

**Web技术与应用I**

**课程设计报告**

**题目 局域网即时通信系统**

**学生姓名**

**学 号**

**院 系 软件学院**

**年级专业 2021软件工程专业**

**班 级 3班**

**指导教师**

**二Ｏ二三年 12 月 2 日**

局域网即时通信系统：设计类似于qq文字通信的软件，可以实时发送、接收文字内容，可以发送、接收文件。

1.实习目的

通过局域网即时通信系统课程设计实习过程，进一步巩固《Web技术与应用I》课程所学的理论知识，增强学生利用所学内容获取相关知识，以解决实习过程中所面临的问题，如实时发送接收文字内容，发送接收文件和技术难点文件的发送和接收，以提高利用Java语言、JSP Web 开发技术实际动手进行程序设计的能力。

2.实验内容

随着互联网的普及和局域网技术的发展，局域网即时通信系统应运而生。在传统的办公环境中，人们通常使用电子邮件、电话或面对面的会议来进行沟通和协作。然而，这些方式存在一些限制，比如电子邮件的延迟较高，电话通话需要实时沟通等。

局域网即时通信系统的出现填补了这些沟通方式的空白，为办公环境提供了更便捷和高效的实时通信解决方案。它利用局域网内部的高速网络连接和内部服务器，实现了快速、实时的消息传递和多媒体通信。

用户可以立即收到对方发送的消息，无需等待，并且可以通过简单的操作即可发送消息和文件，不需要复杂的步骤。

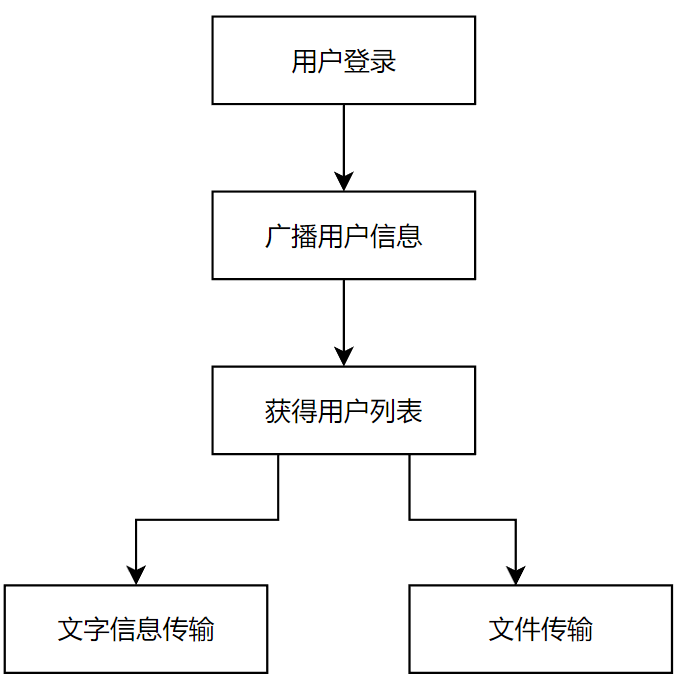
但是现有的即时通信系统也有很多缺点，例如存在安全漏洞，容易受到黑客攻击，用户的个人信息和文件可能会被窃取。并且由于网络环境的不稳定性，可能会导致消息延迟或丢失，影响用户体验。

因此为了提高用户体验，设计一个快捷，实时，多功能，安全并且不易收到网络影响的局域网即时通信系统就很有必要。

在局域网即时通信系统中，用户可以即时发送、接收文字消息，保证沟通的及时性和准确性。同时还可以发送、接收各种类型的文件，方便用户之间的文件共享。同时用户还可以在不同的平台上面进行登录，实现了多平台模式。

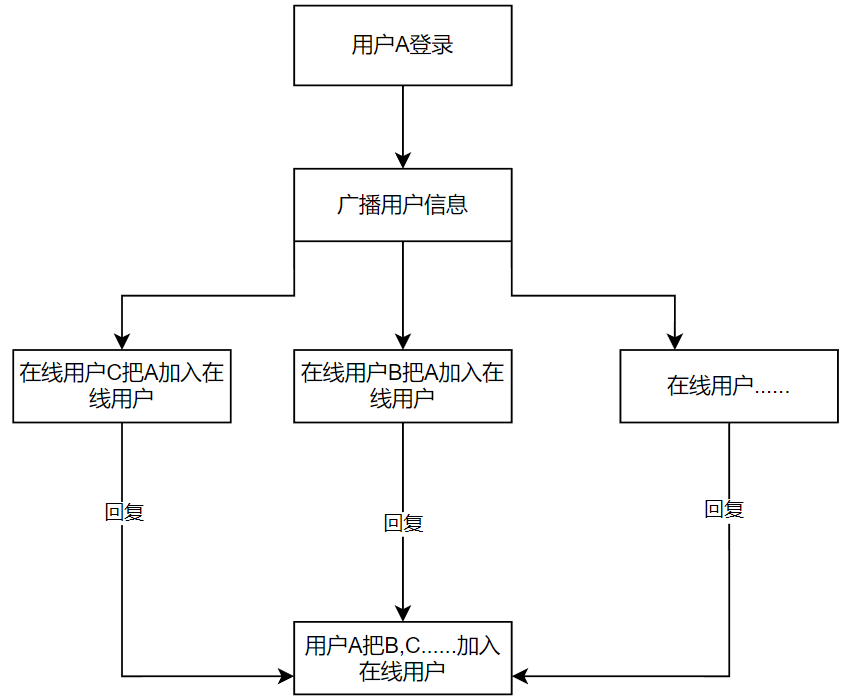
3.总体设计方案

（1）总体功能结构图

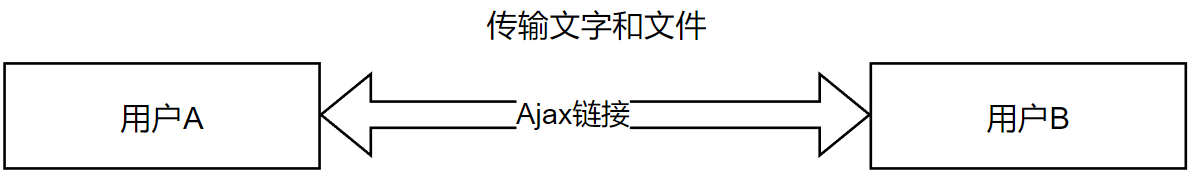


（2）各子功能的主要算法流程图（标准流程图，根据系统运行情况详细设计）

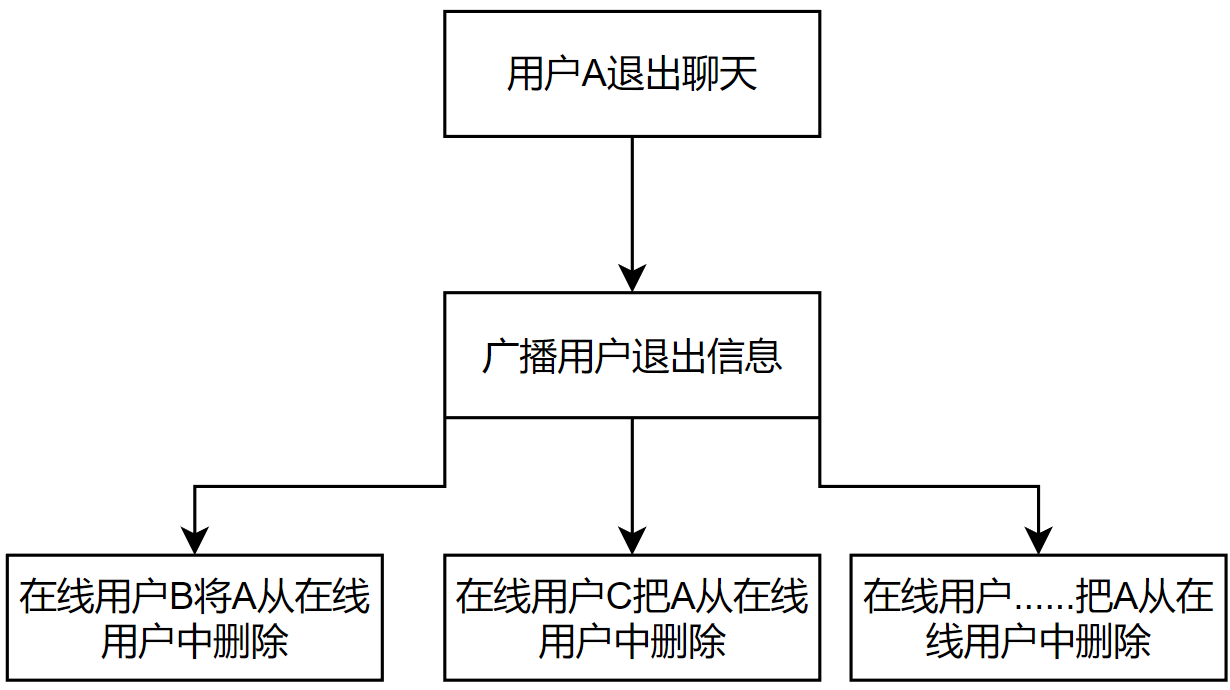
用户登录阶段



用户聊天阶段



用户退出阶段



4.系统实现

用户输入用户名即可进入聊天室



//index.jsp部分代码：

<script type="text/javascript">

//发送请求

function startRequest(val) {

createXMLHttpRequest();

xmlHttp.onreadystatechange = function() {

if(xmlHttp.readyState == 4) { //如果已经完成解析

if(xmlHttp.status == 200) { //如果请求是成功，并得到响应

var flag = $.trim( xmlHttp.responseText );

console.log(flag);

if(flag == "true"){

$("#myh").text("该用户已经登录").css("color", "red");

$("#submit").attr("disabled", "true");

}else{

$("#myh").text("该用户可以登录").css("color", "green");

$("#submit").attr("disabled", null);

}

}

}

};

xmlHttp.open("POST", "doinfo.jsp");

xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

xmlHttp.send("name="+val);

}

</script>

//doinfo.jsp部分代码

request.setCharacterEncoding("utf-8");

String name = request.getParameter("name");

List<String> userList = new ArrayList<String>();

boolean flag = false;

if( application.getAttribute("userList") != null ){

userList = (List<String>)application.getAttribute("userList");

if( userList.size()>0 ){

for(String str : userList ){

if( str.equals(name) ){

flag = true;

break;

}

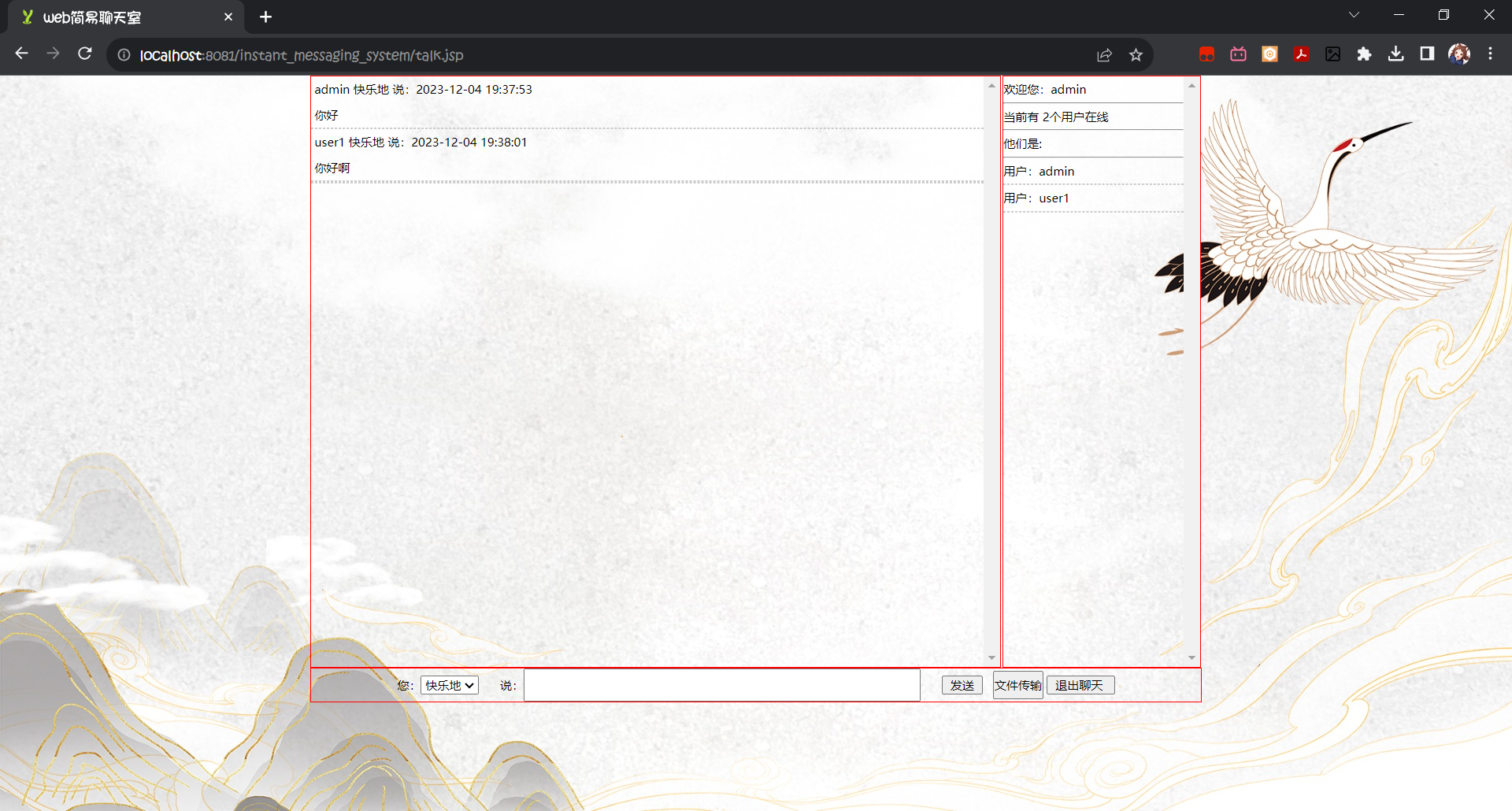
}

}

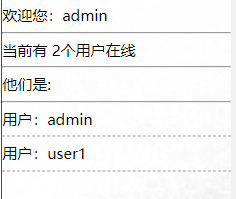
}

out.print(flag);

用户进入聊天室后，进入如下界面



界面右边是用户列表，显示当前时刻有多少用户在线



//content.jsp部分代码

<div id="info1">

欢迎您：<%= session.getAttribute("currentUser") %>

<br /><hr />

<%

List<String> userList = (List<String>)application.getAttribute("userList");

%>

当前有 <%= userList.size() %>个用户在线

<br /><hr />他们是:<br /><hr />

<%

for(String str:userList){

%>

用户：<%=str %><hr style="border:1px dashed #CCC"/>

<%

}

%> </div>

界面下方是用户输入信息的地方，用户在这里发送自己的内容，可以输入文字或者传输文件，并且可以选择自己的心情。



//talk.jsp部分代码：

<form action="dotalk.jsp" method="post" id="myform">

您：<select name="status" id="status">

<option value="快乐地">快乐地</option>

<option value="悲伤地">悲伤地</option>

<option value="无聊地">无聊地</option>

<option value="得瑟地">得瑟地</option>

</select>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

说：<input type="text" name="content" id="content"/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="button" value=" &nbsp;发送&nbsp; " onclick="addContent()"/>&nbsp;&nbsp;

<button type="button" onclick="openNewPage()">文件传输</button>

<script>

// JavaScript函数，用于在按钮点击时打开另一个页面

function openNewPage() {

// 使用 window.location.href 跳转到另一个页面

window.location.href = 'upload.jsp';

}

</script>

<input type="button" value=" &nbsp;退出聊天 &nbsp; " onclick="loginOut()"/>

</form>

界面中间则是用户发送的消息，各个用户可以在这里看到自己和别人发送的消息。通过这个地方来进行交流。



//talk.jsp部分代码：

<script type="text/javascript">

//ajax的封装

function myAjax(url, params, fun){

var xmlHttp;

if(window.XMLHttpRequest) {

xmlHttp = new XMLHttpRequest();

}else if(window.ActiveXObject) {

xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlHttp.onreadystatechange = function() {

if(xmlHttp.readyState == 4) { //如果已经完成解析

if(xmlHttp.status == 200) { //如果请求是成功，并得到响应

fun(xmlHttp.responseText);

}

}

};

xmlHttp.open("POST", url);//请求的方式

xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

if( params == null ){

xmlHttp.send( null );

}else{

var str = "";

for( var s in params){

str += s + "=" + params[s] + "&";

}

xmlHttp.send(str);

}

}

//发送聊天信息

function addContent(){

var content = $.trim( $("#content").val() );

var status = $.trim( $("#status").val() );

if(content != ""){

myAjax("dotalk.jsp", {status:status,content:content}, function(data) {

$("#talkInfo").html(data);

$("#content").val("");

});

}

}

//……

</script>

当用户想要发送文件时，点击文件传输按钮，即可进入文件传输界面



//upload.jsp部分代码：

<form action="upload" method="post" enctype="multipart/form-data">

<table>

<tr>

<td></td>

<td><h1>文件上传</h1></td>

</tr>

<tr>

<td>文件描述:</td>

<td><input type="text" name="desc"/></td>

</tr>

<tr>

<td>上传文件:</td>

<td><input type="file" name="file"/></td>

</tr>

<tr>

<td></td>

<td><input type="submit" value="上传文件"/></td>

</tr>

</table>

</form>

<div>

<h2>下载文件</h2>

<c:forEach items="${files }" var="name">

<a href="upload/${name }">${name }</a><br/>

</c:forEach>

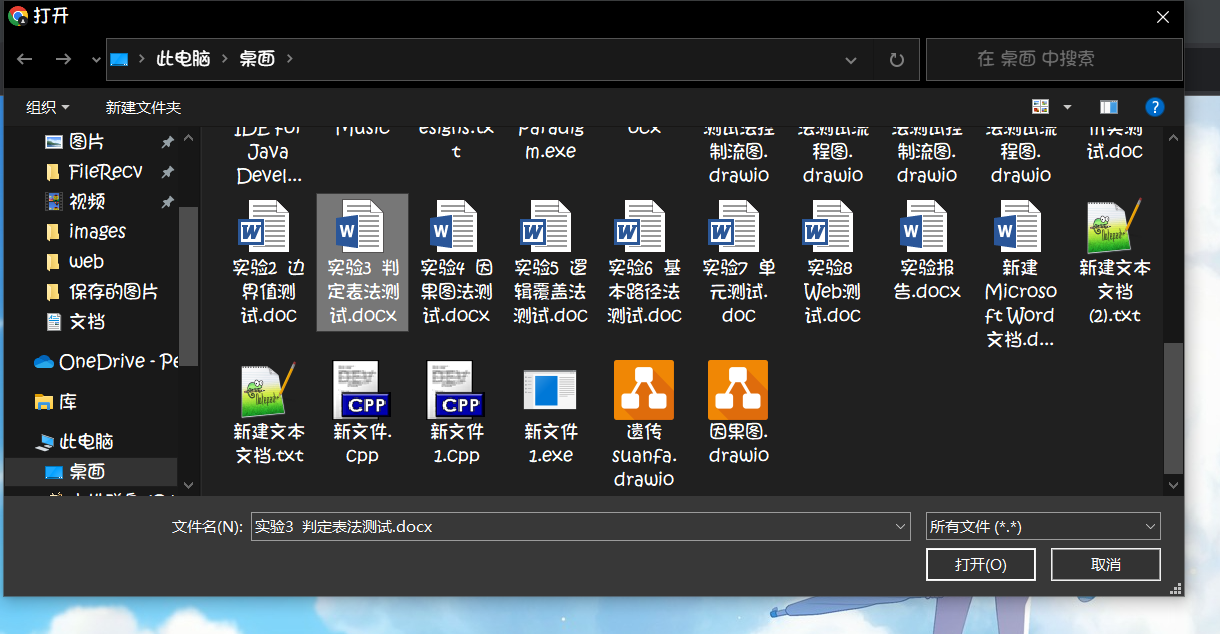
<form action="talk.jsp" method="get">

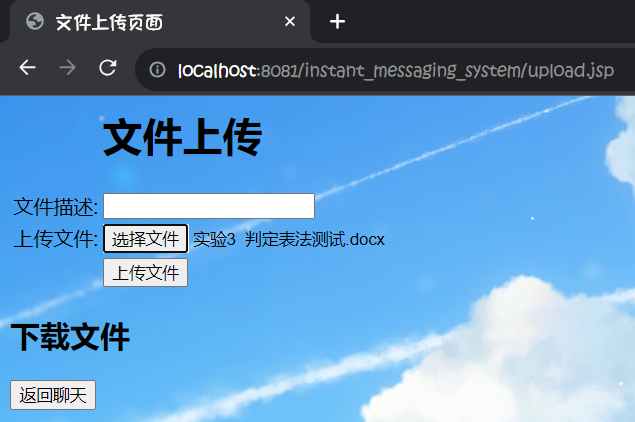
<button type="submit">返回聊天</button>

</form>

</div>

用户可以选择上传或者下载文件，用户点击选择文件，即可在自己的电脑中选择自己想要发送的文件

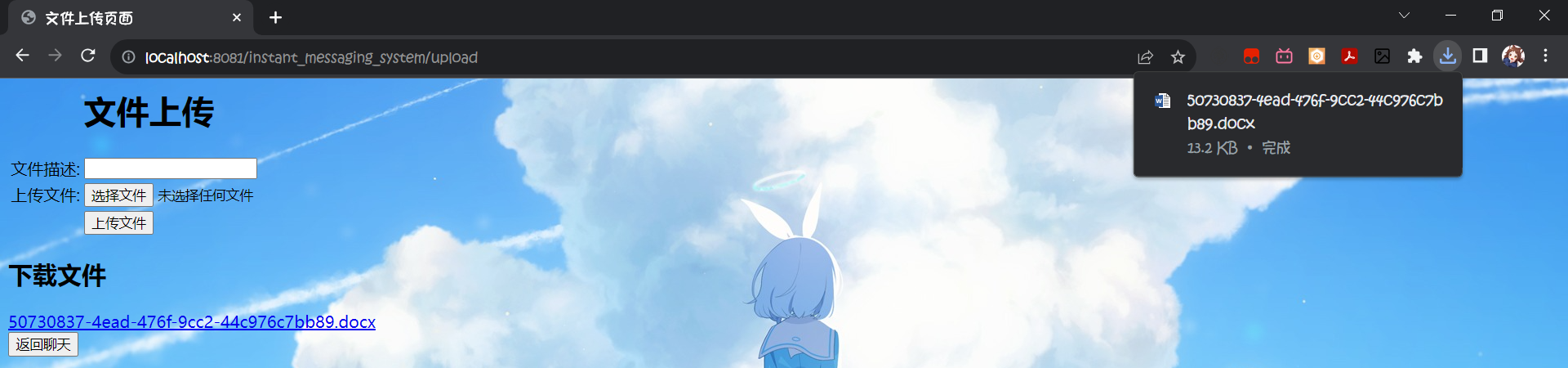




选择好想要发送的文件后，点击上传即可发送文件, 上传文件以后，会显示上传文件成功



用户想要下载文件时，点击文件名即可



//UploadServlet.java文件传输相关代码  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

//模拟数据库存储文件名

HttpSession session=request.getSession();

List<String> list=(List<String>)session.getAttribute("files");

if(list==null){

//如果集合为空，就创建一个集合

list=new ArrayList<String>();

}

try {

//获取文件描述信息

String desc=request.getParameter("desc");

//获取上传的文件

Part part=request.getPart("file");

//获取请求的信息

String name=part.getHeader("content-disposition");

//获取上传文件的目录

String root=request.getServletContext().getRealPath("/upload");

System.out.println("测试上传文件的路径："+root);

//获取文件的后缀

String str=name.substring(name.lastIndexOf("."), name.length()-1);

System.out.println("测试获取文件的后缀："+str);

//生成一个新的文件名，不重复，数据库存储的就是这个文件名，不重复的

String fname=UUID.randomUUID().toString()+str;

//将文件名保存到集合中

list.add(fname);

//将保存在集合中的文件名保存到域中

session.setAttribute("files", list);

String filename=root+"\\"+fname;

System.out.println("测试产生新的文件名："+filename);

//上传文件到指定目录，不想上传文件就不调用这个

part.write(filename);

request.setAttribute("info", "上传文件成功");

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("info", "上传文件失败");

}

request.getRequestDispatcher("/upload.jsp").forward(request, response);

}

5.总结

在这次的Web JSP课程设计中，我们着手设计了一个类似于QQ文字通信的局域网即时通信系统，该系统具备实时发送和接收文字内容的功能，并支持文件的传输。这个设计挑战让我们深入了解了JavaServer Pages（JSP）技术和Web开发的方方面面，同时也让我们更好地理解了即时通信系统的设计与实现。

首先，在系统的设计阶段，我们明确了需求和功能，通过分析用户的使用场景和期望，我们定义了系统的核心功能：文字通信和文件传输。这帮助我们明确了系统的架构和各个模块之间的关系。我们充分利用了JSP的特性，将前端和后端进行了有效的分离，使得系统更加模块化和可维护。

在前端设计中，我们充分利用了HTML、CSS和JavaScript等技术，打造了一个直观、友好的用户界面。通过JSP，我们能够轻松地在页面中嵌入Java代码，实现内容的生成和展示，这为用户提供了良好的操作体验，使得文字通信和文件传输变得更加直观和方便。

后端的核心是Java Servlet和JSP，我们通过这两者的协作，处理了用户的请求和数据的传递。实时通信系统的设计中，对于消息的实时性要求较高，通过Servlet的异步特性，我们能够更加高效地处理用户的消息发送和接收请求，保障了通信的即时性。同时，通过合理的数据库设计，我们实现了用户信息和消息的持久化存储，确保了系统的稳定性和可靠性。

在实现文件传输功能时，我们考虑了数据的可靠性。通过Java的文件操作和网络编程技术，我们实现了文件的UUID唯一识别码，保证了文件传输的稳定性和高效性。总的来说，这次的Web JSP课程设计不仅让我们深入学习了JSP技术和Web开发的实践，还锻炼了我们对系统设计和开发的能力。

通过设计局域网即时通信系统，我们不仅仅实现了基本的文字通信和文件传输功能，更在实践中加深了对于网络通信、数据库设计、权限控制等方面的理解。这个课程设计是一个全面而深刻的实践，为我们今后的职业生涯打下了坚实的基础。

6.参考文献

[1] 刘晓华，张健，周慧贞.JSP应用开发详解[M].北京：电子工业出版社,2023.

[2] CHOPRA V, EAVES J, JONES R. JSP程序设计[M].张文静，林琪，译.北京：人民邮电出版社,2023

[3] 耿祥义.JSP基础教程[M].北京：清华大学出版社,2023