武汉纺织大学

Web应用开发课程设计

**物流打印单系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11803**

**姓 名： 王成功**

**学 号： 1804280329**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1 需求分析 1](#_Toc60451104)

[1.1生成物流打印单图片 1](#_Toc6247)

[1.2主页面分析 1](#_Toc15319)

[1.3系统分析 2](#_Toc14458)

[1.4静态页面 2](#_Toc14458)

[1.5动态交互 2](#_Toc14458)

[1.3系统分析 2](#_Toc60451107)

[2 系统设计 3](#_Toc60451111)

[2.1 Web项目目录结构 3](#_Toc60451112)

[2.2系统流程 4](#_Toc60451113)

[3 系统实现 5](#_Toc60451114)

[3.1 项目结构 5](#_Toc60451115)

[3.2第一步：前端代码的实现 5](#_Toc60451116)

[3.3第二步：使用JQ和AJAX实现页面交互 7](#_Toc60451117)

[3.3第三步：使用exlipse创建createimg.java文件 8](#_Toc60451118)

[3.4第四步：在eclipse中创建test.java文件对createimg文件进行调用 9](#_Toc60451120)

[4 系统测试 10](#_Toc60451121)

[5 系统总结 13](#_Toc60451122)

# 1 需求分析

设计一个物流单打印系统，实现“寄件人”、“寄件人电话”、“寄件公司”、“寄件人地址”、“收件人”、“收件人电话”、“收件人公司”、“收件地址”等功能，具体要求如下：

## 1.1生成物流打印图片



当程序运行时，输入冒号后的信息后，生成以上格式的图片，443×649

其中，第一个条码，第三个条码和二维码对应的信息是输入的物流单号，第二个条码对应的数字是自动生成的订单号,生成格式是yyyyMMddHHmmSS+三位时间戳+三位随机数

## 1.2主页面分析

由图可分析出，左边的内容冒号前面的内容都为静态可显示的，冒号右边的内容是动态生成的，二维码是随机生成的，条形码是根据自己输入的数据自动生成的。

## 1.3系统分析

## 

## 1.4静态页面

在图中分析可得其中最左边的一排为静态固定无变化的内容，如下：

===物流打印单生成系统===

1. 始发网点
2. 寄件人
3. 寄件人电话
4. 寄件公司
5. 寄件人地址
6. 收件人
7. 收件人电话
8. 收件公司
9. 收件地址
10. 官方网址

这些内容都为默认的不会发生变化。

## 1.5动态交互

有题意可得出右边的内容是动态生成的，二维码是随机生成的，条形码是根据自己输入的内容中提取数据，再根据提取到的数据自动生成的。

# 系统设计

## 2.1 Web项目目录结构

Web项目由**前端代码，后端代码，配置文件，第三方库**组成，目录如下



* **src目录：**java源代码及配置文件的存放目录。可根据需要在该目录下创建包，如controller，model，tools，config等。java源码或配置文件会自动编译存到**WebRoot->WEB-INF--classes**目录下
* **WebRoot：**web项目根目录，所有能在客户端浏览器访问的到的资源（html，jsp，css，js，图片等）必需放到此目录下，可根据需要创建子目录存放前端文件（如css，js，img等）。利用Myeclipse打包发布Web项目时，本质是将webRoot下的所有文件复制到tomcat的Webapps->“项目名称”目录下面。
* **WebRoot--->WEB-INF->lib：**存到web项目运行时所需的.jar包，如访问访问数据库时所需的JDBC包。
* **WebRoot--->WEB-INF->web.xml文件**：web项目的核心配置文件，如配置Servlet,过滤器，欢迎页面等。

特别注意：WebRoot->WEB-INF是受保护目录，无法在客户端通过浏览器直接访问该目录下的文件。

## 2.2系统流程

否

开始

输入快递单号

验证用户信息

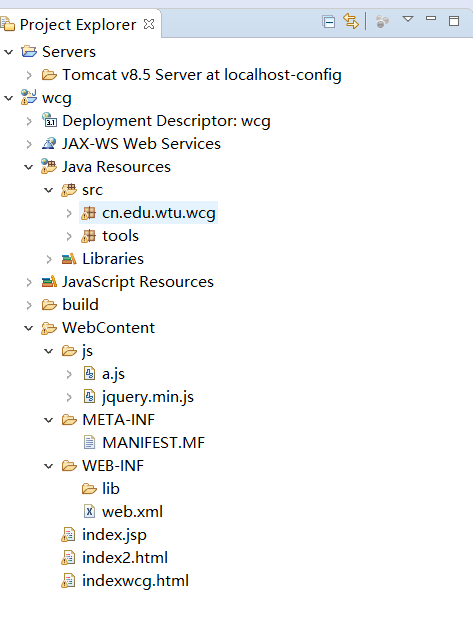
用户存在与否

生成快递单

是

# 3 系统实现

## 3.1 项目结构



## 3.2第一步：前端代码的实现

在项目下新建jsp文件，名为index.jsp

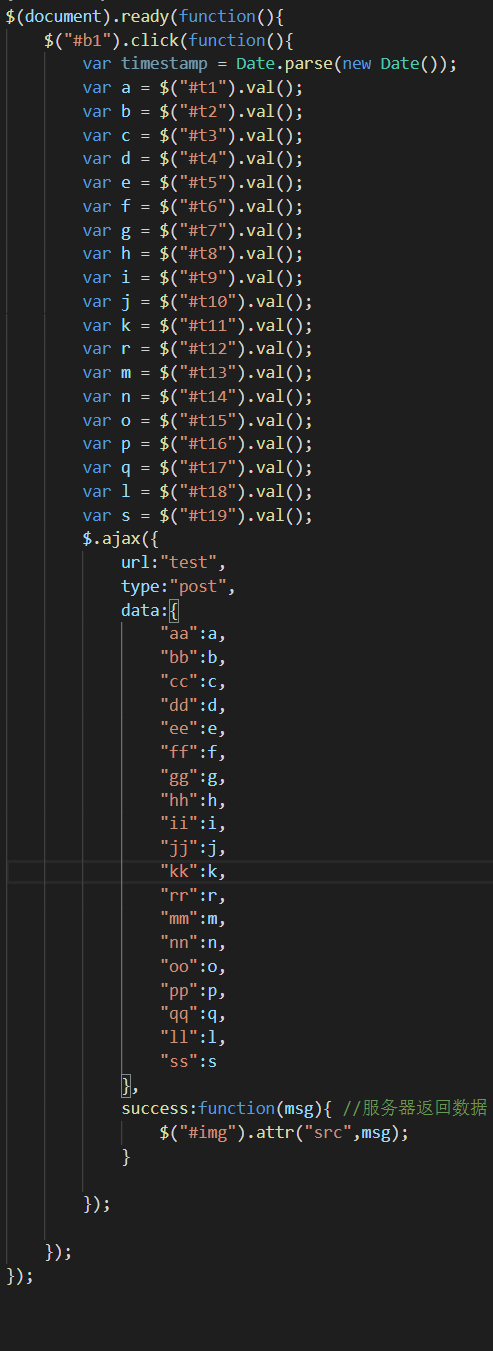
* 在前端文件中在前端框架中设置文件的路径。（所使用的jq库下载在文件之中了）并搭建好所需要获取数据的文本框。
* 提交按钮的提交方式，以及位置
* 生成的图片，以及执行的test文件。



## 3.3第二步：使用JQ和AJAX实现页面交互

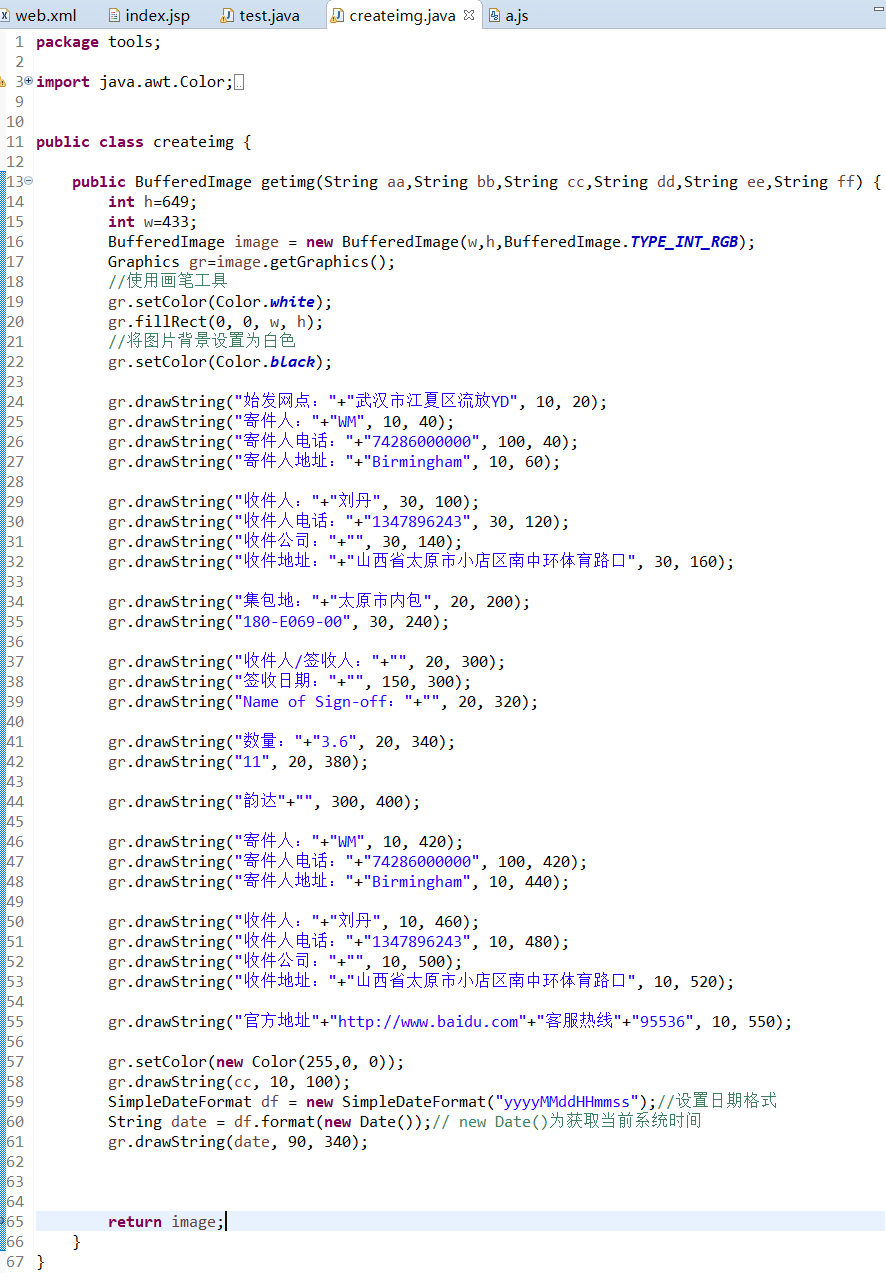
创建一个a.js文件，将其实现后端交互。

* 获取到页面的值
* 当单击鼠标时，将从网页上获取到的数据定义为（a,b···）再将其发送给后端代码，使用Ajax批量获取数据。成功后服务器返回数据为msg。
* data结尾不需要符号。



## 3.3第三步：使用exlipse创建createimg.java文件

### Createimg.java文件创建了一个画笔对象，设置画笔颜色为白色对文件背景进行更改，再将笔色设置为黑色对内容进行增加，获取到当前时间的值，并将其转化为数字，再根据数字的内容生成条形码。返回图片。



## 3.4第四步：在eclipse中创建test.java文件对createimg文件进行调用

* 对设置的对象进行引用
* 对createimg.java文件进行引用，并设置输出的图片格式
* package cn.edu.wtu.wcg;
* import java.awt.image.BufferedImage;
* import java.io.IOException;
* import java.io.OutputStream;
* import java.io.PrintWriter;
* import javax.imageio.ImageIO;
* import javax.servlet.ServletException;
* import javax.servlet.annotation.WebServlet;
* import javax.servlet.http.HttpServlet;
* import javax.servlet.http.HttpServletRequest;
* import javax.servlet.http.HttpServletResponse;
* import tools.createimg;
* /\*\*
* \* Servlet implementation class test
* \*/
* @WebServlet("/test")
* public class test extends HttpServlet {
* private static final long serialVersionUID = 1L;
* /\*\*
* \* @see HttpServlet#HttpServlet()
* \*/
* public test() {
* super();
* // TODO Auto-generated constructor stub
* }
* /\*\*
* \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
* \*/
* protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
* // TODO Auto-generated method stub
* // String a = request.getParameter("aa");
* // String b = request.getParameter("bb");
* // String c = request.getParameter("cc");
* // String d = request.getParameter("dd");
* // String e = request.getParameter("ee");
* // String f = request.getParameter("ff");
* createimg img = new createimg();
* BufferedImage image=img.getimg("a","b","c","d","e","f");

* //System.out.println(b);
* // System.out.println(a);
* //PrintWriter pw = response.getWriter();
* //pw.append(a);
* //pw.close();
* response.setContentType("image/jpeg");
* OutputStream os = response.getOutputStream();
* ImageIO.write(image, "jpeg", os);
* os.close();
* }
* /\*\*
* \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
* \*/
* protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
* // TODO Auto-generated method stub
* doGet(request, response);
* }
* }

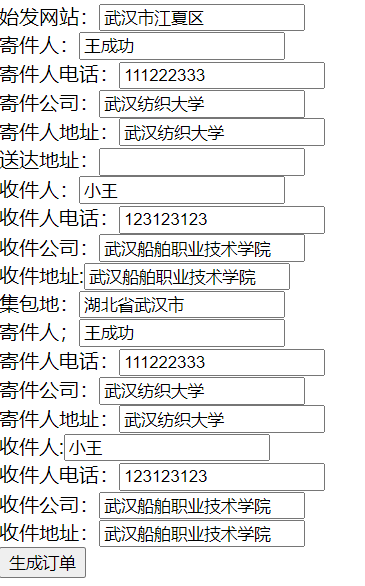
# 4 系统测试

在网页中输入<http://127.0.0.1:8080/wcg/>回车

在exlipse中点击Tomcat右击—Restart in Debug。会显示以下界面



输入以下信息，点击生成订单，则会生成一张相应的物流打印单。如图





# 5 系统总结

这次实验的效果仍然不是非常明显，主要是因为自己之前没学过java，因此在基础方面还是有些薄弱的，在做这次大作业的时候请教了很多人，但这次大作业也然我清楚地了解到我哪些地方的不足，我会在寒假的时间里将其补回来，并不断完善此次作业。