

《软件工程》 课程报告

(2021-2022 学年第 2 学期)

题目：基于 WEB 的软件工程课程文件管理系统

学生姓名： 李文基 刘静

学 号: 41912064 41912162

提交日期: 2022 年 6 月 12 日

所在学院	计算机与人工智能学院	专 业	计算机科学与技术
教师评语：			
本论文成绩评定： _____分			

一、课程设计任务概述

1. 目的与任务

文件管理系统的主要工作过程为：当进入网站首页，用户可以看到网站数据库中存的所有文件，用户可以上传文件、下载文件或者删除文件，还可以把复制的内容添加到网站数据库，然后再另一个客户端点击复制按钮复制内容到剪切板。

2. 开发环境

操作系统：windows 10

开发工具：MySQL 8.0.23 ,

IntelliJ IDEA 2020.3.2

JDK1.8,

Maven

Springboot 2

Mabatis-plus

java server pages

3. 软件配置

在 IntelliJ IDEA 完成网页的编写。

使用 Visio 2007 进行数据流图的绘制。

使用 IntelliJ IDEA 服务器端的编写

使用 maven 管理项目所需依赖包

使用 mybatis-plus 操作数据库

使用 MySql 进行数据库端的编写

4. 参考资料

《软件工程》

《数据库系统开发》

《JavaWeb 整合开发王者归来》

JDK1.8 参考文档

J2EE5 参考文档

二、介绍主题、构思与创意

我们小组选择的课题为“基于 WEB 的软件工程课程文件管理系统”，文件管理系统是一个可以轻松完成文件互传的系统，它主要用于传文件，可以把手机端的文件传给电脑端，同时也可以电脑端传给手机端，还可以实现把复制的内容通过本系统互相传递给不同的客户端。

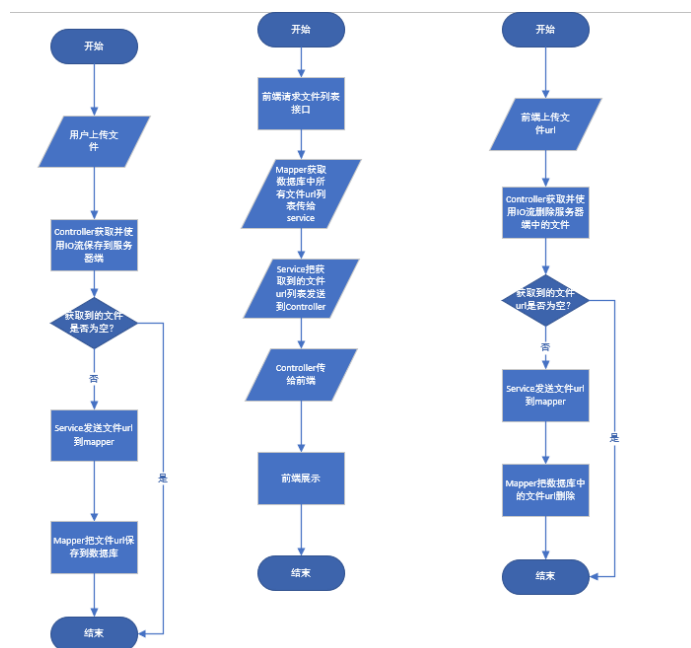
三、系统设计

在开始着手之前，我首先构思数据库的设计。确定前台需要哪些数据，数据结构式什么，数据库是怎么设计，dao 层设计了哪些接口，然后我再根据这些需求设计出一套传送数据的方案。传输数据的防范被设计出之后交付给前台，由前台进行测试，再由测试结果修改代码。

在传递参数的过程中，如果在表单中使用 post 方式提交中文名的文件，数据库收到的文件名会变成乱码，造成异常，进而向浏览器返回 500 错误。之后我进一步查找错误，发现数据库在创建的时候没有设置字符编码进而默认为 latin1，不能存储中文，将其设置为 utf-8mb4 之后解决了这个问题。

四、介绍完成的工作的具体过程

在项目中，首先构思并画了流程图



1. 文件列表

文件列表主要通过 axios 请求后端接口，获取文件列表并渲染到页面上

Axios

```
const ule = document.querySelector('.ule');
axios.get(`${pageContext.request.contextPath}/fileList`)
  .then((value) => {
    a = value.data;
    for (var i = 0; i < a.length; i++) {
      urlxz = "/download/" + a[i].url;
      urlsc = "/delete/" + a[i].url;
      str += '<li>' + a[i].url + '&nbsp;' + '<a href="' + urlxz + '">' + '点击下载' + '</a>' + '<a href="' + urlsc + '">' + '点击删除' + '</a>' + '</li>';
    }
    ule.innerHTML = str;
  })
```

Controller 层

```
@Autowired
private FileUrlService fileUrlService;

@GetMapping("/fileList")
public List<FileUrl> list(){
    return fileUrlService.list();
}
```

Service 层

```
@Override
public List<FileUrl> list() {
    return mapper.selectList(Wrappers.<~>lambdaQuery()
        .select(FileUrl::getUrl)
        .orderByDesc(FileUrl::getId)
    );
}
```

Mapper 层

```
@Mapper
public interface FileUrlMapper extends BaseMapper<FileUrl> {
}
```

因使用了 mybatis-plus 中的 insert 语句，使 mapper 层不需要写 sql 语句

2. 上传文件

上传文件主要通过 jsp 的表单提交，把上传的文件发送到后端，后端在通过 controller 层获取并使用 IO 流把文件保存到服务器端，经 service 和 mapper 层保存文件 url 到数据库中

Jsp

```
<input id="input" type="text"/>
<div class="box">
    <div>
        <form action="/upload" method="post" enctype="multipart/form-data">
            <input type="file" id="file" class="xzwj" name="file" multiple/>
            <button type="submit" class="dwsc">点我上传</button>
        </form>
        <div id="notice"></div>
    </div>
    <ul class="ule">
    </ul>
</div>
```

Controller 层

```
@PostMapping("/upload")
public String fileUpload(@RequestParam(value = "file") MultipartFile file){
    if (file == null) {
        return "error";
    }
    if (file.getSize() > 1024 * 1024 * 1000) {
        return "error";
    }
    //获取文件名
    String suffix = file.getOriginalFilename();
    String savePath = UPLOAD_FOLDER;
    File savePathFile = new File(savePath);
    if (!savePathFile.exists()) {
        //若不存在该目录，则创建目录
        savePathFile.mkdirs();
    }
    try {
        //将文件保存指定目录
        file.transferTo(new File(pathname: savePath + suffix));
        fileUrlService.add(suffix);
    } catch (Exception e) {
        e.printStackTrace();
        return "error";
    }
    return "ok";
}
```

Service 层

```

@Override
public void add(String suffix) {
    FileUrl fileUrl = new FileUrl();
    • fileUrl.setUrl(suffix);
    mapper.insert(fileUrl);
}

```

Mapper 层

```

@Mapper
public interface FileUrlMapper extends BaseMapper<FileUrl> {
}

```

3. 删除文件

删除文件主要通过 axios 把要删除文件 url 通过 html 中的 a 标签发送到 controller, 使用 io 流把服务器端的文件删除并通过 service 和 mapper 把数据库中的文件 url 删除

Axios

```

const ule = document.querySelector('.ule');
axios.get(`${pageContext.request.contextPath}/fileList`)
    .then((value) => {
        a = value.data;
        for (var i = 0; i < a.length; i++) {
            urlxz = "/download/" + a[i].url;
            urlsc = "/delete/" + a[i].url;
            str += '<li>' + a[i].url + '&nbsp;&nbsp;&' + '<a href="' + urlxz + '">' + '点击下载' + '</a>' + '<a href="' + urlsc + '">' + '点击删除' + '</a>' + '</li>';
        }
        ule.innerHTML = str;
    })

```

Controller

```

@Value("${prop.upload-folder}")
private String UPLOAD_FOLDER;
@Autowired
private FileUrlService fileUrlService;
@GetMapping("/delete/{path}")
public String uploadCustomPathFile(@PathVariable String path){
    FileSystemUtils.deleteRecursively(new File( pathname: UPLOAD_FOLDER + path));
    fileUrlService.delete(path);
    return "ok";
}

```

Service

```

@Override
public void delete(String path) {
    mapper.delete(Wrappers.<~>lambdaQuery()
        .eq(FileUrl::getUrl, path)
    );
}

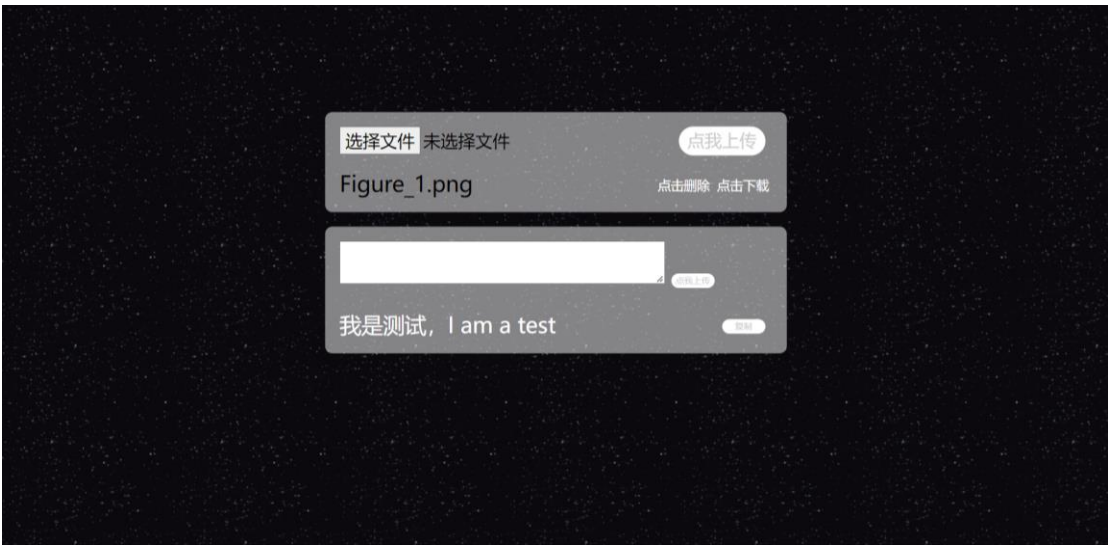
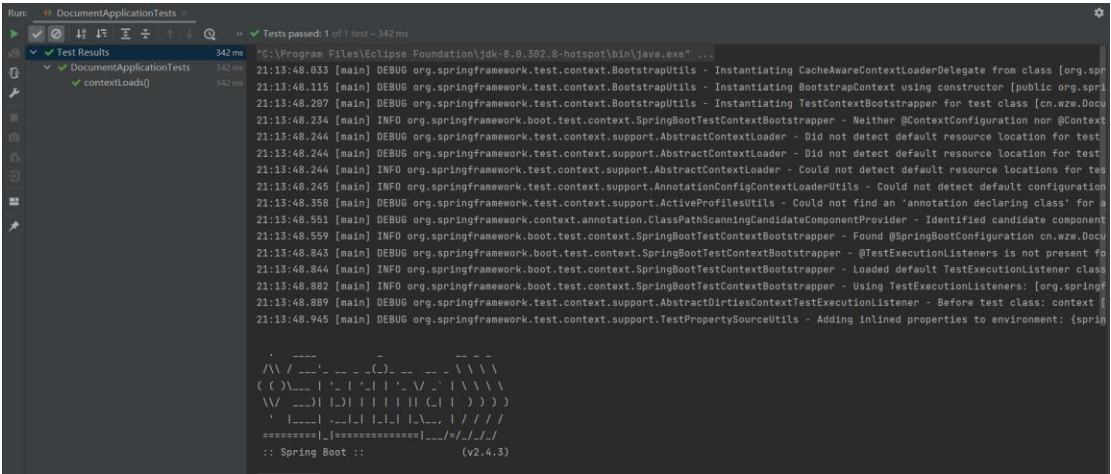
```

Mapper

```
@Mapper
public interface FileUrlMapper extends BaseMapper<FileUrl> {
}
```

五、单元测试

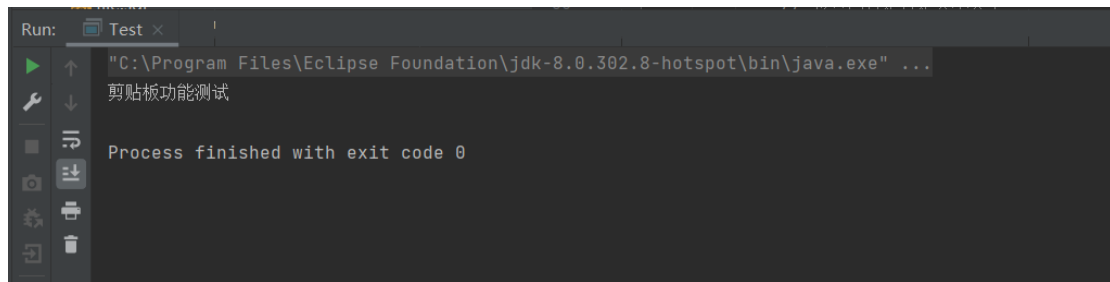
1. 对整个项目进行测试



项目基本功能——上传文件与上传文字均能正常运行。

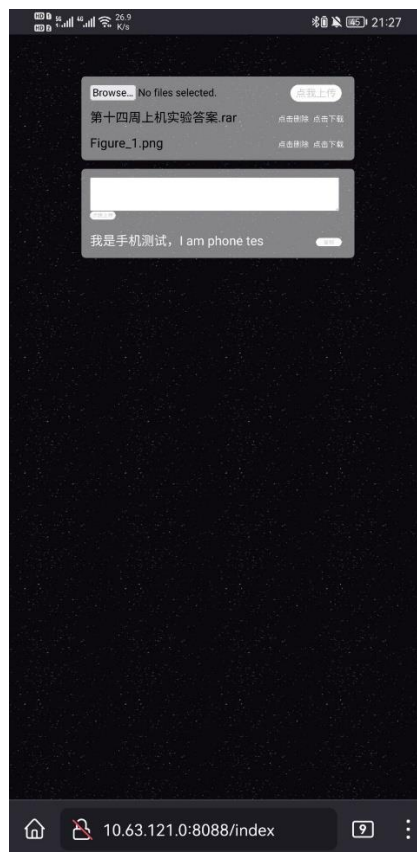
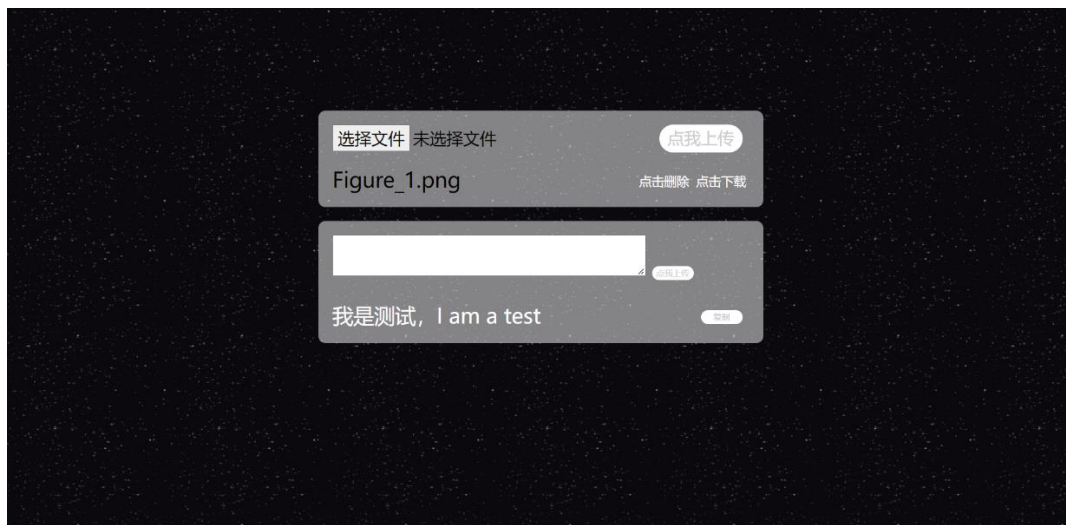
2. 测试读取剪贴板功能

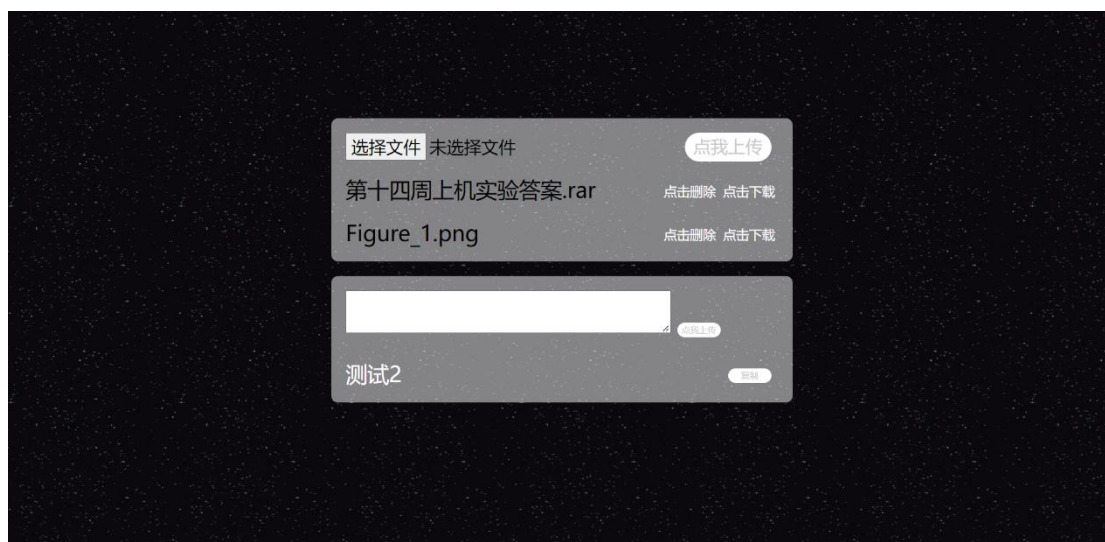
复制“剪贴板功能测试”，情况如下：



项目能正常读取剪贴板内容。

3. 测试局域网（校园网）下上传情况



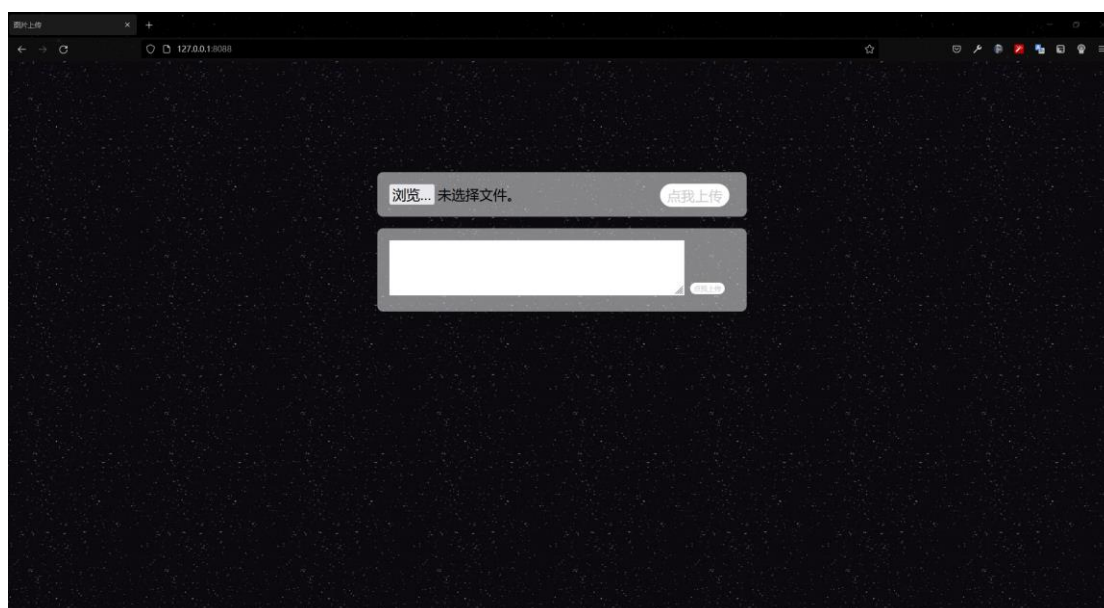


如上图所示，手机端操作成功，且项目能自适应移动设备浏览器，并完成上传删除操作。

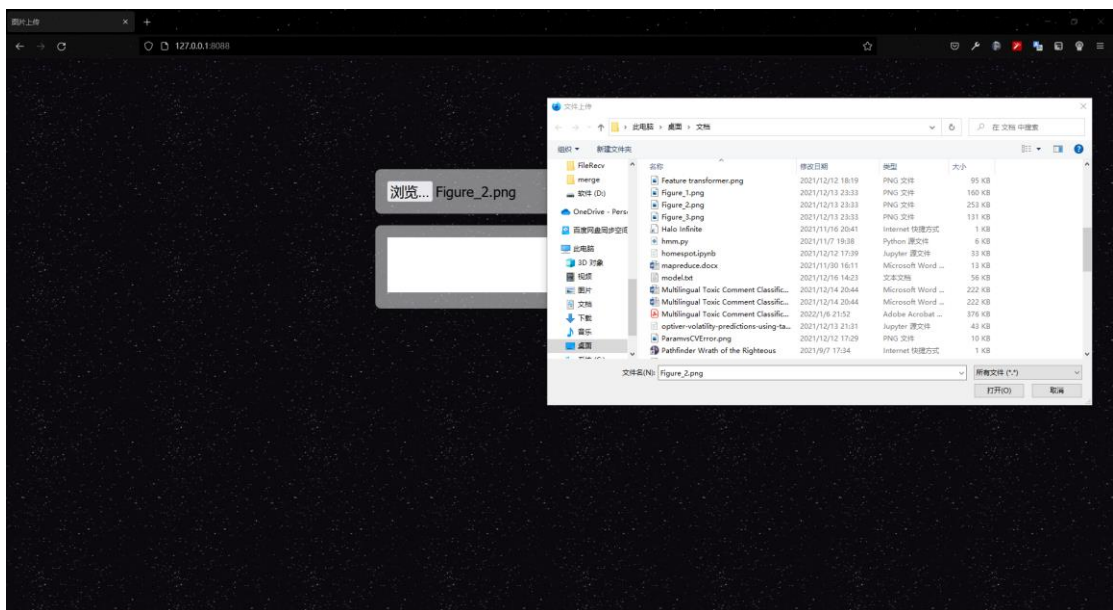
六、网站项目使用说明

1. 文件上传

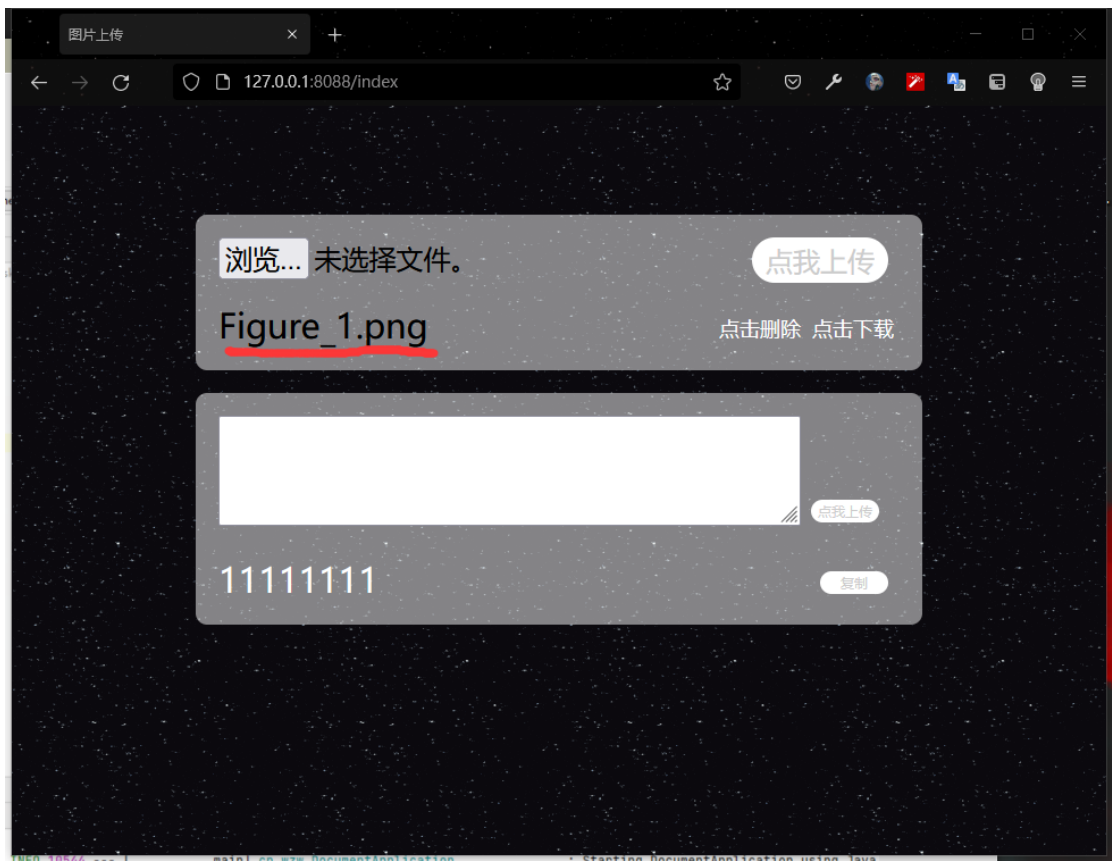
第一步：运行项目，打开网站



第二步：点击浏览按钮，选择文件上传

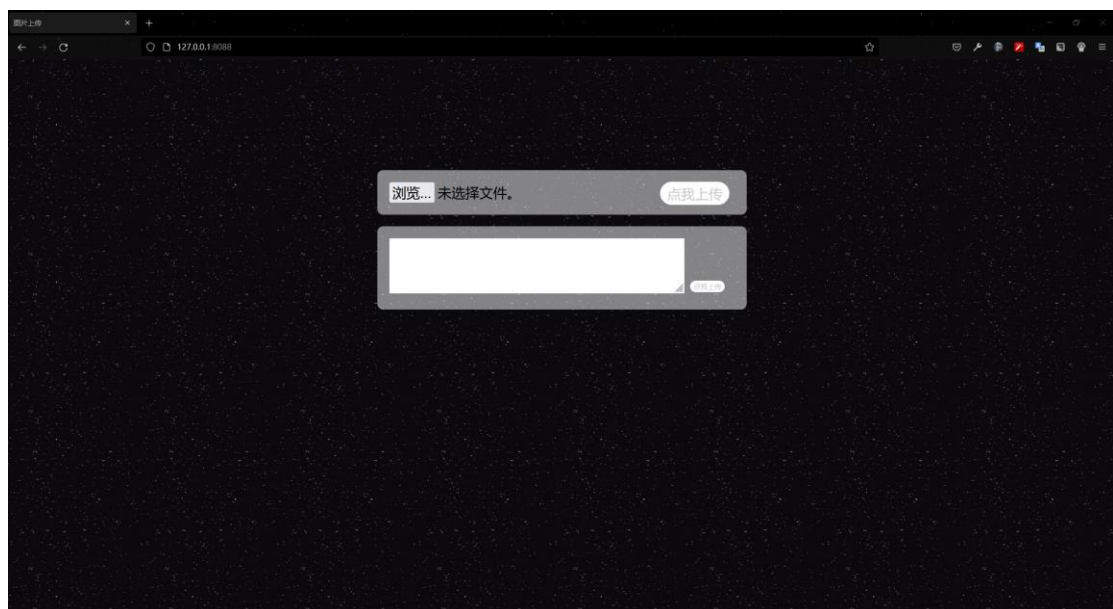


第三步：上传文件



2. 文字上传

第一步：打开项目，在文本框中输入上传文字



第二步：点击上传按钮，文字即可上传



七、个人小结

在参与代码实现过程中,不仅要熟悉 SQLyog 工具的使用,还要熟悉 IntelliJ IDE 工具的使用,用代码实现系统的功能会用到各种技术,包括数据库架构的设计、接口的实现、代码实现与数据库的链接等等,这些都需要好好学习,在系统实现过程中,只有边学习,边应用才能更好地掌握知识,学习到更多的东西。

通过这次课程设计,我了解到本身还有很多不足之处,对理论知识的掌握不够透彻,平时缺乏把理论知识应用于实践,从而减弱了动手能力。在代码编写方面,自己所掌握的编程语言还不够应用于实践,而且编码的水平有待提高。同使我也收获了很多知识,学会了流程图的画法、数据字典的确定、系统功能模块的划分、对系统的测试方法等等,学习到了各种知识点。在今后的学习过程中,我会既学习好理论知识,又要注意将理论知识应用于实践,这样才能更好地掌握知识,提高动手能力,增强个人能力,这样才能更快的学习到新知识,更快的提高自己。