
4.1.3 文件上传/下载功能模块.....	26
4.1.4 文件/文件夹拖拽功能模块.....	28
4.2 关键技术难点与解决方案.....	30
第 5 章 系统测试.....	31
5.1 功能测试.....	31
5.1.1 文件相关操作模块测试用例.....	31
5.1.2 个人信息修改模块测试用例.....	31
5.3 测试总结.....	32

第 6 章 结论与展望.....	33
参考文献.....	34
致 谢.....	35

第 1 章 项目概述

1.1 问题描述

自互联网发展以来，它提供了大量的信息存储，高速的信息传播，广泛的受众群体和多点互动等优势，迅速的互联网扩张使其成为目前重要的媒介，它改变和加速了信息的传播，还有人们的生活方式和媒体习惯。首先，互联网的普及使得大众传媒广为流行，爆炸式的发展和信息的传播使人们更容易获得各种信息，网盘的出现为文件的传输，和存放以及文件的分享提供了极好的平台。其次，用户积极参与传播过程的意识受到极大的刺激。在大众传播中，普通人发挥的作用已经从被动接受者转变为积极的传播者。文件传播的主体越来越多样化，网盘使得群众的生活更加便捷。

本论文所研究的是云盘管理系统的设计与实现，该系统严格按照软件工程的步骤来实现，从需求分析、系统的设计、系统的实现到系统的测试都是按照规定步骤进行操作，并且采用建模语言对整个系统进行分析与设计。本系统主要由云盘管理，文件上传，文件下载等功能模块构成。用户可以通过使用该系统进行文件的上传和下载，文件的管理，例如文件夹的分类，删除和重命名等一些比较简单的操作。

1.2 目的及意义

由于现在网络大范围的覆盖了人们的生活，所以一些实体提供的服务均可以在网络上实现，云盘也因此出现在人们的视野中，它可以提供海量的存储空间，不必局限于硬盘或 U 盘的存储空间。无论是硬盘还是 U 盘，无论发展到何种地步都是一个实体的物件，带在身上总会有一些负担，而且稍有不慎就有可能丢失，如果在硬盘或 U 盘里面放了比较重要的文件，就很容易造成一定的损失，网络云盘就可以避免这些情况的出现。网络云盘的操作简单，只要使用过电脑的都可以轻而易举的操作。只要有网络的地方就可以获取自己想要的东西，而且相对来说安全性也有了一定的提升。该系统有非常人性化的分文件夹操作，可以使用户轻而易举地将文件分类。

1.3 项目适用范围

用户可通过使用该管理系统对自己电脑中的文件进行一系列的管理，例如上传，下载，文件夹的新建，文件夹的拖拽等功能。用户还可以对自己的网盘进行文件管理，分类，对自己的云盘中的格式进行排版，以便查找到自己想要的內容，并且更好的对自己的云盘进行更加符合自己要求的管理。该管理系统适合人们对自己文件进行备份，方便携带，等特点。

1.4 涉众

早期时，人们使用网盘更多的是需要其存储的特性，U 盘普及之前，移动存储设备稀缺，1.44MB 的磁盘缺乏稳定性，而光盘又是一次性的，利用网盘存储相对更具有经济性与便利性。而当 U 盘普及之后，存储空间不再是稀缺的，网盘主要的用途转到了长距离分享,这主要利用了网络的优势。随着网盘的发展，越来越多的人抛弃了传统的存储方式，开始使用云盘来对文件进行传输和存储，虚拟化的云盘对大部分用户来说更加符合他们对文件存储和传输的需求。

第 2 章 需求分析

2.1 业务需求

一个用户可以上传多个文件，一个用户只可以拥有一个云盘，一个用户可以进行多条留言，用户可以对多个文件进行操作和编排，用户可以对使用文件夹对文件进行管理，领域模型如图 2.1 所示。

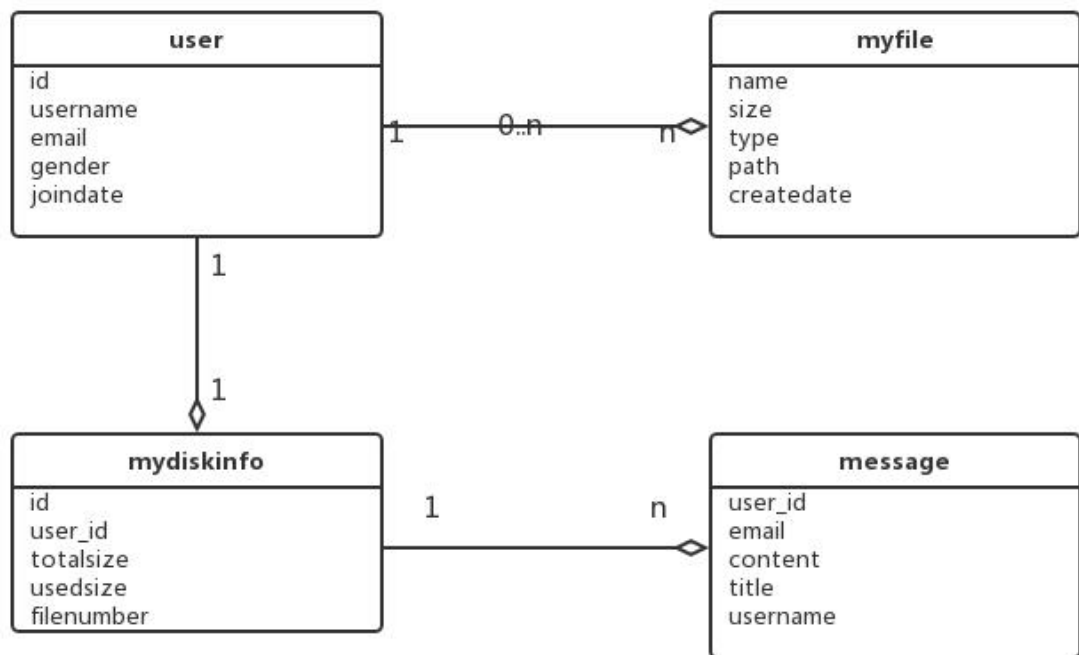


图 2.1 领域模型

2.2 功能性需求

2.2.1 用例概述

该系统的总用例图如图 2.2 所示，从下图用例中抽取一些比较重要的用例进行分析，例如，文件的上传，下载，文件的删除，文件与文件夹的拖拽功能，头像截取功能，文件夹的加密，解密，改密功能等操作，由于本系统是云盘系统，所以具有一定的私密性，如果擅自设定管理员去查看用户的云盘内容，难免会看到一些隐私的东西，这样会大大破坏用户对该系统的信任度，用户的体验也会受到极大的影响，而且在法律可行性上也会受到很大的影响，并且因为权限设计方面对于身为开发者的我有点困难，所以本系统不设计管理员。

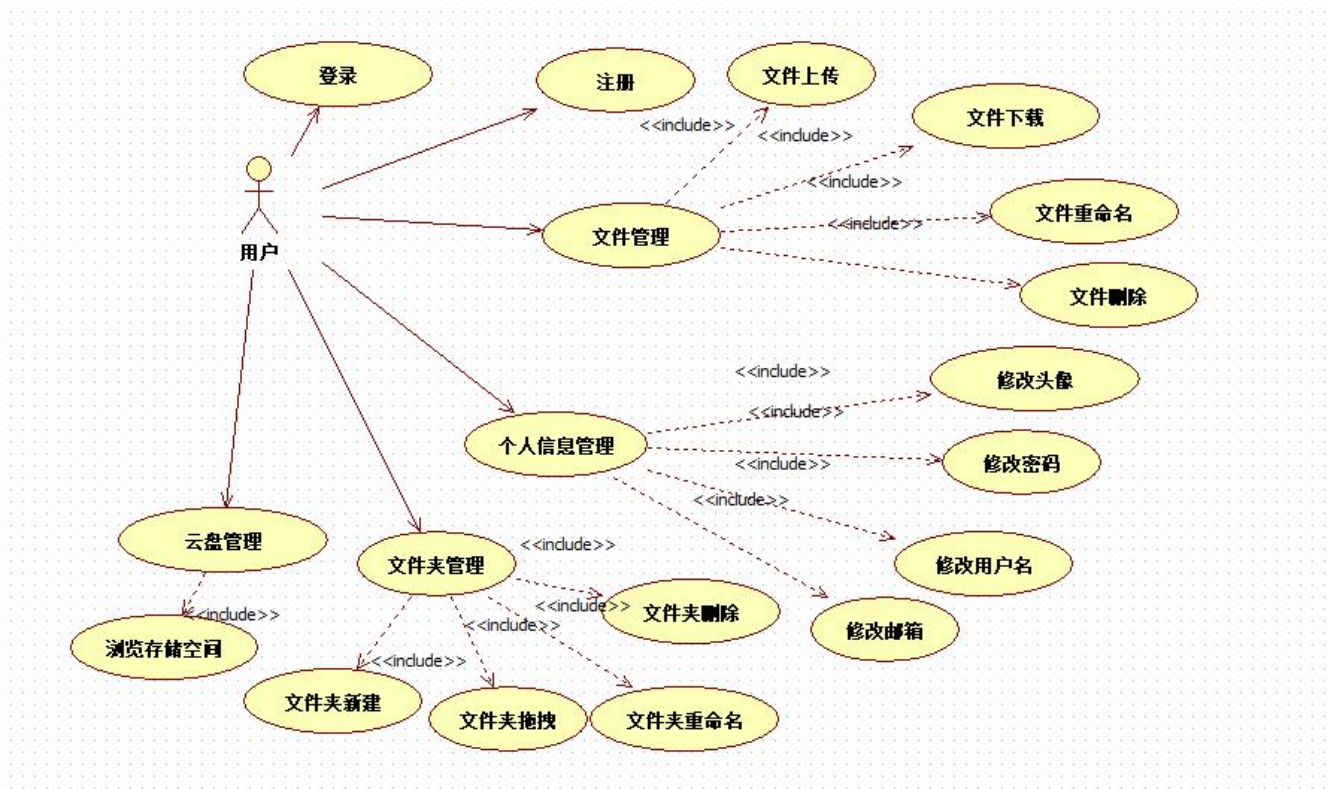


图 2.2 系统整体用例图

用例摘要描述如表 2.1 所示。

表 2.1 用例摘要描述

用例标识	用例名称	摘要描述
1	文件上传	用户上传文件
2	文件下载	用户下载文件
3	文件删除和重命名	用户删除文件或重命名文件
4	新建文件夹	用户新建文件夹
5	用户管理	用户基本信息修改
6	文件夹上锁	用户对文件夹进行加密
7	文件夹删除和重命名	用户对文件夹删除和重命名
8	文件及文件夹拖拽	用户对文件及文件夹拖拽
9	用户上传图片截取头像	用户上传图片截取头像

2.2.2 用例描述

（1）文件上传

文件上传用例描述如表 2.2 所示。

表 2.2 文件上传用例描述表

用例名称	文件上传
参与者	用户
前置条件	启动系统，登录用户账号。
基本事件流	<ol style="list-style-type: none"> 1. 用户点击“上传”按钮上传文件，系统弹出上传框； 2. 用户在上传框中操作进行文件上传，点击确定按钮； 3. 系统使屏幕左下方出现弹出框，这时用户点击上传按钮确认上传； 4. 文件上传成功，当前云盘的容量将会随着上传的文件大小有所减少，系统显示当前云盘剩余容量； 5. 用户刷新页面后，上传文件将出现到指定文件夹或云盘根目录下； 6. 上传完成，用例结束。
异常事件流	<ol style="list-style-type: none"> 4.1 因为文件过大，导致上传失败，云盘显示容量不变。 5.1 用户上传的文件大小大于云盘所剩余的容量，上传失败。
后置条件	显示更新后的文件及文件夹布局。

（2）文件下载

文件下载用例描述如表 2.3 所示。

表 2.3 文件下载用例描述表

用例名称	文件下载
参与者	用户
前置条件	启动系统，并登录系统
基本事件流	<ol style="list-style-type: none"> 1. 用户右键点击目标文件，出现下载按钮，点击下载按钮； 2. 用户点击弹出框，选择下载路径后点击下载按钮； 3. 文件下载成功，用例结束。
异常事件流	<ol style="list-style-type: none"> 3.1 电脑目标文件夹的容量不足，无法完成下载。
后置条件	下载完成，页面无改变。

（3）文件删除和重命名

文件删除和重命名用例描述如表 2.4 所示。

表 2.4 文件删除和重命名用例描述表

用例名称	文件删除和重命名
参与者	用户
前置条件	启动系统，并登录系统
基本事件流	<ol style="list-style-type: none"> 1. 用户右键点击目标文件，出现删除按钮； 2. 用户点击删除按钮进行删除； 3. 删除文件成功，用户右键选择第二个目标文件； 4. 用户点击文件重命名按钮； 5. 用户输入文件的新名字； 6. 文件重命名成功，用例结束。
异常事件流	无。
后置条件	页面刷新，文件布局改变。

（4）新建文件夹操作

新建文件夹操作用例描述如表 2.5 所示。

表 2.5 新建文件夹用例描述表

用例名称	新建文件夹
参与者	用户
前置条件	启动系统，用户登录
基本事件流	<ol style="list-style-type: none"> 1. 用户点击新建文件夹按钮新建文件夹； 2. 用户在弹出框中输入该文件夹的命名； 3. 文件夹创建完成； 4. 新建文件夹成功，结束用例。
异常事件流	无。
后置条件	文件和文件夹布局改变。

（5）用户管理

用户管理用例描述如表 2.6 所示。

表 2.6 用户管理用例描述表

用例名称	用户管理
参与者	用户
前置条件	启动系统，登录主页
基本事件流	<ol style="list-style-type: none"> 1. 用户选中个人信息设置按钮； 2. 用户点击修改基本信息； 3. 用户修改性别，用户名个人基本信息； 4. 用户刷新页面，头像及个人信息修改成功； 5. 用户点击修改密码按钮； 6. 用户修改登录密码，输入新密码和旧密码和确认新密码； 7. 用户修改密码成功； 8. 用例结束。
异常事件流	<ol style="list-style-type: none"> 3.1 用户名重复，修改失败； 5.1 用户两次输入的密码不同，修改失败； 5.2 用户旧密码输入错误，修改密码失败； 5.3 用户输入的新密码少于 6 位，提示后用户修改密码失败。
后置条件	正确显示变更后的个人信息。

（6）文件夹上锁

文件夹上锁用例描述如表 2.7 所示。

表 2.7 文件夹上锁用例描述表

用例名称	文件夹上锁
参与者	用户
前置条件	启动系统，登录主页
基本事件流	<ol style="list-style-type: none"> 1. 用户选择目标文件夹； 2. 用户右键点击目标文件夹，点击弹出的上锁按钮； 3. 用户在弹出框中输入该文件夹的密码，并重复确认输入该文件夹的上锁密码； 4. 上锁成功，用例结束。
异常事件流	<ol style="list-style-type: none"> 3.1 用户输入文件加密密码的时候，密码长度小于 4 位。
后置条件	文件夹被加密，文件夹左上方有个锁的图标。

（7）文件夹删除和重命名

文件夹删除和重命名用例描述如表 2.8 所示。

表 2.8 文件夹删除和重命名用例表

用例名称	文件夹删除和重命名
参与者	用户
前置条件	启动系统，登录主页
基本事件流	<ol style="list-style-type: none"> 1. 用户右键选择目标文件夹，点击删除按钮； 2. 经过系统提示用户点击确定后目标文件夹被删除； 3. 用户选择另一个目标文件夹，右键点击选择重命名； 4. 用户输入新名称后修改完毕； 5. 文件夹重命名成功，用例结束。
异常事件流	无。
后置条件	文件夹布局改变，文件夹名称改变。

（8）文件及文件夹拖拽功能

文件及文件夹拖拽用例描述如表 2.9 所示。

表 2.9 文件及文件夹拖拽用例描述表

用例名称	文件及文件夹拖拽
参与者	用户
前置条件	启动系统，登录主页
基本事件流	<ol style="list-style-type: none"> 1. 用户选择目标文件夹，左键点住； 2. 用户将该文件夹拖拽到想要让其成为另一个文件夹的子文件夹的上方； 3. 用户松开左键，目标文件夹消失，点击存放目标文件夹的文件夹可查看； 4. 用户选择目标文件，左键点住； 5. 用户将该文件拖拽到目标文件夹图标上方； 6. 用户松开左键，目标文件消失，点击存放文件的文件夹可查看； 7. 文件及文件夹拖拽成功，用例结束。
异常事件流	无。
后置条件	正确显示变更后的文件布局。

（9）用户上传图片截取头像

用户上传图片截取头像用例描述如表 2.10 所示。

表 2.10 用户管理用例描述表

用例名称	用户上传图片截取头像
参与者	用户
前置条件	启动系统，登录主页
基本事件流	<ol style="list-style-type: none"> 1. 用户点击个人信息设置按钮，进入修改个人信息页面； 2. 用户点击更改头像，准备进行头像修改； 3. 用户点击上传进行头像的上传，在弹出的文件选择框中选择电脑中已有的图片文件上传； 4. 用户在弹出框中点击上传按钮上传； 5. 用户用鼠标点击画框来截取图像，下方可看图片的坐标和宽高； 6. 用户可以点击画好的方框进行调整，也可以改变已经框好的图片大小； 7. 用户点击保存按钮进行头像的保存， 8. 页面刷新后头像改变； 9. 头像更改成功，用例结束。
异常事件流	无。
后置条件	正确显示变更后的头像。

2.3 非功能性需求

（1）技术性

因为本管理系统使用的是 java 语言为开发语言，使用了 spring+springMVC+Bybatis 框架，数据库方面则使用了 Mysql 关系型数据库，所以说无论是框架还是所使用的语言，再到所使用的数据库都是十分普遍的，网上有大量的开源代码。作为软件工程的学生，大学里以对该系统所需要的技术经过了系统的学习和大量的练习，因此具备了开发该软件的能力，因此具有技术可行性。

（2）经济性

这个系统的开发工具所使用的是 IntelliJ IDEA 2016.3 (64)，Tomcat8.0，Mysql，IntelliJ IDEA 2016.3(64)是一个十分好用的开发软件，Android Studio 开发软件就来源于 IntelliJ IDEA，费用上只需要购买就可以永久使用，tomcat 和 mysql 均为免费使用的开

发环境，所以说本软件的开发具有经济可行性。

（3）合法性

本系统无论是从开发角度还是使用角度，均不违反国家相应的法律法规，软件开发过程中未使用不正当手段获取他人代码，工具包和代码编写均为自己编写或网上开源代码拷贝，所以不涉及侵权。

2.4 运行环境

操作系统：Microsoft Windows 10, 64-bit

处理器：Intel(R) Core(TM)2 Duo CPU T9600 @ 2.80GHz 双核

内存：8GB

硬盘：320GB

软件：IntelliJ IDEA 2016.3 (64), Tomcat8.0, JDK1.8

第 3 章 系统设计

3.1 技术路线

本系统使用的开发技术是 Spring+SpringMVC+Mybatis 框架进行开发，操作界面友好，很容易上手，数据库方面主要使用了 Mysql 关系型数据库，干净简洁，非常的容易操作和修改，该系统采用 B/S 架构，该系统所使用的开发工具是 IntelliJ IDEA 2016.3(64)，是一个非常便捷的开发工具，由于本系统使用了 MVC 架构技术，所以把该系统分成了三大层次，即为 Model 层，View 层和 Controller 层，这样很有利于系统的解耦，使得该系统能有相好的可维护性和可扩展性。

3.2 软件体系结构设计

3.2.1 功能结构

根据之前需求的调研结果，结合已经上线的各大云盘的核心功能模块，该系统主要分以下几个模块，文件上传模块，文件下载模块，文件编辑模块，文件夹创建及修改模块，用户管理模块。详细结构如图 3.1 所示。

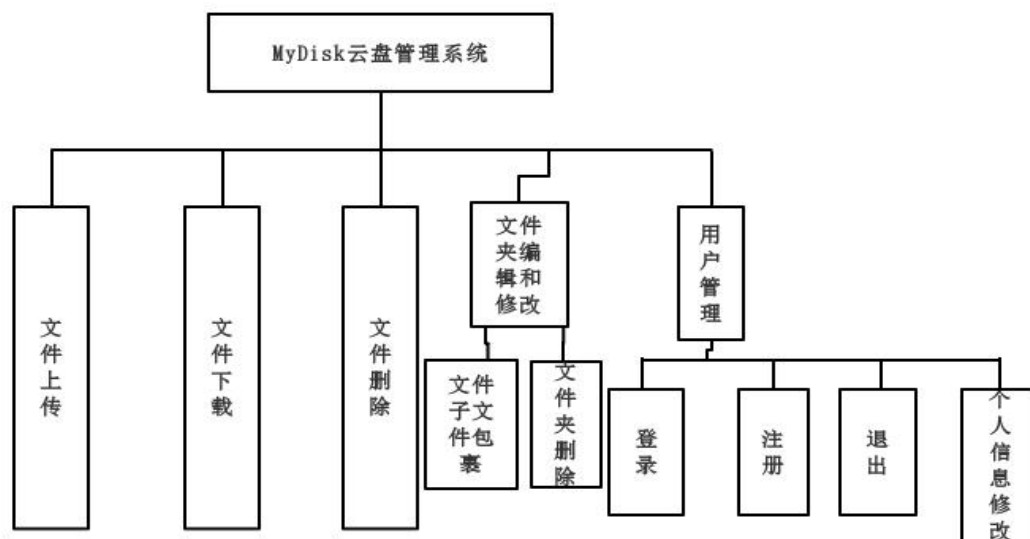


图 3.1 系统功能结构图

3.2.2 系统架构

本软件采用 MVC 模式分为三层架构，分为 web 端视图、后台控制层和业务逻辑层。后端服务又可以分为 Controller 控制层、Service 接口层、ServiceImpl 实现层、Dao 接口层和 XML 实现层等不同层次，所以后台系统大致可以分 Controller 控制层、Service 业务逻辑层和 Dao 数据访问层。而控制层（Controller）是 SpringMVC 框架对其进行管理，

业务逻辑层（Service）和数据访问层（Dao）是 Spring 框架对其进行管理。

3.3 外部接口

（1）Service 层接口设计

DeleteFileService: 提供删除文件及文件夹的业务接口。

MoveFileService: 提供用户拖拽文件夹的业务接口。

MyDiskInfoService: 提供显示用户网盘信息的业务接口。

MyFileService: 提供云盘中已有文件和文件夹基本操作的业务接口。

RegisterService: 提供显示上传文件的相关信息的业务接口。

UploadService: 提供刷新页面显示更新后文件及文件夹的业务接口。

UserService: 提供用户登录、注册、查询用户信息等业务接口。

（2）数据访问接口设计

DeleteFileDao: 提供删除云盘中已有文件夹和文件数据访问接口。

MessageDao: 提供展示用户基本信息的数据访问接口。

MoveFileDao: 提供用户拖拽文件夹的数据访问接口。

MydiskInfoDao: 提供显示用户主页信息的数据访问接口。

MyFileDao: 提供云盘中已有文件和文件夹基本操作的数据访问接口。

RegisterDao: 提供显示上传文件的相关信息的数据访问接口。

UploadDao: 提供刷新页面显示更新后文件及文件夹的数据访问接口。

UserDao: 提供用户登录、注册、查询用户信息等数据访问接口。

3.4 数据库设计

3.4.1 概念结构设计

本系统数据库设计包含 4 张表，分别为用户表、文件表、用户信息表和云盘信息表，数据库 E-R 图如图 3.2 所示。

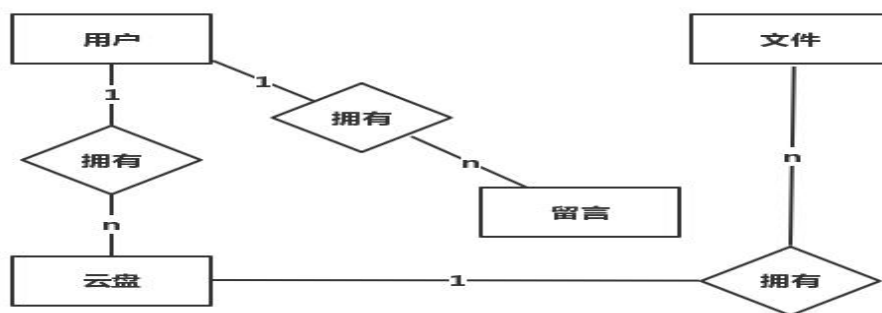


图 3.2 E-R 图

3.4.2 表的详细设计

(1) 用户表，用户表主要是用来存放用户的一些基本信息，如用户名，密码，邮箱等字段，用户表还有一个非常主要的功能，就是在本系统设计的时候，用户表要和所有的其他表进行关联，进行联合查询，所以说用户表结构的设计尤为重要，其具体表结构如表 3.1 所示。

表 3.1 USER 表

字段名称	字段描述	主键	类型	非空
id	用户唯一 id	是	INT(11)	是
email	用户注册邮箱	否	VARCHAR(50)	是
username	用户名	否	VARCHAR(30)	是
password	用户密码	否	char(32)	是
gender	用户性别	否	int(1)	是
portrait	用户头像	否	char(50)	否
joindate	注册日期	否	char(30)	是
state	用户状态	否	varchar(50)	否

(2) 用户留言信息表，用户可以进行留言，并且自己可以给自己设计备忘录，以方便提醒自己要要对云盘将要进行的操作，用户留言信息表的表结构具体设计如下表 3.2 所示。

表 3.2 message 表

字段名称	字段描述	主键	类型	非空
id	用户唯一 id	是	INT(11)	是
user_id	用户在该表外键	否	INT(11)	是
email	用户注册邮箱	否	VARCHAR(50)	是
username	用户名	否	VARCHAR(30)	是
title	用户留言标题	否	VARCHAR(30)	是
content	用户留言正文	否	Text	是

(3) 文件表，文件表是本系统尤为重要的一个表，对于云盘系统来说，文件表的设计需要格外谨慎，在表设计之前要将需求理清，比如说文件的格式，文件夹的加密的判定，文件和文件夹的路径设计，文件夹下的文件在删除文件夹时如何使用字段将它们关联起来，文件表的表结构设计具体如下表 3.3 所示。

表 3.3 myfile 表

字段名称	字段描述	主键	类型	非空
id	用户唯一 id	是	INT(11)	是
user_id	用户在该表外键	否	INT(11)	是
parent_id	文件 id	否	INT(11)	否
name	文件名称	否	VARCHAR(100)	是
size	文件大小	否	mediumtext	是
type	文件类型	否	Char(20)	是
path	文件路径	否	VARCHAR(500)	是
createdate	创建日期	否	Char(30)	否
password	文件上锁密码	否	VARCHAR(20)	否
islook	是否上锁	否	INT(1)	否
isshare	是否分享	否	INT(1)	否
sharedownload	分享路径	否	INT(11)	否
location	文件地址	否	VARCHAR(220)	否
shareurl	分享地址	否	VARCHAR(110)	否
md5	盐值	否	Char(32)	否
description	文件描述	否	VARCHAR(200)	否

(4) 云盘信息表，如表 3.4 所示。

表 3.4 mydiskinfo 表

字段名称	字段描述	主键	类型	非空
id	用户唯一 id	是	INT(11)	是
user_id	用户在该表外键	否	INT(11)	是
totalsize	云盘总容量	否	INT(11)	是
usedsize	云盘已用容量	否	INT(11)	否
filenumber	文件编号	否	INT(11)	否

3.5 用例实现

3.5.1 文件上传用例实现

文件上传功能，用户启动系统之后，登陆完成进入主界面，点击上传按钮上传文件，然后在屏幕的左下方会有弹出框，点击弹出框上的开始上传按钮，进度条开始读取，顺

序图如图 3.3 所示。

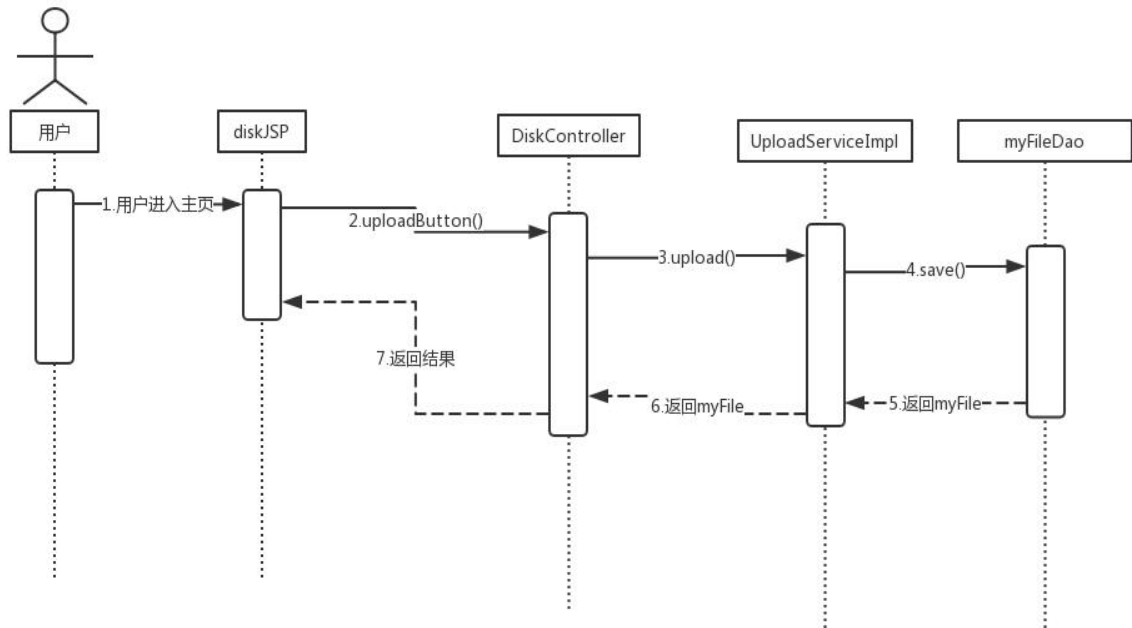


图 3.3 文件上传功能顺序图

3.5.2 文件下载用例实现

文件下载功能，用户启动系统之后，登陆完成进入主界面，右键点击目标文件或文件夹下载文件，顺序图如图 3.4 所示。

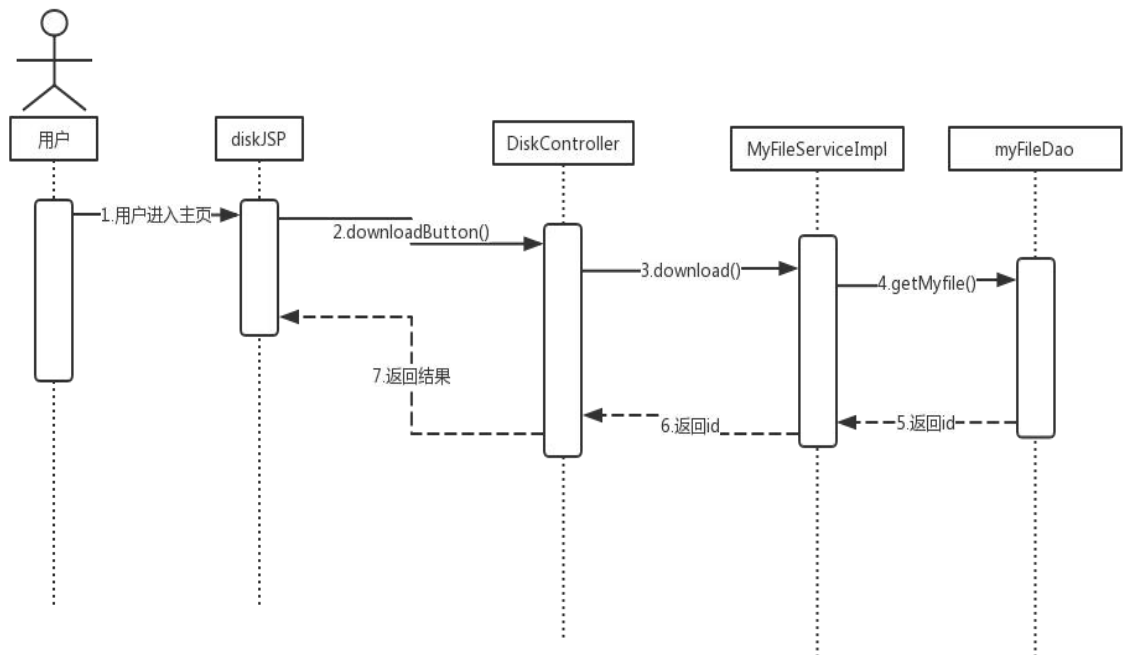


图 3.4 文件下载功能顺序图

3.5.3 文件删除用例实现

删除文件功能，用户启动系统之后，登陆完成进入主界面，右键点击目标文件删除文件，顺序图如图 3.5 所示。

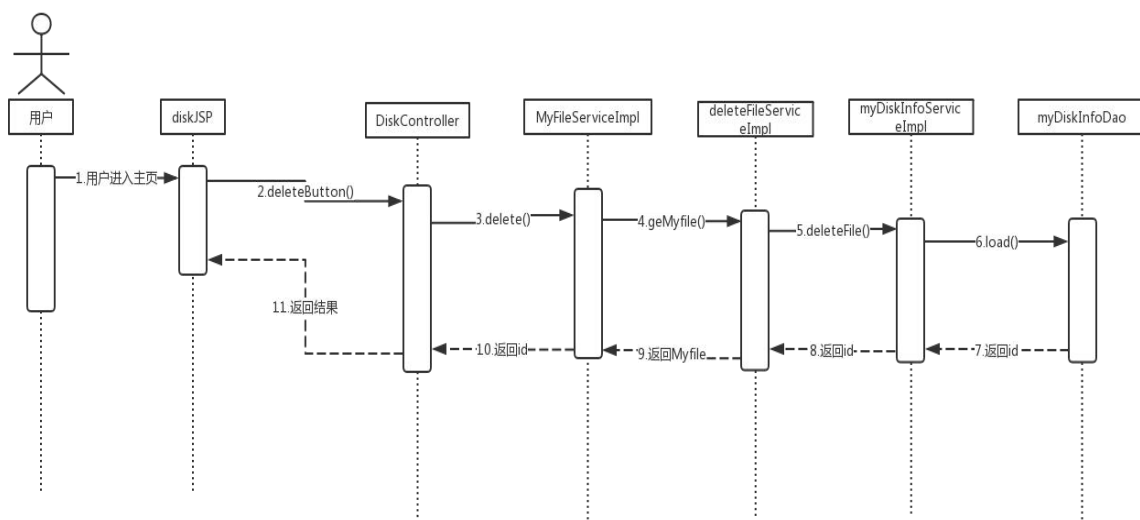


图 3.5 文件删除功能顺序图

3.5.4 文件或文件夹拖拽用例实现

拖拽文件或文件夹功能，用户启动系统之后，登陆完成进入主界面，点住目标文件或文件夹拖拽文件，使用了开源工具包，顺序图如图 3.6 所示。

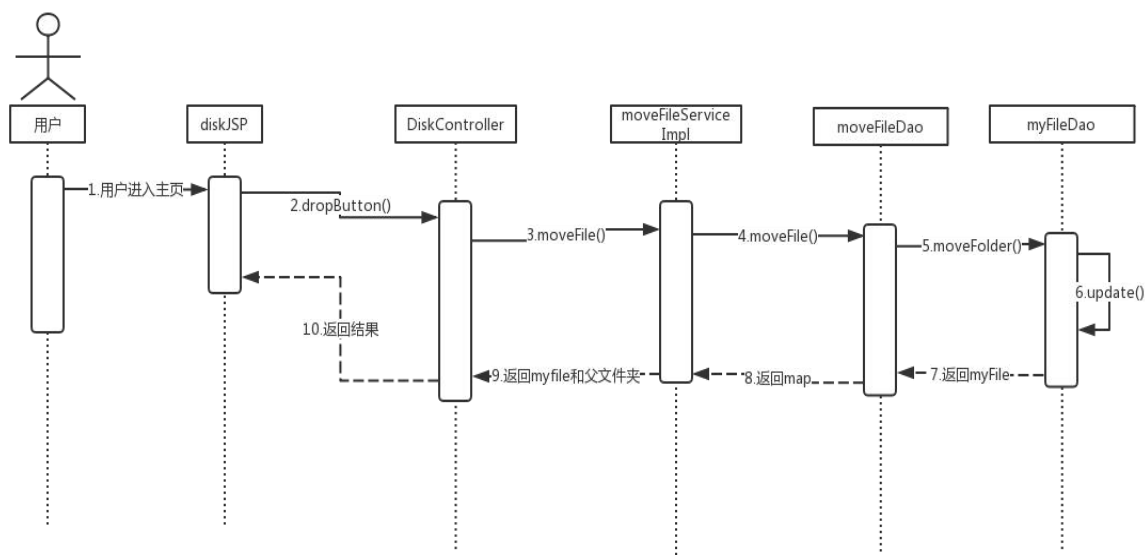


图 3.6 文件或文件夹拖拽功能顺序图

3.5.5 文件加密功能用例实现

文件夹加密功能，用户在登陆到系统之后，进入主界面，右键点击目标文件夹可以对其进行加密操作，其顺序图如图 3.7 所示。

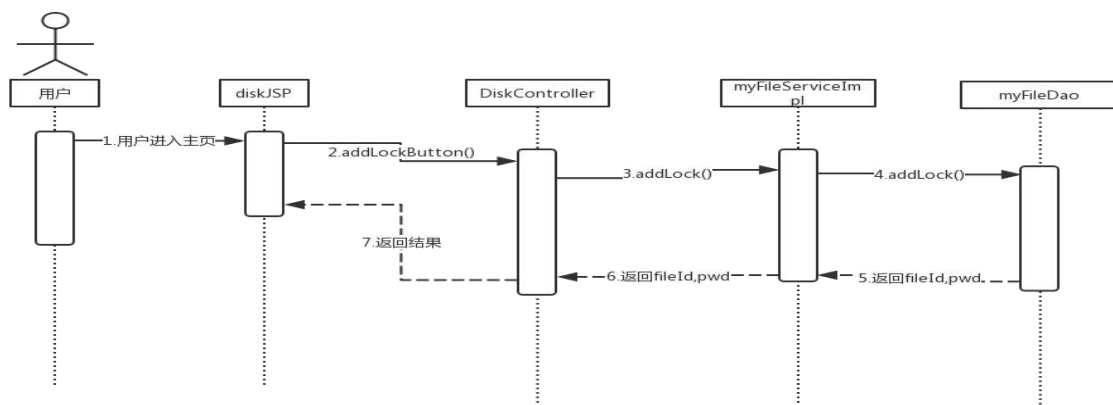


图 3.7 文件加密功能顺序图

3.6 对象设计

3.6.1 模块设计

用户是本软件功能的主要使用者，包括文件上传，文件下载，文件夹拖拽，文件及文件夹重命名等功能。具体功能如图 3.8 所示。

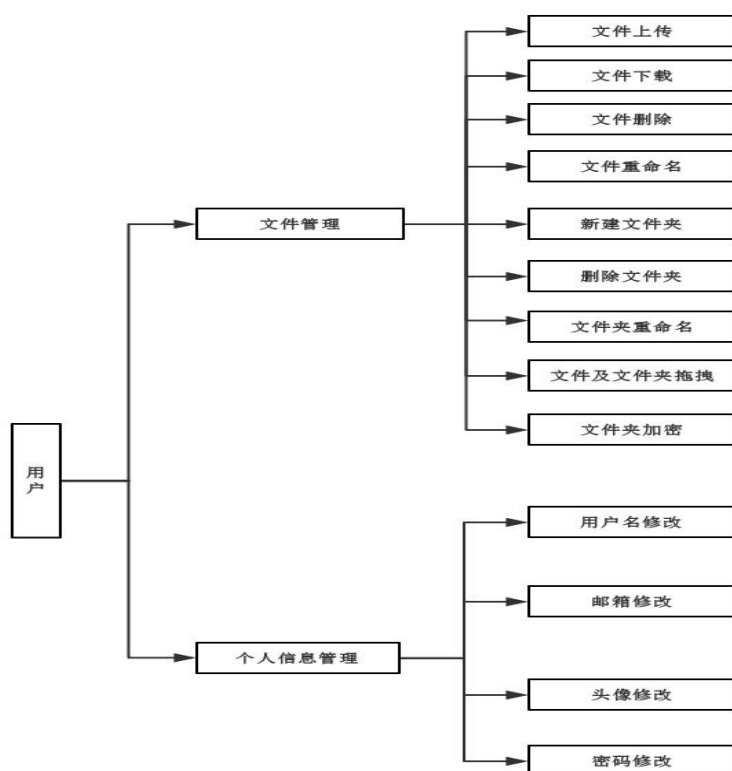


图 3.8 用户功能结构图

3.6.2 业务实体设计类图

业务实体包括软件所有业务中涉及到的独立实体，数据访问层，顾名思义就是与数据库打交道。对每个业务逻辑层中的类都设计了对应的数据访问的接口，分别为 DeleteDao、MessageDao、MoveFileDao、MyDiskDao、MyFileDao、RegisterDao、UpLoadDao、 UserDao 接口。jdbc.properties 管理了访问数据库的属性；每个数据访问层的接口都必须有一个实现功能的 XML 文件，才能进行对数据库中的数据进行操作。数据访问层的各个接口如图 3.9 所示。

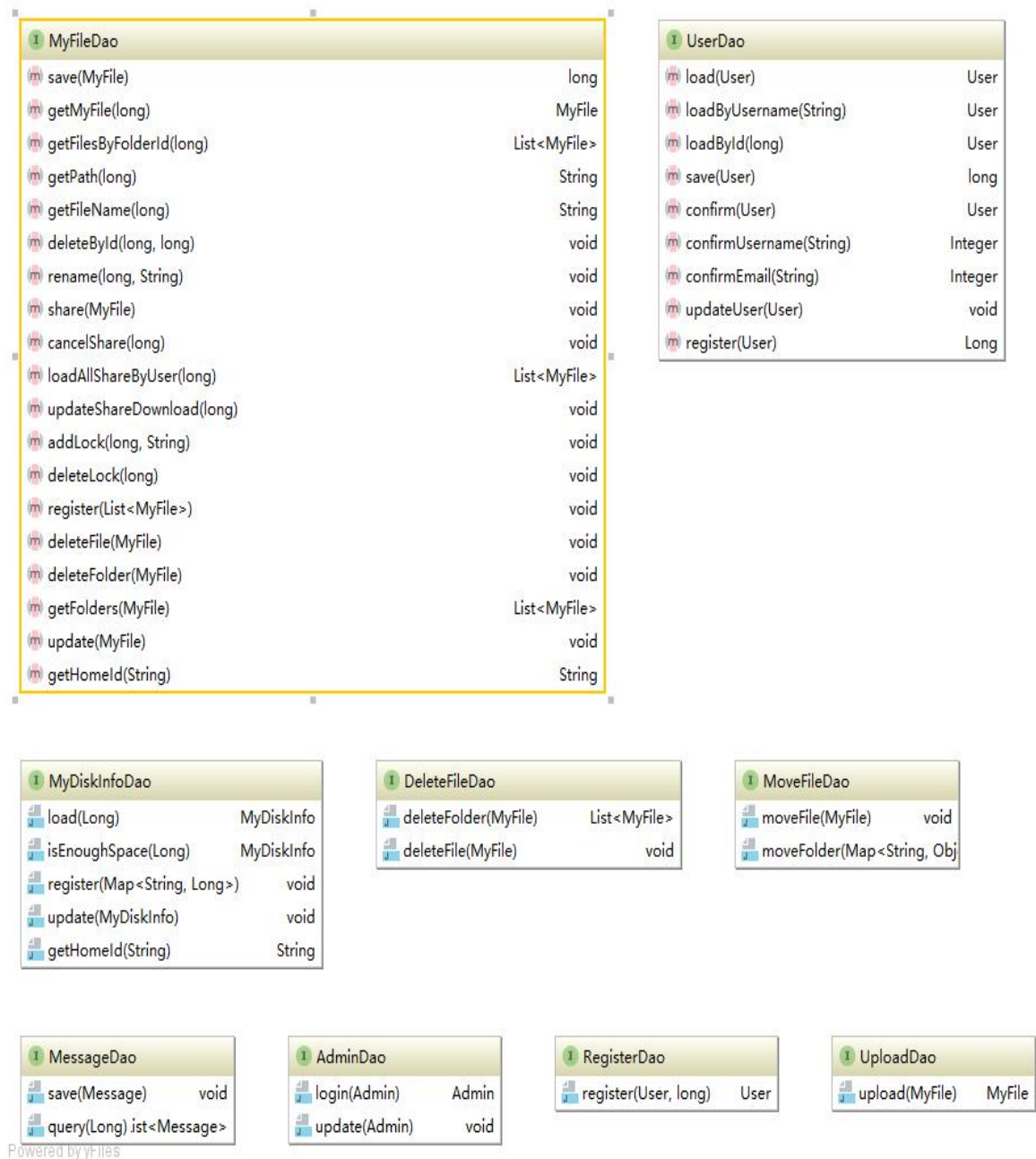


图 3.9 业务实体图

3.6.3 业务对象设计类图

(1) Controller 控制层类图

控制层中分别设计了 DiskController、LoginController、RegisterController、ShareController、UserSettingController 分别对各个业务逻辑的控制类。控制层各个类之间的关系如图 3.10 所示。

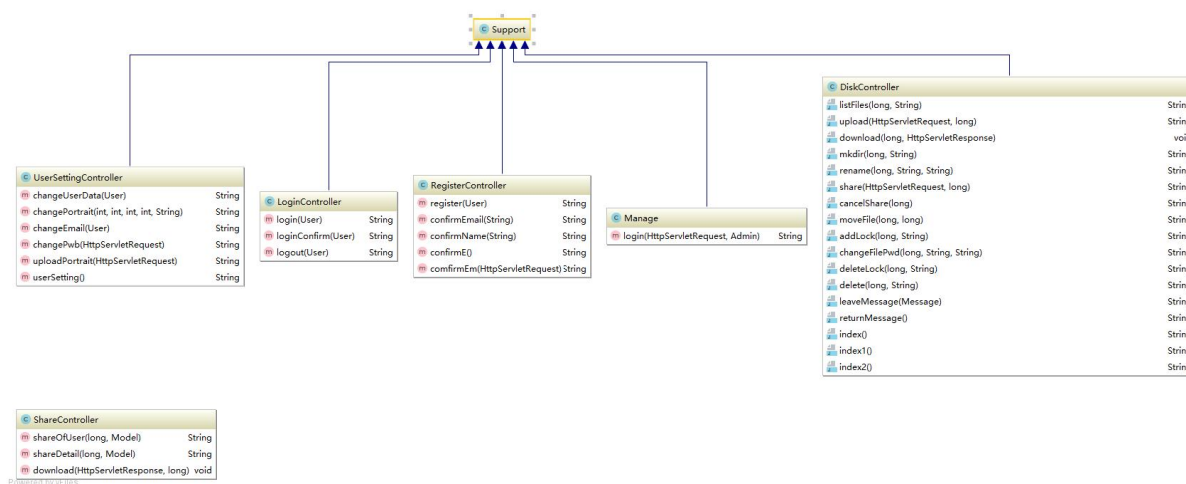


图 3.10 Controller 层控制类图

(2) Service 业务逻辑层类图

业务逻辑层中对控制层中的 Controller 设计了相应的业务逻辑实现类，分别为 DeleteServiceImpl、MessageServiceImpl、MoveFileServiceImpl、MyDiskInfoServiceImpl、MyFileServiceImpl、RegisterServiceImpl、UpLoadServiceImpl、UserServiceImpl 类。所有类都实现了相对应的 Service 接口。业务逻辑层系如图 3.11 所示。

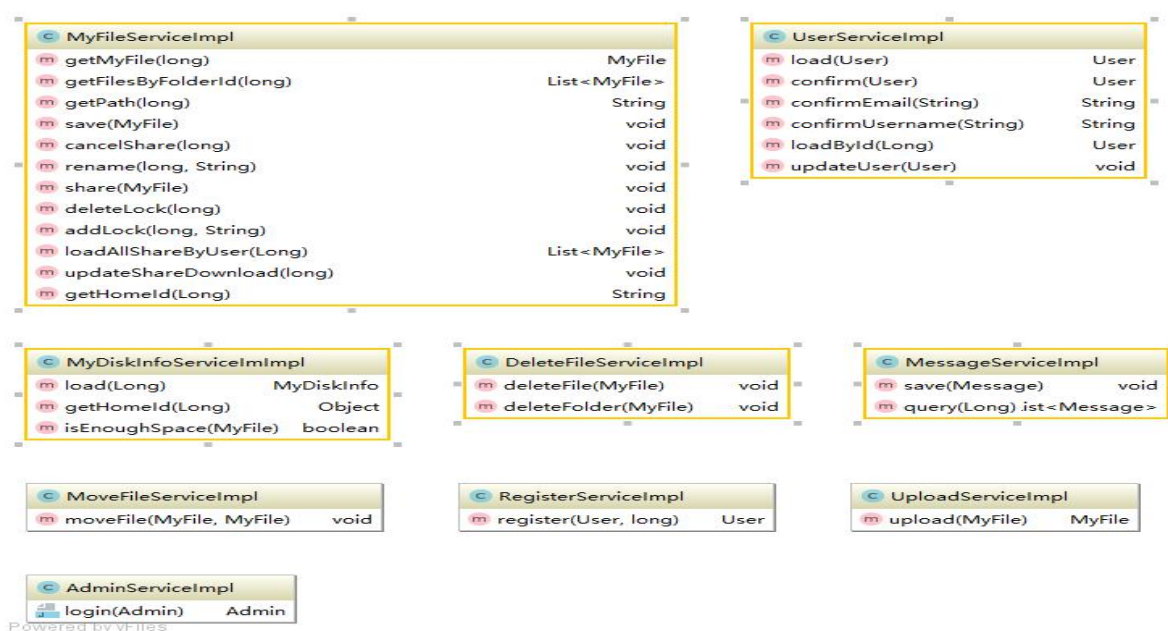


图 3.11 Service 业务逻辑层类图

第 4 章 系统实现

4.1 核心功能实现

4.1.1 文件夹加密/解密模块

因为本系统要为用户的文件安全性考虑，所以需要对文件夹进行加密，这样可以使得用户的文件信息更加安全，其功能演示如图 4.1 所示。

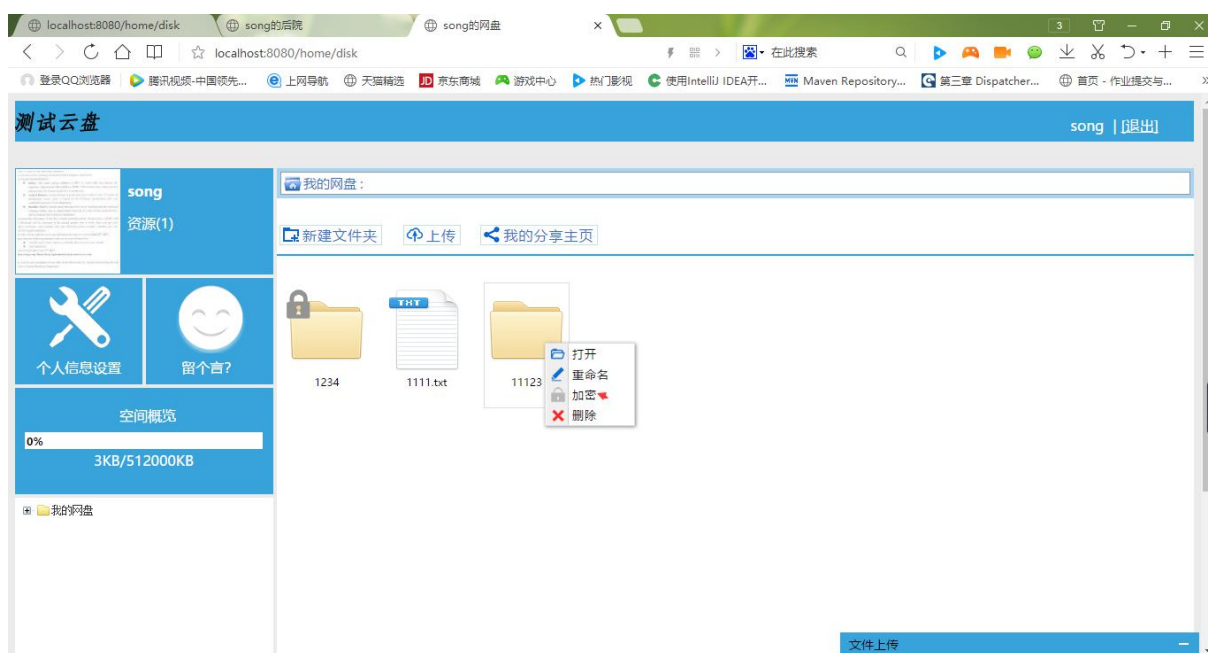


图 4.1 文件加密功能演示图

加密时密码长度要大于或等于 4 位，否则安全性会降低，设置文件密码和加密后的文件状态如图 4.2 所示和图 4.3 所示。

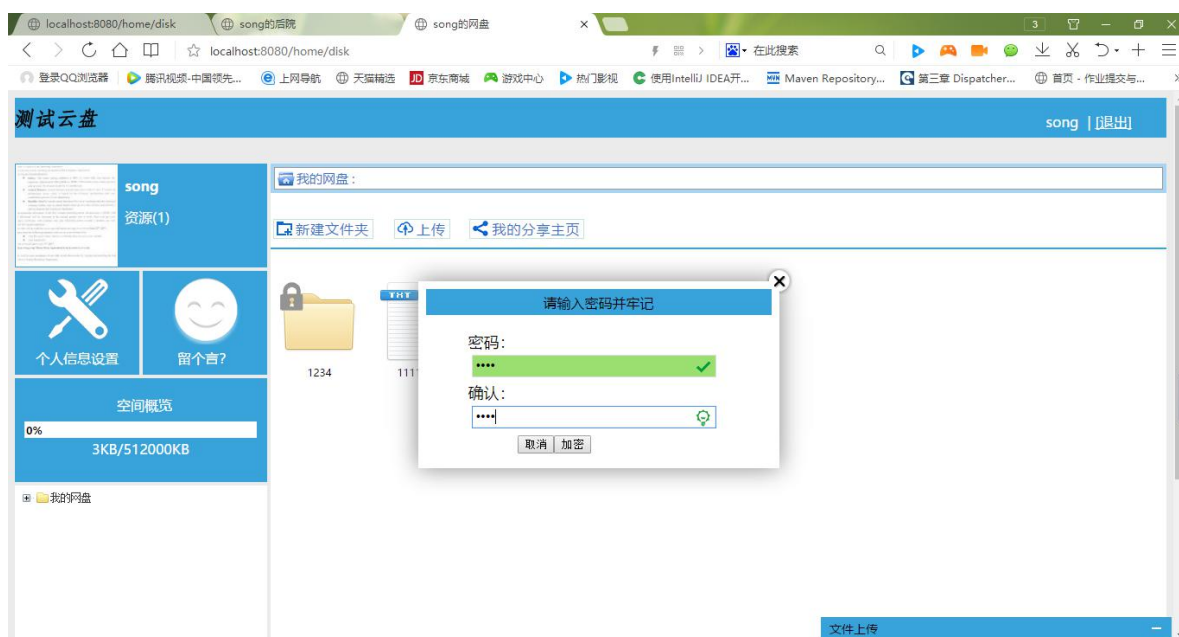


图 4.2 文件加密功能演示图

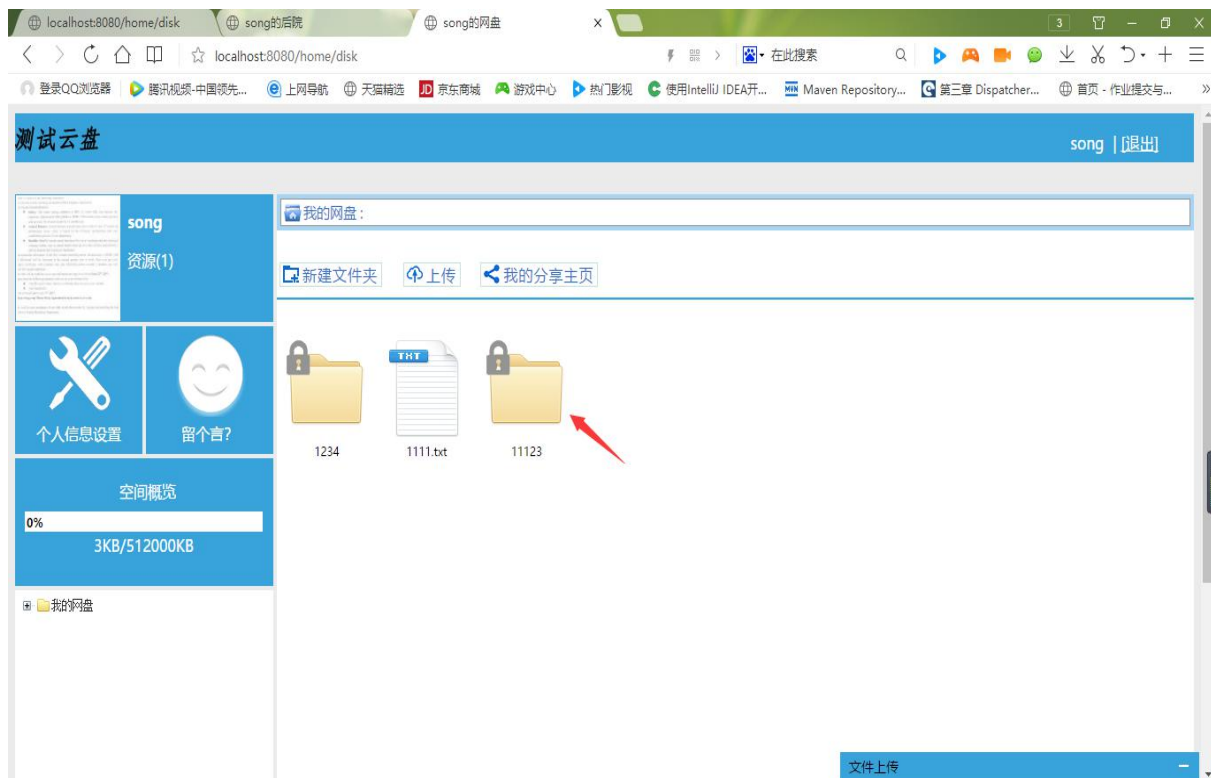


图 4.3 文件加密功能演示图

如果我们想给文件夹取消加密功能，只需要右键点击目标文件夹然后点击删除密码，随后输入之前设计的文件夹密码后即可完成删除，演示如图 4.4 和图 4.5 所示。

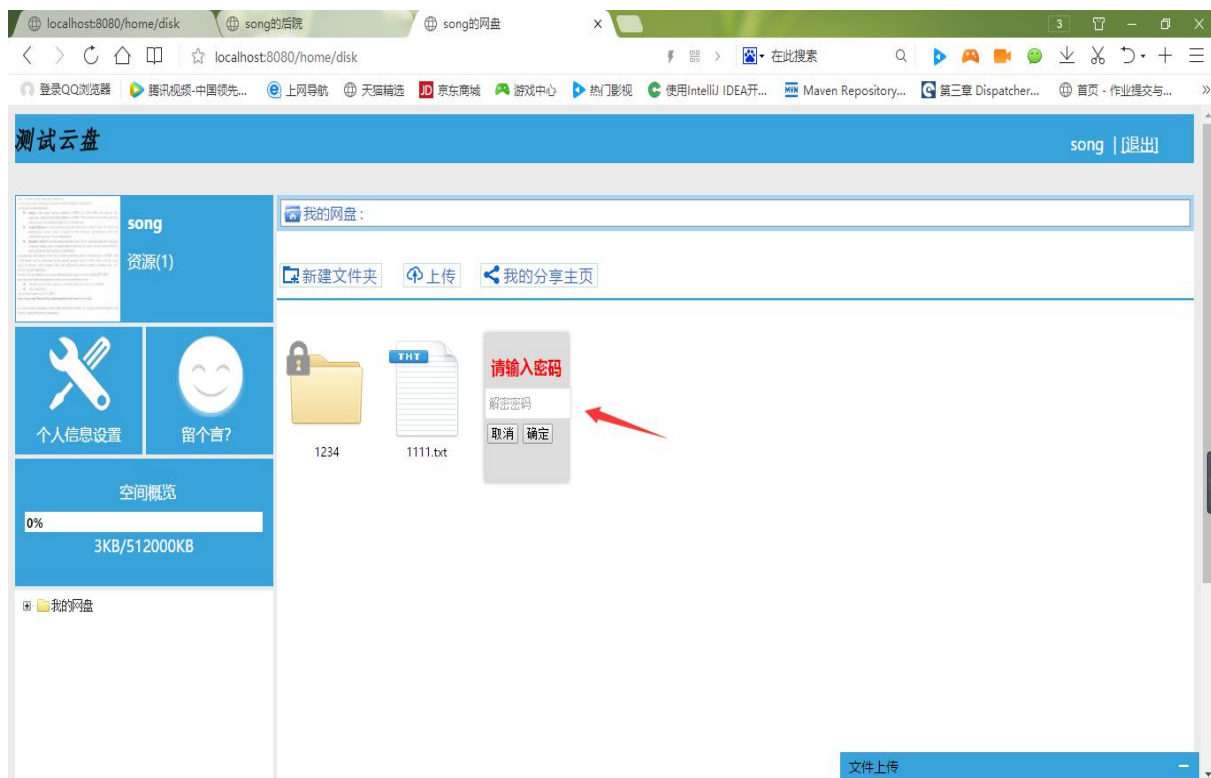


图 4.4 文件加密功能演示图

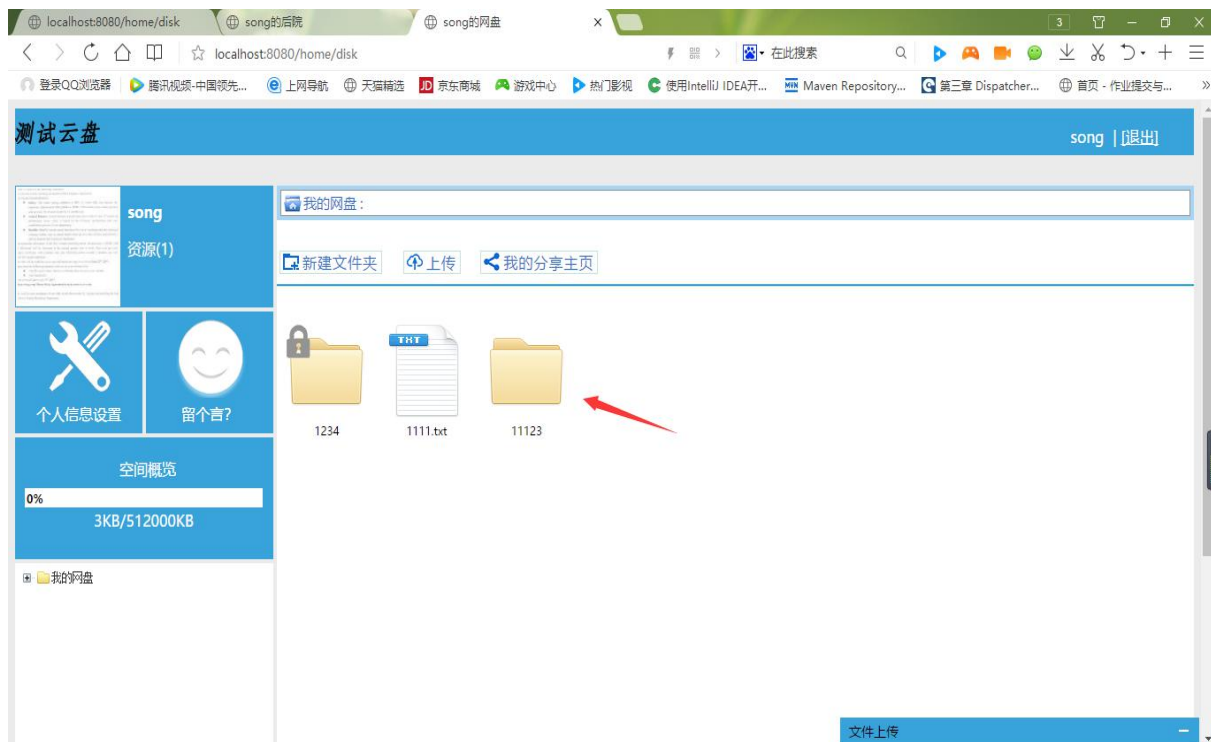


图 4.5 文件加密功能演示图

用户也可以修改文件夹的密码，右键点击目标文件夹后选择更改密码，输入旧密码和想要改的新密码后即可完成密码的修改，演示如图 4.6 和图 4.7 所示。

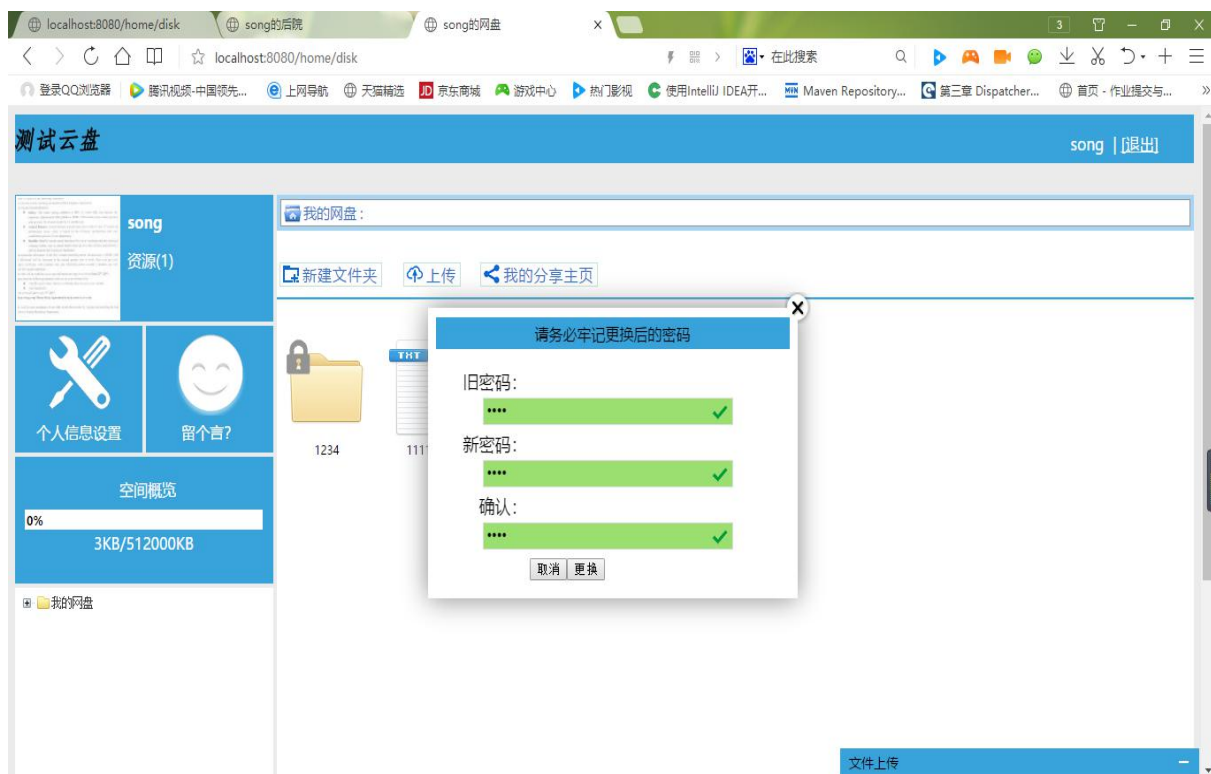


图 4.6 文件加密功能演示图

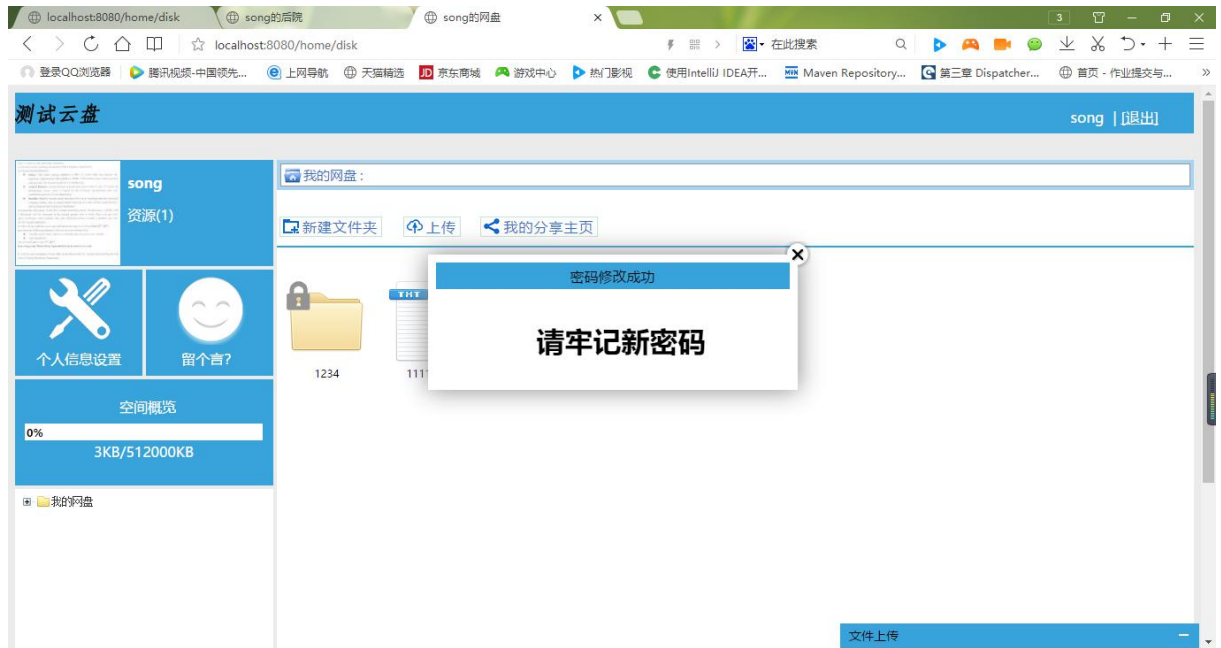


图 4.7 文件加密功能演示图

其相关代码如下；

```
@RequestMapping("/addlock/{fileId}")
@ResponseBody
public String addLock(@PathVariable long fileId, String pwd) {
    MyFile myFile = myFileService.getMyFile(fileId);
    if (myFile.getIsLock() != 1) {
        myFileService.addLock(fileId, pwd);
    }

    return "success";
}
```

其对应的 serviceImpl 代码如下；

```
@Override
public void addLock(long fileId, String pwd) {
    // TODO Auto-generated method stub
    myFileDao.addLock(fileId, pwd);
}
```

其改变文件夹密码的代码如下：

```
public String changeFilePwd(@PathVariable long fileId, String oldPwd, String newPwd) {
    MyFile myFile = myFileService.getMyFile(fileId);
    String result = "fail";
    if (myFile.getPassword() == null && oldPwd == "" ||
        myFile.getPassword().equals(oldPwd)) {
        myFileService.addLock(fileId, newPwd);
        result = "success";
    }
    return result;
}
```


4.1.2 头像上传功能模块

用户可以通过上传自己想要作为图像的图片，然后截取一部分作为头像，其功能演示如图 4.8 所示。

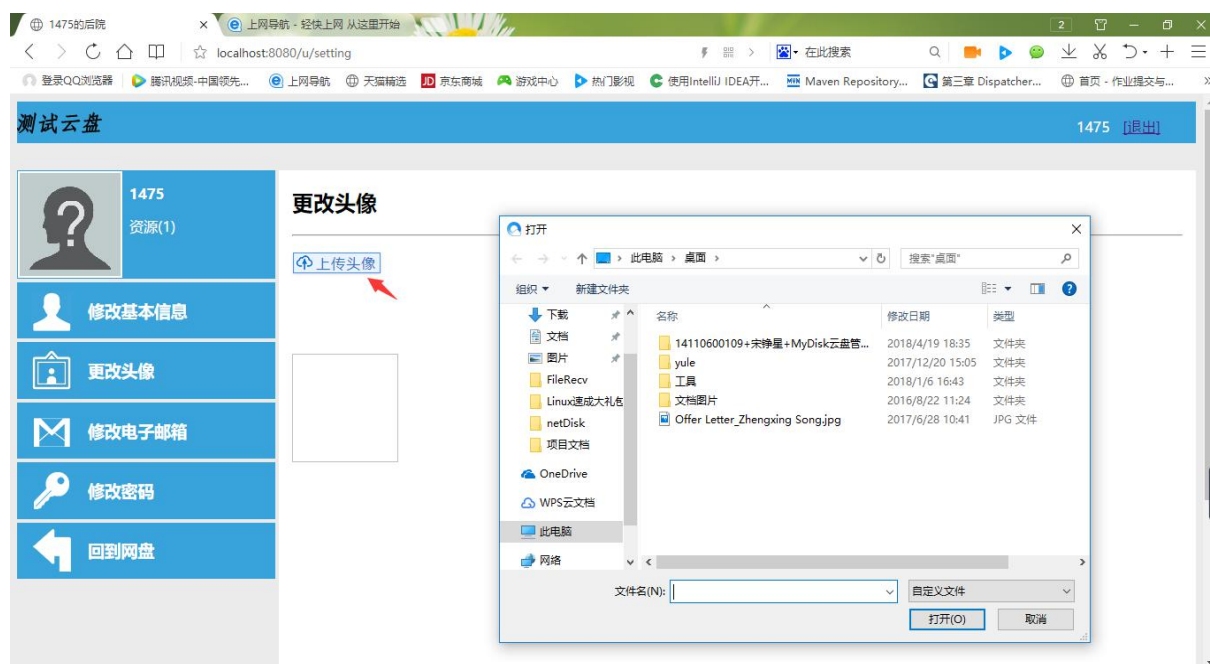


图 4.8 头像上传功能演示图

点击上传头像即可选择电脑中已有的图片进行图片上传，然后用户即可进行已上传的图片的截取，如图 4.9 所示。

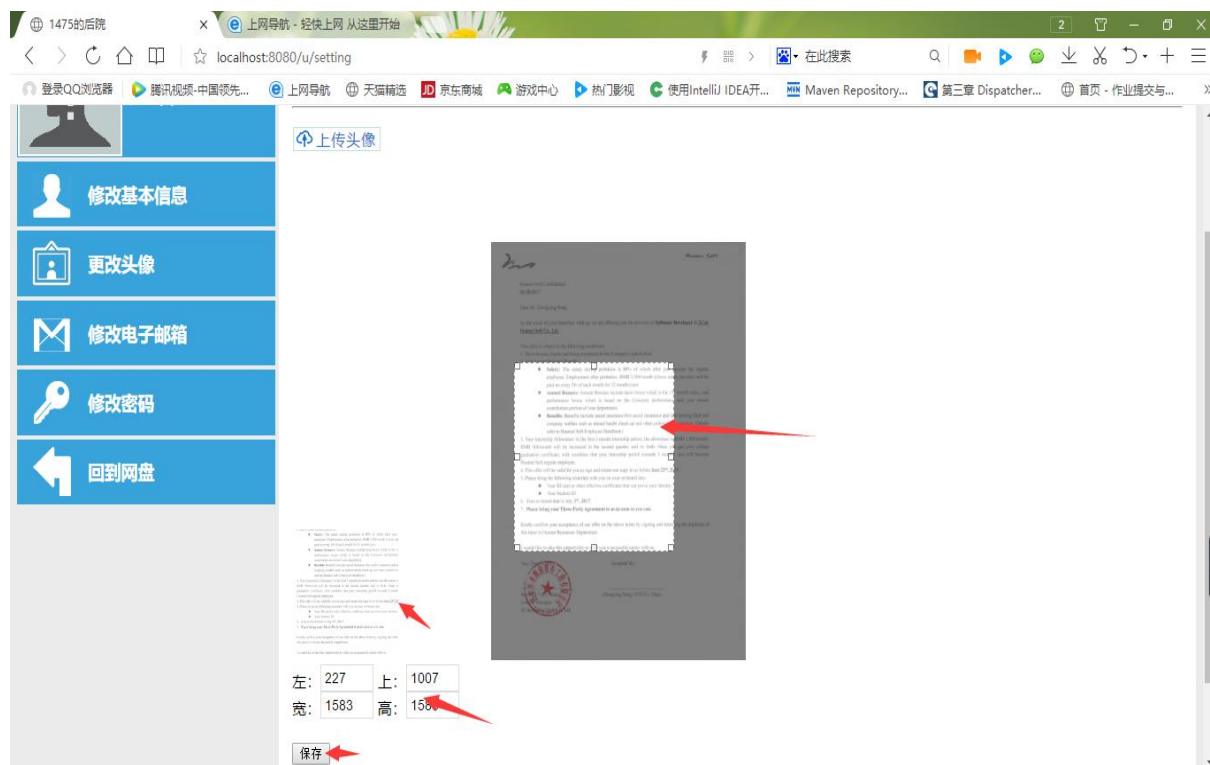


图 4.9 头像上传功能演示图

截取图片后保存即可使用该图片做自己的头像如图 4.10 所示。

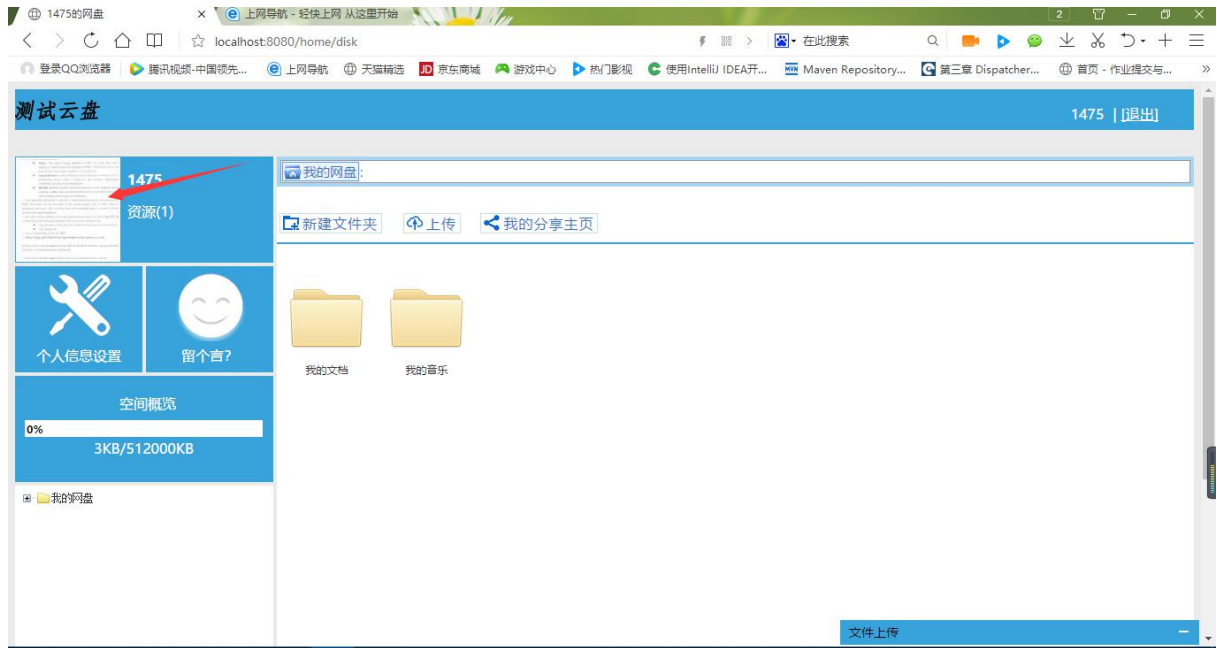


图 4.10 头像上传功能演示图

其相关代码如下；

```
public String changePortrait(int x,int y,int width,int height,String imgName){
    User user = (User) session.getAttribute("user");
    if(imgName.trim().equals("")) return "";
    String realPath = session.getServletContext().getRealPath("/");
    ImgTool imgT = new ImgTool();
    imgT.cut(realPath+UPLOADPATH+imgName, x, y, width, height);
    imgT.save("jpg", user.getId()+"_jpg", realPath+PORTRAITPATH, 120, 120);
    if(user.getPortrait().equals("portrait")){
        user.setPortrait(user.getId()+"");
        userService.updateUser(user);
    }
    return user.getId()+"_jpg";
}
```

其使用的工具类以及上传图片的代码如下所示；

```
public String uploadPortrait(HttpServletRequest request){
    UploadHelper upload = new UploadHelper();
    MultipartFile file = upload.getFiles(request).get(0);
    User user = (User) session.getAttribute("user");
    try {
        File file2=new File(realPath);
        //如果文件夹不存在则创建
        if (!file2.exists() && !file2.isDirectory())
        {
            System.out.println("//不存在");
            file2.mkdirs();
        }
        upload.upload(file, realPath+"/"+imgName);
    }
```

```

    } catch (IOException e) {
        e.printStackTrace();
    }
    return imgName;
}

```

4.1.3 文件上传/下载功能模块

用户登陆进入主页面后，即可使用该操作系统完成相关的文件或文件夹操作，文件上传功能模块如图 4.11 所示。

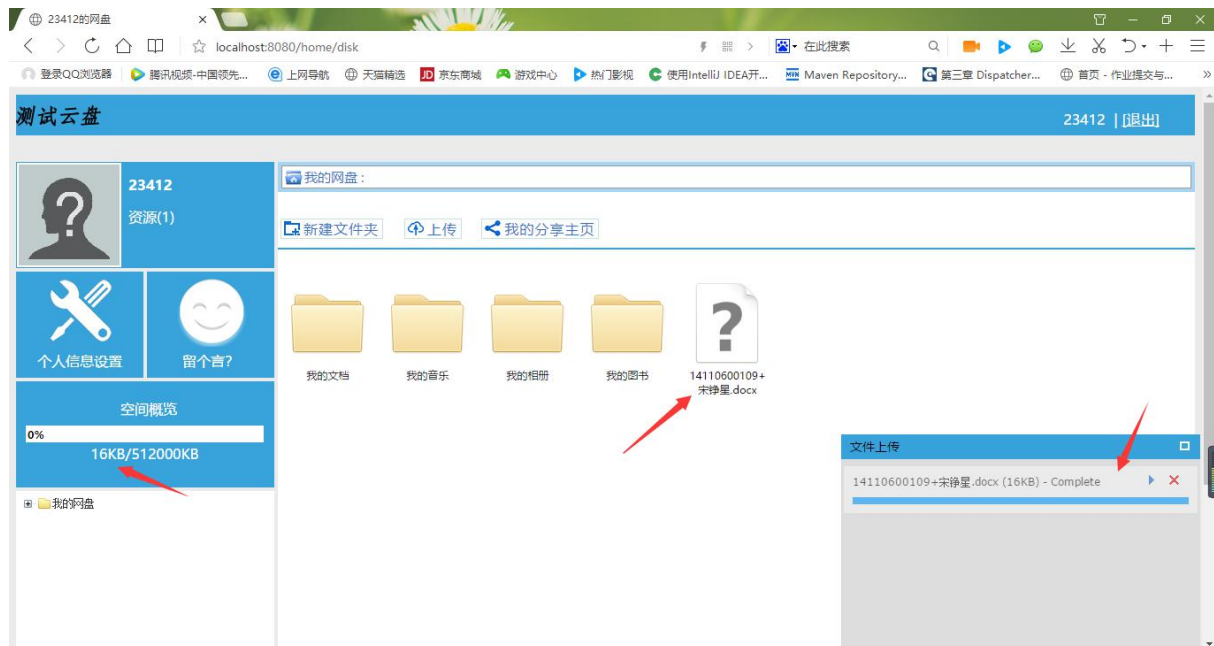


图 4.11 文件上传功能演示图

上图中有红色箭头的地方均为完成上传操作后主页面的变化，主要改变有，新上传的文件出现在根目录中，云盘当前剩余容量有了变化。其代码实现如下。

Controller 层文件上传代码如下：

```

if (myDiskInfoService.isEnoughSpace(myFile)) {
    String filePath = FILEBASEPATH + new Date().getTime() + "." + suffix;
    try {
        upload.upload(file, filePath); // 文件没有成功保存返回失败信息
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        myFile.setCreateDate(sdf.format(new Date()));
        myFile.setName(fileName);
        myFile.setParent_id(folderid);
        myFile.setType(suffix.toLowerCase());
        myFile.setPath(myFileService.getPath(folderid) + folderid + "/");
        myFile.setLocation(filePath);
        myFile.setIsShare(0);
        myFile.setDescription("");
        myFile = uploadService.upload(myFile);
        /* 同步网盘信息 */
    }
}

```


MyDiskInfo diskInfo = myDiskInfoService.load(user.getId());
 session.setAttribute("diskInfo", diskInfo);
 下载文件页面如图 4.12 和图 4.13 所示，右键点击目标文件即可完成下载。

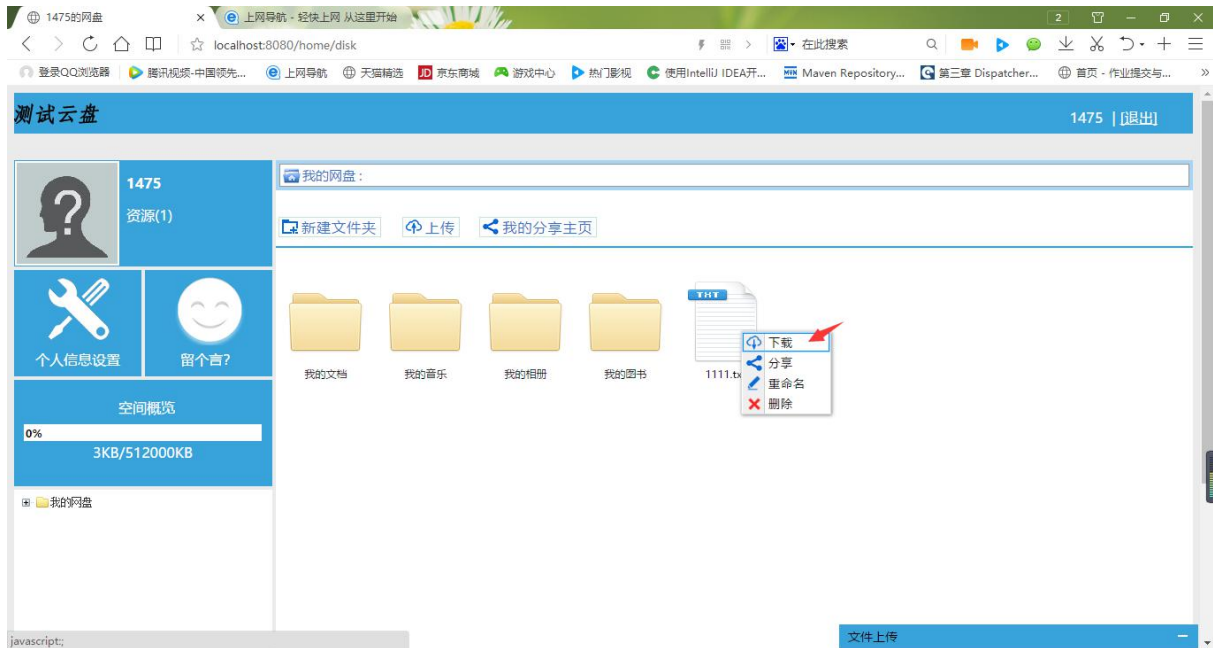


图 4.12 文件下载功能演示图

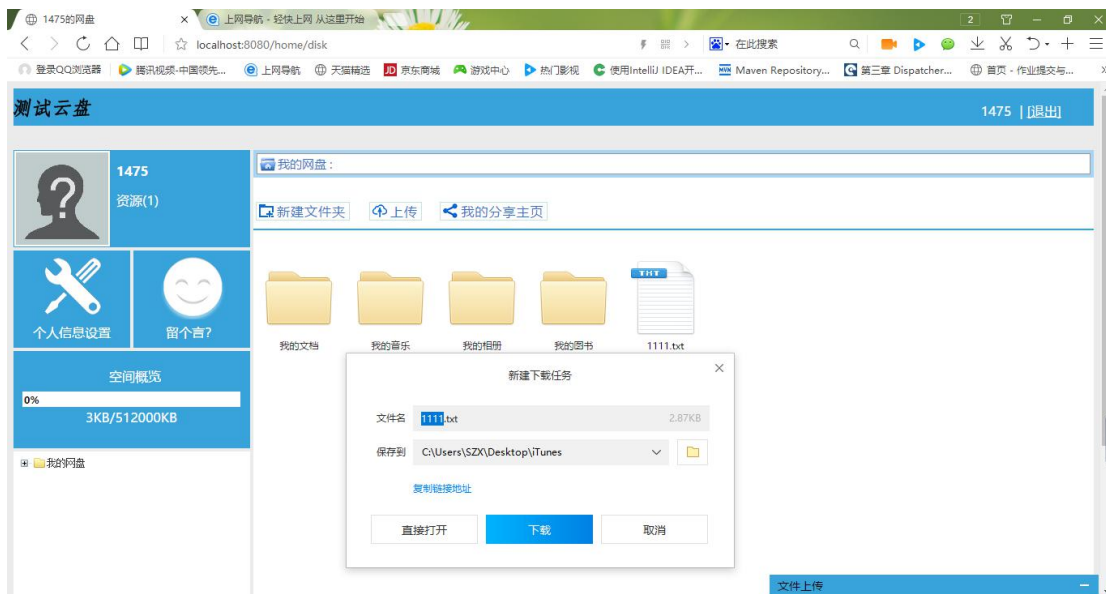


图 4.13 文件下载功能演示图

其相关代码如下：

Controller 层文件下载代码如下：

```
@RequestMapping("/download/{fileId}")
public void download(@PathVariable long fileId, HttpServletResponse response) {
    MyFile myFile = myFileService.getMyFile(fileId);
    DownloadSupport.download(response, myFile);
}
```

相关 ServiceImpl 类中代码如下：

```
public MyFile getMyFile(long id) {  
    // TODO Auto-generated method stub  
    return myFileDao.getMyFile(id);  
}
```

4.1.4 文件/文件夹拖拽功能模块

在该系统中，文件和文件夹之间可以实现相互包裹（拖拽）功能，此功能可以方便用户对文件进行编排，方便文件的堆放，不会使文件杂乱无章，其功能演示如图 4.14 所示。

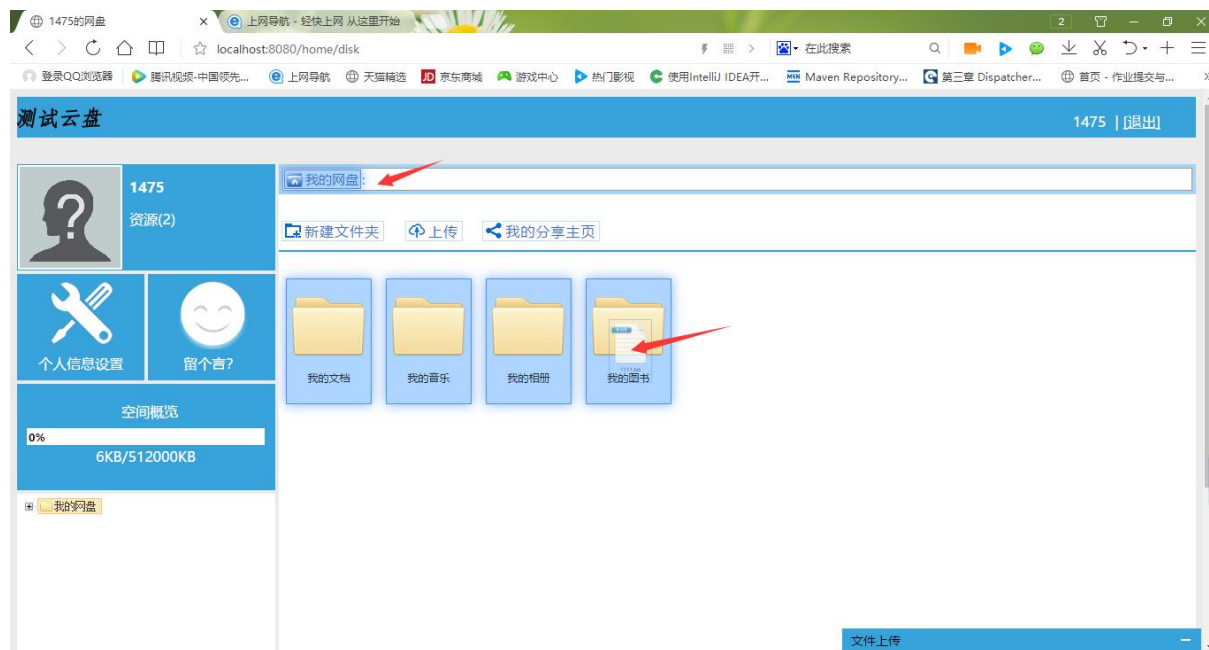


图 4.14 文件/文件夹拖拽功能演示图

如上图所示，拖拽目标文件到想放置的文件夹图标上方即可完成文件的拖拽，拖拽完成后的页面变化如图 4.15 所示。

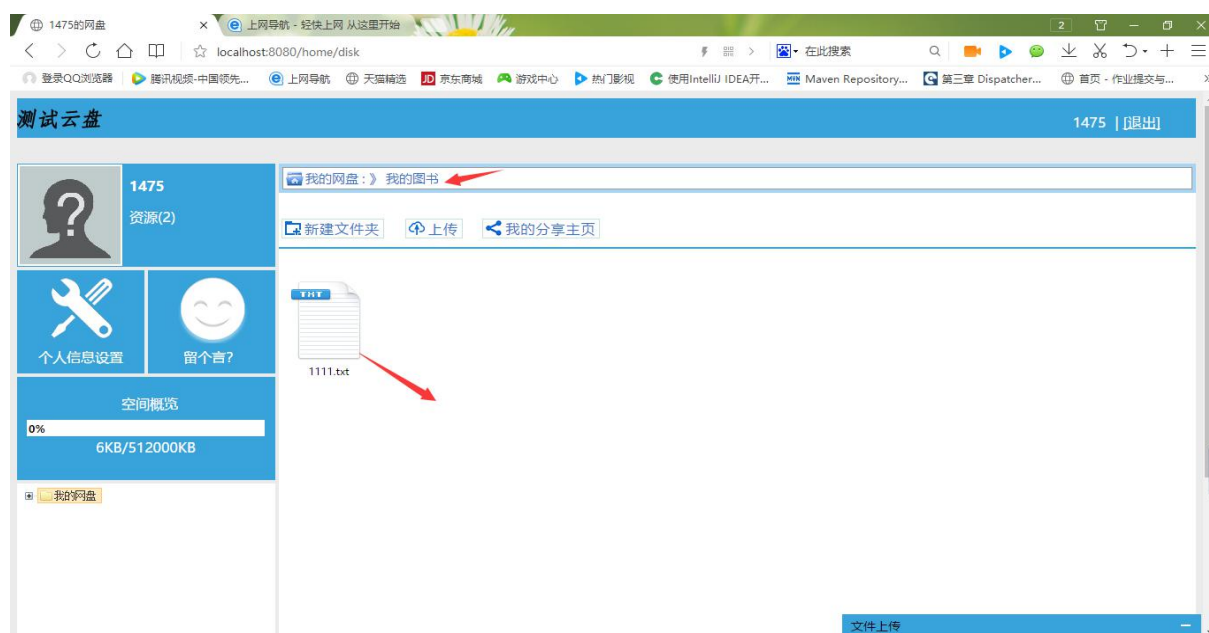


图 4.15 文件/文件夹拖拽功能演示图

在“我的图书”文件夹目录下已经出现了咱们想要拖拽的文件，这就完成了文件夹的拖拽，文件夹的拖拽如图 4.16 和图 4.17 所示。

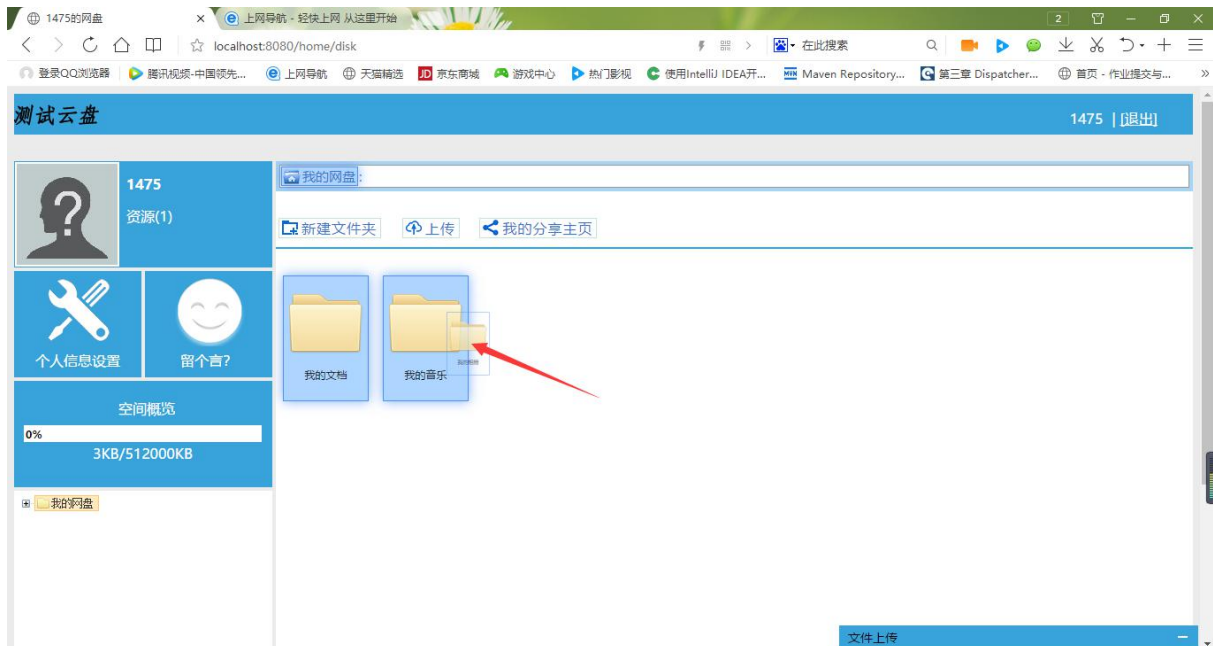


图 4.16 文件/文件夹拖拽功能演示图

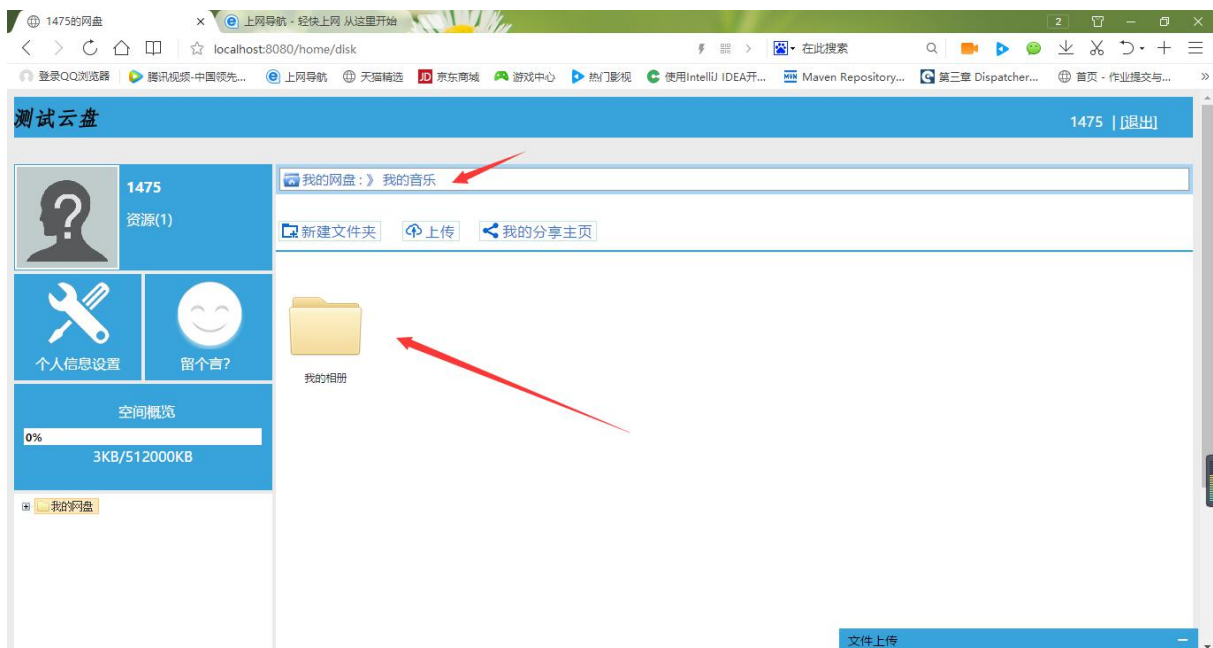


图 4.17 文件/文件夹拖拽功能演示图

相关文件/文件夹拖拽的代码如下：

```
@RequestMapping("/movefile")
@ResponseBody
public String moveFile(long sourceId, long targetId) {
    MyFile sourceFile = myFileService.getMyFile(sourceId), targetFile =
    myFileService.getMyFile(targetId);
    moveFileService.moveFile(sourceFile, targetFile);
}
```

```

    return "success";
}

```

其对应的 serviceImpl 的代码如下；

```

public void moveFile(MyFile sourceFile, MyFile targetFile) {
    MyFile myFile= new MyFile();
    myFile.setId(sourceFile.getId());
    myFile.setPath(targetFile.getPath());
    myFile.setParent_id(targetFile.getId());
    if(sourceFile.getType().equals("adir")){
        Map<String,Object> map = new HashMap<String,Object>();
        map.put("pathSource", sourceFile.getPath());
        map.put("pathSourceet", sourceFile.getPath()+"%");
        map.put("pathtargetFile", targetFile.getPath());
        moveFileDao.moveFolder(map);
        myFileDao.update(myFile);
    }else{
        moveFileDao.moveFile(myFile);
    }
}
}

```

4.2 关键技术难点与解决方案

通过前期需求分析、软件设计和系统框架的整理，在对整个软件业务比较熟悉的情况下，总结出本软件的技术难点在于如何实现文件夹树结构，文件及文件夹拖拽，用户头像截取，上传和下载等比较复杂的功能。为了能够解决这些比较难的功能，我在各大开源网站上寻找相关的代码，并且深入研究了 JQuery 框架和其相关的插件，在此期间多和指导老师彭志豪进行沟通，受益匪浅，终于问题再老师的帮助和自己不断地翻阅资料，寻找开源代码下得到了解决。

第 5 章 系统测试

5.1 功能测试

本章主要通过黑盒测试方法对软件进行功能测试。本章测试会根据系统功能不同分为文件相关操作模块、个人信息修改模块，通过模块的划分分别写出对应的测试用例并测试得出测试结果。

5.1.1 文件相关操作模块测试用例

文件相关操作测试用例，因为本系统主要是和文件打交道，所以文件及文件夹相关的测试尤为重要，该测试模块需要高度重视，尤其是云盘必不可少的重要功能需要进行多次测试，如文件的上传和下载，文件和文件夹的拖拽，文件夹的加密和解密等，具体测试用例如下表 5.1 所示。

表 5.1 文件相关操作测试用例表

编号	用例描述	输入值	期望结果	实际结果	P/F
1	点击上传按钮	选择想上传的文件	出现上传的文件，网盘容量增加	与期望值一致	P
2	点击下载按钮	右键选择想要下载的文件	选择要下载的文件下载	与期望值一致	P
3	右键点击文件后删除	无	文件被删除，容量变化	与期望值一致	P
4	右键点击文件后重命名	输入文件新名称	文件名称改变	与期望值一致	P
5	右键点击文件夹后加密	输入加密密码	文件夹被加密	与期望值一致	P
6	文件及文件夹拖拽	无	文件及文件夹拖拽成功	与期望值一致	P
7	新建文件夹	输入文件夹名称	文件夹创建成功	与期望值一致	P
8	删除文件夹	无	指定文件夹在提示用户后被删除	与期望值一致	P
9	文件夹重命名	输入新名称	目标文件夹被重命名	与期望值一致	P

5.1.2 个人信息修改模块测试用例

个人信息修改模块功能测试用例，该用例主要是对用户的一些修改个人信息的功能进行测试，具体测试用例如下表 5.2 所示。

表 5.2 个人信息修改模块测试用例表

编号	用例描述	输入值	期望结果	实际结果	P/F
1	修改基本信息	无	在个人中心页面显示可修改的基本信息包括用户昵称，性别，点击保存后修改成功	与期望值一致	P
2	点击“更改头像”	上传新头像并截取想要的部分	上传相应的图片后即可截取想要的图片，点击保存后即可使用该图片作为自己的头像	与期望值一致	P
3	修改电子邮箱	获取文本框中的值	输入新的 Email 地址后保存即可使用新的邮箱	与期望值一致	P
4	修改密码	输入新密码	输入新的密码后即可使用新的密码	与期望值一致	P

5.3 测试总结

本章最主要的是对本系统进行软件测试，它是保证软件质量的关键步骤。测试的目的是以最少的人力物力和时间找出软件中潜在的错误和缺陷。从整体的角度来看，动态测试可分为两种，分别是黑盒测试和白盒测试。此次测试将采用黑盒测试，列举部分模块的详细测试用例，尽可能来检验软件的正确性和准确性，并根据测试结果进行修复和改进。通过前期的需求分析、软件设计和编码阶段的开发后，需要对软件进行单独的测试，以此来发现在前期工作中未被发现的错误和缺陷，以便编码人员及时更正，以免软件在使用过程中出现重大 BUG。所以软件测试仍是发现软件中错误和缺陷的主要手段，也是软件设计中不可缺少的一部分。

第 6 章 结论与展望

MyDisk 云盘管理系统利用 Maven 进行依赖管理,服务端使用当前最流行的 SpringMVC 和 Mybatis 框架和简单方便的 MySQL 数据库来完成设计与开发。该系统主要参考各大云盘系统完成的,通过不断的学习研究最终达到了预期效果,以实现了云盘应有的大部分功能,由于现在市场上流通的成熟的云盘系统比较复杂,所以有一些比较尖端的功能没有实现,但出于客户体验,该云盘操作简洁,容易上手,使得客户用起来更加得心应手。

MyDisk 云盘管理系统实现了用户登录与注册功能,登录系统后可以对自己的云盘进行一系列的管理:上传和下载文件,删除不需要的文件,新建和删除文件夹以及文件和文件夹的重命名,这些都是作为一个云盘管理系统应该实现的基本功能;为了方便用户可以使用该系统更加便捷的管理自己的文件和文件夹,添加了文件和文件夹的拖拽功能,为了用户的信息安全,添加了文件夹加密/解密功能。用户信息设置方面,用户可根据自己的需要对自己的密码,邮箱,性别,头像进行修改,整体是一个比较人性化好操作的管理系统。

在该系统的开发过程中,遇到过许多棘手的问题,如用户头像上传的时候,无法对图像进行截取,造成用户无法使用称心的头像,还有后台程序设计等问题。然后在各方面的努力和帮助下问题又一个个被顺利解决,整个过程就是一个不断学习与进步的过程,每次发现问题、解决问题都是一个宝贵的财富,积累了经验,学到了新知识。虽然现在成功完成了该软件系统,但由于时间以及各方面的影响,该云盘管理系统还有一些不足与缺陷,如界面的设计、用户体验以及性能的改善等等。对于这些不足之处,我将在以后的学习中不断来对它进行改进和更新。

参考文献

- [1] 宋艳红,程乐.基于云盘技术的高校档案管理系统设计[J].信息与电脑(理论版), 2017, 97-99.
- [2] 罗时飞.精通 Spring[M].北京: 电子工业出版社, 2015, 55-78
- [3] 宋丽艳.个人云盘在勘探开发图档管理系统中的应用[J].内蒙古石油化工, 2016, 90-92.
- [4] 李婉茹. 乐视云分享管理系统的设计与实现[D].北京交通大学, 2015, 30-37.
- [5] 王伟.浅析云盘技术及存储原理[J].电脑知识与技术, 2015, 218-224.
- [6] ZHANG Shan,LI Haibo,CHEN Xue,DONG Yi,ZHANG Fang,XIAO Tian,ZHANG Wuchang,ZHAO Yuan.Differences in planktonic ciliate spatial distribution in spring and autumn in the southern Yellow Sea[J].Acta Oceanologica Sinica, 2018, 48-57.
- [7] 王明鑫, 孙仲霖.简述 Java 编程语言的发展及优势[J].通讯世界, 2016, 88-90.
- [8] 杨旭.JAVA 编程语言在计算机软件开发中的应用[J].电子技术与软件工程, 2018, 59-60.
- [9] 惠强.计算机软件开发中 JAVA 编程的应用[J].电子技术与软件工程, 2018, 35.
- [10] 王旭.软件开发中 JAVA 编程的应用[J].中国管理信息化, 2018, 174-176.
- [11] 黄俊勇.基于 Spring 框架的研究与设计[J].电脑知识与技术, 2018, 116-117.
- [12] 姚云飞,杜洪波,梁建辉.基于 SpringMVC 框架毕业设计管理系统设计[J].软件, 2018, 91-93.
- [13] Shuang YU,Jiangjiang XIA,Zhongwei YAN,Kun YANG.Changing Spring Phenology Dates in the Three-Rivers Headwater Region of the Tibetan Plateau during 1960–2013[J].Advances in Atmospheric Sciences, 2018, 116-126.
- [14] 焦正洋.基于 Java 技术的纺织企业 ERP 系统开发[J].信息技术与信息化, 2016, 10-45.
- [15] 乔岚.基于 MyBatis 和 Spring 的 JavaEE 数据持久层的研究与应用[J].信息与电脑(理论版), 2017, 73-76.
- [16] 邹红霆.基于 SSM 框架的 Web 系统研究与应用[J].湖南理工学院学报(自然科学版), 2017, 39-43.

致 谢

当自己还沉浸在校园时光的时候，学生二字带给我的永远是无忧无虑，悠然自得的惬意。直到此篇论文接近尾声之时，我倏然明白，自己的学生时代也将在不久后画下句点。回首过往，在东软四年的青春岁月，我并非一人走过，正是有了老师和同学们的一路陪伴，我的求学生涯才会如此美好。

首先感谢我的导师彭志豪老师，彭老师不仅治学严谨，而且平易近人，以诚待人。因为我自己的分析设计水平不高，所以在一开始总是信心不足。而彭老师总会耐心的教导我们，鼓励我们，经常召集同门一起进行交流，询问我们学习上的情况，对未来的想法，为我们指点迷津。彭老师，再次谢谢您，希望我在未来的道路上，能够不负您的希望。其次，我要感谢在东软有缘相识的各位老师：付丽梅老师、陈鹏老师、林跃进老师、赵旭老师、王文生老师、骆伟老师、刘丹老师、张冬青老师、张晓煜老师、刘珊珊老师。你们渊博的学识，谦和仁爱的品性，春风化雨，深深感染着我。

感谢可爱的同学们，学术的道路上难免枯燥寂寞，可你们的陪伴与关怀，总能让我倍感温暖，感谢一路同行的室友，相聚即是莫大的缘分，愿我们都能永远保持年轻的热血，一路勇敢前行。

感谢父母一直以来的支持与鼓励。谢谢你们尊重我的每一个决定，谢谢你们为我一路遮风挡雨。愿你们永远平安康健！希望我的努力，能对得起你们二十余年的辛勤。

感谢一直以来对我照顾有加的辅导员车艳茹老师，我不是您最出色的学生，而您却是最最尊敬的老师，我从认识您开始一直就这样认为，您与其他老师不同，您给予学生的关心要比其他老师多得多，只要是有利于学生成长的事您都要过问，即使会换来非议的目光与言语，您依然以您的认真和执着坚持着白雪般的追求，也许您的人生信条是纯洁伟大的。

最后，感谢母校，大连东软信息学院，我生命中最美好的四年，是与你一起走过的。学子园的静谧风光，图书馆内的书声琅琅，都是我求学生涯中深刻的记忆。临行之际，不诉离殇。只愿我离开你后的每一步，都能坚毅笃定，只愿我离开你后的每一天，都能不忘初心。

大连东软信息学院

毕业设计（论文）原创承诺书

1、本人承诺：所提交的毕业设计（论文）是认真学习理解学校的《毕业设计（论文）工作规范》后，在教师的指导下，独立地完成了任务书中规定的内容，不弄虚作假，不抄袭别人的工作内容。

2、本人在毕业设计（论文）中引用他人的观点和研究成果，均在文中加以注释或以参考文献形式列出，对本文的研究工作做出重要贡献的个人和集体均已在文中注明。

3、在毕业设计（论文）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

4、本人完全了解学校关于保存、使用毕业设计（论文）的规定，即：按照学校要求提交论文和相关材料的印刷本和电子版本；同意学校保留毕业设计（论文）的复印件和电子版本，允许被查阅和借阅；学校可以采用影印、缩印或其他复制手段保存毕业设计（论文），可以公布其中的全部或部分内容。

5、本人完全了解《毕业（设计）论文工作规范》关于“学生毕业设计（论文）出现购买、他人代写、或者抄袭、剽窃等作假情形的，取消其学位申请资格；已经获得学位的，依法撤销其学位。取消学位申请资格或者撤销学位者，从处理决定之日起3年内，学校不再接受学生学位申请”的规定内容。

6、本人完全了解《学生手册》中关于在“毕业设计（论文）等环节中被认定抄袭他人成果者”不授予学士学位，并且“毕业学年因违纪受处分影响学位的学生不授予学士学位，并且无学士学位申请资格”的规定内容。

以上承诺的法律结果、不能正常毕业及其他不可预见的后果由学生本人承担！

学生本人签字：

年 月 日