武汉纺织大学

Java应用开发课程设计

**基于MyBatis的超市收银系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 王玉蛟**

**学 号： 1804240204**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月20日**

目 录

[1 需求分析 1](#_Toc2965)

[1.1输出EMS物流面单 1](#_Toc6465)

[1.2将物流图片转换为pdf格式 1](#_Toc27274)

[2 系统设计 3](#_Toc15323)

[2.1用户用例图 3](#_Toc21899)

[2.2 ER图 4](#_Toc16671)

[2.3 UML类图（Class Diagram） 4](#_Toc1640)

[2.4 UML时序图（Sequence Diagram） 6](#_Toc18606)

[2.4.1 系统登录 6](#_Toc3306)

[2.4.2 \*\*\*\*\*\*模块 6](#_Toc6852)

[2.4.3 \*\*\*\*\*\*模块 6](#_Toc11897)

[2.5 UML活动图（Activity Diagram） 6](#_Toc5309)

[2.5.1 登陆 6](#_Toc12858)

[2.5.1 收银 7](#_Toc2967)

[2.5.3 \*\*\*模块 8](#_Toc16882)

[2.5.4 \*\*\*\*模块 8](#_Toc4032)

[3 系统实现 9](#_Toc27433)

[3.1 项目结构 9](#_Toc1061)

[3.2 配置文件 9](#_Toc2679)

[3.2.1 jdbc.properties文件 9](#_Toc19162)

[3.2.2 log4j.properties文件 9](#_Toc31259)

[3.2.3 mybatis-config.xml文件 10](#_Toc7117)

[3.3 VO类User.java 10](#_Toc26179)

[3.4 DAO接口类IUserDAO.java 11](#_Toc32366)

[3.5 接口映射文件UserMapper.xml 11](#_Toc31927)

[3.6 工具包Util 11](#_Toc25901)

[3.6.1 MybatisUtils.java 11](#_Toc6856)

[3.6.2 MD5Util.java 12](#_Toc8757)

[3.7 服务层UserService.java 13](#_Toc29606)

[3.8 用户界面Driver.java 14](#_Toc26330)

[3.9 \*\*\*\*\*\*.java 14](#_Toc14997)

[4 系统测试 14](#_Toc19341)

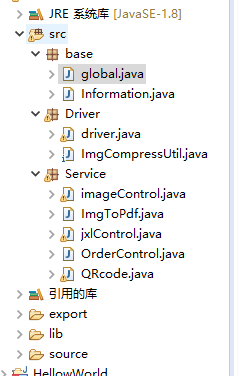
[5 系统总结 14](#_Toc30615)

# 1 需求分析

设计一个EMS物流下单程序，从Excel文件中读取订单数据，生成EMS物流面单图片文件（程序生成订单号，规则yyyyMMddhhmmssSSS+三位随机号），并将图片文件转换为pdf文件。

# **2 系统实现**

## 2.1 项目结构



## 2.2 配置文件

### 2.2.1 global.class文件

package Global;

public class global {

public static final String FILENAME = "./source/订单文件.xls";

public static final String FORMAT=".jpg";

public static final String PICPLACE = "./export/";

public final static int TEXTCOLOR = 0xFFFFFFFF;

public final static int BACKCOLOR = 0xFF00000;

public final static int IMAGEWIDTH = 955;

public final static int IMAGEHEIGHT = 1400;

public final static String TAG="pic";

}

### 2.2.2 information.class文件

package base;

public class Information {

private String orderNum;

private String EMSTrackNum;

private double pureWeight;

private double roughWeight;

private double number;

private String mainGoods;

private String receiName;

private String receiProvinCode;

private String receiAddr;

private String receiTel;

private String consignerName;

private String consignerProvinCode;

private String consignerAddr;

private String consignerTel;

private String note;

public String getOrderNum() {

return orderNum;

}

public void setOrderNum(String orderNum) {

this.orderNum = orderNum;

}

public String getEMSTrackNum() {

return EMSTrackNum;

}

public void setEMSTrackNum(String eMSTrackNum) {

EMSTrackNum = eMSTrackNum;

}

public double getPureWeight() {

return pureWeight;

}

public void setPureWeight(double pureWeight) {

this.pureWeight = pureWeight;

}

public double getRoughWeight() {

return roughWeight;

}

public void setRoughWeight(double roughWeight) {

this.roughWeight = roughWeight;

}

public double getNumber() {

return number;

}

public void setNumber(double number) {

this.number = number;

}

public String getMainGoods() {

return mainGoods;

}

public void setMainGoods(String mainGoods) {

this.mainGoods = mainGoods;

}

public String getReceiName() {

return receiName;

}

public void setReceiName(String receiName) {

this.receiName = receiName;

}

public String getReceiProvinCode() {

return receiProvinCode;

}

public void setReceiProvinCode(String receiProvinCode) {

this.receiProvinCode = receiProvinCode;

}

public String getReceiAddr() {

return receiAddr;

}

public void setReceiAddr(String receiAddr) {

this.receiAddr = receiAddr;

}

public String getReceiTel() {

return receiTel;

}

public void setReceiTel(String receiTel) {

this.receiTel = receiTel;

}

public String getConsignerName() {

return consignerName;

}

public void setConsignerName(String consignerName) {

this.consignerName = consignerName;

}

public String getConsignerProvinCode() {

return consignerProvinCode;

}

public void setConsignerProvinCode(String consignerProvinCode) {

this.consignerProvinCode = consignerProvinCode;

}

public String getConsignerAddr() {

return consignerAddr;

}

public void setConsignerAddr(String consignerAddr) {

this.consignerAddr = consignerAddr;

}

public String getConsignerTel() {

return consignerTel;

}

public void setConsignerTel(String consignerTel) {

this.consignerTel = consignerTel;

}

public String getNote() {

return note;

}

public void setNote(String note) {

this.note = note;

}

public Information(String orderNum, String eMSTrackNum, double pureWeight, double roughWeight, double number,

String mainGoods, String receiName, String receiProvinCode, String receiAddr, String receiTel,

String consignerName, String consignerProvinCode, String consignerAddr, String consignerTel, String note) {

super();

this.orderNum = orderNum;

EMSTrackNum = eMSTrackNum;

this.pureWeight = pureWeight;

this.roughWeight = roughWeight;

this.number = number;

this.mainGoods = mainGoods;

this.receiName = receiName;

this.receiProvinCode = receiProvinCode;

this.receiAddr = receiAddr;

this.receiTel = receiTel;

this.consignerName = consignerName;

this.consignerProvinCode = consignerProvinCode;

this.consignerAddr = consignerAddr;

this.consignerTel = consignerTel;

this.note = note;

}

public Information() {

super();

}

public Information(double roughWeight,double number,String mainGoods) {

this.roughWeight=roughWeight;

this.number=number;

this.mainGoods=mainGoods;

}

@Override

public String toString() {

return "Information [orderNum=" + orderNum + ", EMSTrackNum=" + EMSTrackNum + ", pureWeight=" + pureWeight

+ ", roughWeight=" + roughWeight + ", number=" + number + ", mainGoods=" + mainGoods + ", receiName="

+ receiName + ", receiProvinCode=" + receiProvinCode + ", receiAddr=" + receiAddr + ", receiTel="

+ receiTel + ", consignerName=" + consignerName + ", consignerProvinCode=" + consignerProvinCode

+ ", consignerAddr=" + consignerAddr + ", consignerTel=" + consignerTel + ", note=" + note + "]\n";

}

}

### 2.2.3 driver.class文件

package Driver;

import java.io.IOException;

import java.util.ArrayList;

import Service.ImgToPdf;

import Service.QRcode;

import Service.imageControl;

import Service.jxlControl;

import base.Information;

public class driver {

public static void main(String[] args) {

String format=base.global.FORMAT;

String fileName =base.global.FILENAME;

ArrayList<Information>list = jxlControl.readFromExcel(fileName);

System.out.println("导出完成");

imageControl.control(format, list);

System.out.println("图片生成完成，在export中查看，只有前三张是结果，后面只是一维码");

for(int i=1;i<4;i++) {

try {

String imgSrc=base.global.PICPLACE+base.global.TAG+String.valueOf(i)+base.global.FORMAT;

String pdfSrc = base.global.PICPLACE+"pdf"+String.valueOf(i)+".pdf";

ImgToPdf.imgToPdf(imgSrc,pdfSrc);

System.out.println(pdfSrc+"导出完成");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

### 2.2.4 ImgCompressUtil.class文件

设计整个表单的样式

package Driver;

import java.awt.Color;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import javax.imageio.IIOImage;

import javax.imageio.ImageIO;

import javax.imageio.ImageTypeSpecifier;

import javax.imageio.ImageWriter;

import javax.imageio.metadata.IIOMetadata;

import javax.imageio.plugins.jpeg.JPEGImageWriteParam;

import javax.imageio.stream.ImageOutputStream;

import org.w3c.dom.Element;

public class ImgCompressUtil {

public static void main(String args[]){

String url = "F:\\FMS\_REPLAY\_MEDIA\\img\\";

String name = "sd456asd.gif";

name = "9d8608c7ea288a691b03ae5e0604de06.jpg";

Tosmallerpic(url,url,name,188,165,(float)0.85);

System.out.println("");

}

private static void Tosmallerpic(String srcFilePath,String destFilePath,String name,int w,int h,float per){

Image src;

try {

src = javax.imageio.ImageIO.read(new File(srcFilePath+File.separator+name)); //构造Image对象

String img\_midname = destFilePath + File.separator + "c\_"+name;

int old\_w = src.getWidth(null);

int old\_h = src.getHeight(null);

int new\_w = 0;

int new\_h = 0;

double w2 = (old\_w\*1.00)/(w\*1.00);

double h2 = (old\_h\*1.00)/(h\*1.00);

BufferedImage oldpic;

if(old\_w>old\_h)

{

oldpic = new BufferedImage(old\_w,old\_w,BufferedImage.TYPE\_INT\_RGB);

}else{if(old\_w<old\_h){

oldpic = new BufferedImage(old\_h,old\_h,BufferedImage.TYPE\_INT\_RGB);

}else{

oldpic = new BufferedImage(old\_w,old\_h,BufferedImage.TYPE\_INT\_RGB);

}

}

Graphics2D g = oldpic.createGraphics();

g.setColor(Color.white);

if(old\_w>old\_h)

{

g.fillRect(0, 0, old\_w, old\_w);

g.drawImage(src, 0, (old\_w - old\_h) / 2, old\_w, old\_h, Color.white, null);

}else{

if(old\_w<old\_h){

g.fillRect(0,0,old\_h,old\_h);

g.drawImage(src, (old\_h - old\_w) / 2, 0, old\_w, old\_h, Color.white, null);

}else{

g.drawImage(src.getScaledInstance(old\_w, old\_h, Image.SCALE\_SMOOTH), 0,0,null);

}

}

g.dispose();

src = oldpic;

if(old\_w>w)

new\_w = (int)Math.round(old\_w/w2);

else

new\_w = old\_w;

if(old\_h>h)

new\_h = (int)Math.round(old\_h/h2);

else

new\_h = old\_h;

BufferedImage image\_to\_save = new BufferedImage(new\_w,new\_h,BufferedImage.TYPE\_INT\_RGB);

image\_to\_save.getGraphics().drawImage(src.getScaledInstance(new\_w, new\_h, Image.SCALE\_SMOOTH), 0,0,null);

FileOutputStream fos = new FileOutputStream(img\_midname);

saveAsJPEG(100, image\_to\_save, per, fos);

fos.close();

} catch (IOException ex) {

}

}

public static void saveAsJPEG(Integer dpi ,BufferedImage image\_to\_save, float JPEGcompression, FileOutputStream fos) throws IOException {

ImageWriter imageWriter = ImageIO.getImageWritersBySuffix("jpg").next();

ImageOutputStream ios = ImageIO.createImageOutputStream(fos);

imageWriter.setOutput(ios);

IIOMetadata imageMetaData = imageWriter.getDefaultImageMetadata(new ImageTypeSpecifier(image\_to\_save), null);

if(dpi != null && !dpi.equals("")){

Element tree = (Element) imageMetaData.getAsTree("javax\_imageio\_jpeg\_image\_1.0");

Element jfif = (Element)tree.getElementsByTagName("app0JFIF").item(0);

jfif.setAttribute("Xdensity", Integer.toString(dpi) );

jfif.setAttribute("Ydensity", Integer.toString(dpi));

}

if(JPEGcompression >= 0 && JPEGcompression <= 1f){

JPEGImageWriteParam jpegParams = (JPEGImageWriteParam) imageWriter.getDefaultWriteParam();

jpegParams.setCompressionMode(JPEGImageWriteParam.MODE\_EXPLICIT);

jpegParams.setCompressionQuality(JPEGcompression);

}

imageWriter.write(imageMetaData, new IIOImage(image\_to\_save, null, null), null);

ios.close();

imageWriter.dispose();

}

}

### 2.2.5 imageControl.class文件

首先提取提取出一个人的信息交给下一层。然后画出二维码样式以及物件信息的排版

package Service;

public class imageControl {

public static void control(String format,ArrayList<Information>list) {

ArrayList<Information> newList= new ArrayList<Information>();

int i=0;

int beginCount=0;

while(i<list.size()) {

newList.add(list.get(i));

i++;

if(i==7)

break;

while(list.get(i).getEMSTrackNum()==null){

newList.add(list.get(i));

i++;

}

String srcImgPath = OrderControl.orderProduct(newList.get(0).getOrderNum(),format);

createimage(srcImgPath,newList);

newList.clear();

}

}

public static void createimage(String srcImgPath, ArrayList<Information>list) {

float d= (float) 0.7;

int realHeight = 200;

int textStandard =15;

int beginInterval=20;

int halfwid=(int) base.global.IMAGEWIDTH/2;

ImageObserver observer = null;

int gap=50;

Image Img = null;

BufferedImage image = new BufferedImage(base.global.IMAGEWIDTH,base.global.IMAGEHEIGHT, BufferedImage.TYPE\_INT\_RGB);

Graphics graphics = image.getGraphics();

graphics.setColor(Color.white);

graphics.fillRect(0, 0, base.global.IMAGEWIDTH,base.global.IMAGEHEIGHT);

graphics.setColor(Color.black);

graphics.setFont(new Font("微软雅黑", Font.ITALIC, 50));

graphics.drawString("EMS",70,100);

graphics.setFont(new Font("微软雅黑", Font.ITALIC, 25));

graphics.drawString("E标准", 85, 135);

BufferedImage QRcodeImage=QRcode.createCode(list.get(0).getEMSTrackNum(),300,300);

graphics.drawImage(QRcodeImage,625,1050,300,300, observer);

graphics.setFont(new Font("Arial", Font.LAYOUT\_LEFT\_TO\_RIGHT, 12));

for(int i=0;i<5;i++)

graphics.drawLine(i, 0, i,base.global.IMAGEHEIGHT);

for(int i=0;i<5;i++)

graphics.drawLine(0, i, base.global.IMAGEWIDTH,i);

for(int i=0;i<5;i++)

graphics.drawLine(0,base.global.IMAGEHEIGHT,base.global.IMAGEWIDTH, base.global.IMAGEHEIGHT-i);

for(int i=0;i<5;i++)

graphics.drawLine(base.global.IMAGEWIDTH-i, 0, base.global.IMAGEWIDTH-i, base.global.IMAGEHEIGHT);

for(int i=0;i<5;i++)

graphics.drawLine(0, realHeight-i, base.global.IMAGEWIDTH, realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(0, 2\*realHeight-i,base.global.IMAGEWIDTH, 2\*realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(0, 3\*realHeight-i, base.global.IMAGEWIDTH, 3\*realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(0, 4\*realHeight-i,base.global.IMAGEWIDTH, 4\*realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(0, 5\*realHeight-i, base.global.IMAGEWIDTH, 5\*realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(0, 6\*realHeight-i,base.global.IMAGEWIDTH-2\*realHeight, 6\*realHeight-i);

for(int i=0;i<5;i++)

graphics.drawLine(base.global.IMAGEWIDTH-2\*realHeight,1000,base.global.IMAGEWIDTH-2\*realHeight,1400);

for(int i=0;i<5;i++)

graphics.drawLine(halfwid,3\*realHeight,halfwid,4\*realHeight);

graphics.setFont(new Font("微软雅黑", Font.ITALIC,25));

int j=1;

for(Information info:list) {

graphics.drawString(info.getMainGoods(), beginInterval,gap\*j+realHeight);

j++;

}

graphics.drawString("收件人:"+list.get(0).getReceiName(), beginInterval, 2\*realHeight+gap);

graphics.drawString("电话:"+list.get(0).getReceiTel(),halfwid+beginInterval, 2\*realHeight+gap);

graphics.drawString("地址:"+list.get(0).getReceiAddr(), beginInterval, 2\*realHeight+2\*gap);

graphics.drawString("邮编:"+list.get(0).getReceiProvinCode(), beginInterval, 2\*realHeight+3\*gap);

graphics.drawString("内件数量:"+list.get(0).getNumber()+"件", beginInterval,3\*realHeight+ gap);

graphics.drawString("总重量"+list.get(0).getRoughWeight()+"kg",beginInterval,3\*realHeight+2\*gap);

graphics.setFont(new Font("微软雅黑",Font.ITALIC,40));

graphics.drawString("全程", halfwid-5\*beginInterval, 3\*realHeight+gap);

graphics.drawString("陆运", halfwid-5\*beginInterval, 3\*realHeight+2\*gap);

graphics.setFont(new Font("微软雅黑",Font.ITALIC,25));

graphics.drawString("收件人/代收件人:",halfwid+ beginInterval,3\*realHeight+ gap/2);

graphics.drawString("签收日期:",halfwid+ beginInterval, 4\*realHeight-gap);

graphics.drawString(" 年",halfwid+12\* beginInterval, 4\*realHeight-gap);

graphics.drawString(" 月",halfwid+15\* beginInterval, 4\*realHeight-gap);

graphics.drawString(" 日",halfwid+18\* beginInterval, 4\*realHeight-gap);

String lsh=OrderControl.lsh(list.get(0).getOrderNum());

graphics.drawString("订单号", beginInterval, 4\*realHeight+gap);

graphics.drawString(lsh, 17\*beginInterval, 5\*realHeight-gap);

String tempaddr = OrderControl.DcodeProduct(lsh);

QRcode.Dencode(lsh, tempaddr, 600, 100, 100);

try {

FileInputStream fis = new FileInputStream(tempaddr);

Img=ImageIO.read(fis);

} catch (IOException e1) {

e1.printStackTrace();

}

graphics.drawImage(Img, 8\*beginInterval, 4\*realHeight+gap/2, 600, 100, observer);

graphics.drawString("发件人:"+list.get(0).getConsignerName(), beginInterval, 5\*realHeight+gap);

graphics.drawString("Tel:"+list.get(0).getConsignerTel(), halfwid-6\*beginInterval, 5\*realHeight+gap);

graphics.drawString("发件人省市区代码"+list.get(0).getConsignerProvinCode(), beginInterval, 5\*realHeight+2\*gap);

graphics.drawString("收件人："+list.get(0).getReceiName(),beginInterval,6\*realHeight+gap);

graphics.drawString(list.get(0).getReceiTel(), halfwid-6\*beginInterval, realHeight\*6+gap);

graphics.drawString("收件地址："+list.get(0).getReceiAddr().substring(0, 15), beginInterval, realHeight\*6+gap\*2);

graphics.drawString(list.get(0).getReceiAddr().substring(15), beginInterval, realHeight\*6+gap\*3);

lsh=list.get(0).getEMSTrackNum();

tempaddr = OrderControl.DcodeProduct(lsh);

QRcode.Dencode(lsh, tempaddr, 600, 100, 100);

try {

FileInputStream fis = new FileInputStream(tempaddr);

Img=ImageIO.read(fis);

}catch(Exception e) {

e.printStackTrace();

}

graphics.drawImage(Img, 16\*beginInterval, gap/2, 600, 100, observer);

graphics.drawString(lsh, 25\*beginInterval, realHeight-gap);

FileOutputStream fos;

try {

fos = new FileOutputStream(srcImgPath);

saveAsJPEG(100, image,d, fos);

} catch (Exception e) {

e.printStackTrace();

}

}

public static void saveAsJPEG(Integer dpi, BufferedImage image\_to\_save, float d, FileOutputStream fos)

throws IOException {

ImageWriter imageWriter = ImageIO.getImageWritersBySuffix("jpg").next();

ImageOutputStream ios = ImageIO.createImageOutputStream(fos);

imageWriter.setOutput(ios);

IIOMetadata imageMetaData = imageWriter.getDefaultImageMetadata(new ImageTypeSpecifier(image\_to\_save), null);

if (dpi != null && !dpi.equals("")) {

Element tree = (Element) imageMetaData.getAsTree("javax\_imageio\_jpeg\_image\_1.0");

Element jfif = (Element) tree.getElementsByTagName("app0JFIF").item(0);

jfif.setAttribute("Xdensity", Integer.toString(dpi));

jfif.setAttribute("Ydensity", Integer.toString(dpi));

}

if (d >= 0 && d <= 1f) {

JPEGImageWriteParam jpegParams = (JPEGImageWriteParam) imageWriter.getDefaultWriteParam();

jpegParams.setCompressionMode(JPEGImageWriteParam.MODE\_EXPLICIT);

jpegParams.setCompressionQuality(d);

}

imageWriter.write(imageMetaData, new IIOImage(image\_to\_save, null, null), null);

ios.close();

imageWriter.dispose();

}

}

### 2.2.6 ImgToPdf.class文件

将图片格式转化为pdf格式

package Service;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import com.lowagie.text.Document;

import com.lowagie.text.DocumentException;

import com.lowagie.text.Image;

import com.lowagie.text.PageSize;

import com.lowagie.text.pdf.PdfWriter;

public class ImgToPdf {

public static boolean imgToPdf(String imgFilePath, String pdfFilePath)throws IOException {

File file=new File(imgFilePath);

if(file.exists()){

Document document = new Document();

FileOutputStream fos = null;

try {

fos = new FileOutputStream(pdfFilePath);

PdfWriter.getInstance(document, fos);

document.addAuthor("学生");

document.addSubject("pdf制作");

document.setPageSize(PageSize.A4);

document.open();

Image image = Image.getInstance(imgFilePath);

float imageHeight=image.getScaledHeight();

float imageWidth=image.getScaledWidth();

int i=0;

while(imageHeight>500||imageWidth>500){

image.scalePercent(100-i);

i++;

imageHeight=image.getScaledHeight();

imageWidth=image.getScaledWidth();

}

image.setAlignment(Image.ALIGN\_CENTER);

document.add(image);

} catch (DocumentException de) {

System.out.println(de.getMessage());

} catch (IOException ioe) {

System.out.println(ioe.getMessage());

}

document.close();

fos.flush();

fos.close();

return true;

}else{

return false;

}

}

}

### 2.2.7 jxlControl.class文件

package Service;

public class jxlControl {

public static ArrayList<Information> readFromExcel(String fileName){

ArrayList<Information> list =new ArrayList<Information>();

try {

File file = new File(fileName);

Workbook workbook = Workbook.getWorkbook(file);

Sheet sheet = workbook.getSheet(0);

String array[]=new String [15];

for(int row = 1; row < sheet.getRows(); row++) {

for(int col = 0; col < sheet.getColumns();col++) {

Cell cell = sheet.getCell(col,row);

array[col]=cell.getContents();

}

double pureWeight=Double.parseDouble(array[2]);

double number = Double.parseDouble(array[4]);

Information info;

if(array[0].equals("")) {

String mainGoods=array[5];

info = new Information(pureWeight,number,array[5]);

}

else {

double roughWeight=Double.parseDouble(array[3]);

info =new Information(array[0],array[1],pureWeight,

roughWeight,number,array[5],array[6],array[7],

array[8],array[9],array[10],array[11],array[12],

array[13],array[14]);

}

list.add(info);

}

}

catch(Exception e) {

e.printStackTrace();

}

return list;

}

}

### 2.2.8 OrderControl.class文件

package Service;

public class OrderControl {

public static String orderProduct(String orderNumber,String format) {

String src;

String head = base.global.PICPLACE;

src = head+base.global.TAG+orderNumber+format;

return src;

}

public static String QRcodeProduct(String orderNumber) {

String src;

String head = base.global.PICPLACE;

src=head+"物流单号"+orderNumber+"一维码.png";

return src;

}

public static String DcodeProduct(String orderNumber) {

String src;

String head = base.global.PICPLACE;

src=head+"物流单号"+orderNumber+"一维码.png";

return src;

}

public static String lsh (String lsh) {

Date data = new Date();

DateFormat format = new SimpleDateFormat("yyyyMMdd");

String time = format.format(data);

lsh=time+lsh;

return lsh;

}

public String getData(String a) {

Date date = new Date();

if (a == "cal") {

DateFormat format2 = new SimpleDateFormat("yyyyMMdd");

String time2 = format2.format(date);

return time2;

} else if (a == "time") {

DateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String time = format.format(date);

return time;

} else

return null;

}

}

### 2.2.9 QRcode.class文件

控制二维码内容位置大小的生成

package Service;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.EncodeHintType;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.WriterException;

import com.google.zxing.client.j2se.MatrixToImageWriter;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.qrcode.QRCodeWriter;

import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;

public class QRcode {

private static final int QRCOLOR = 0xFF000000;

private static final int BGCOLOR = 0xFFFFFFFF;

private static final String FORMAT = "PNG";

public static BufferedImage createCode(String qrUrl,int h,int w) {

MultiFormatWriter multiFormatWriter = null;

BitMatrix bm = null;

Map<EncodeHintType, Object> hints = getDecodeHintType();

BufferedImage image=new BufferedImage(w,h,BufferedImage.TYPE\_INT\_RGB);

try {

multiFormatWriter = new MultiFormatWriter();

bm = multiFormatWriter.encode(qrUrl, BarcodeFormat.QR\_CODE, w, h, hints);

for (int x = 0; x < w; x++) {

for (int y = 0; y < h; y++) {

image.setRGB(x, y, bm.get(x, y) ? QRCOLOR : BGCOLOR);

}

}

} catch (Exception e) {

e.printStackTrace();

}

return image;

}

private static Map<EncodeHintType, Object> getDecodeHintType() {

Map<EncodeHintType, Object> hints = new HashMap<EncodeHintType, Object>();

hints.put(EncodeHintType.ERROR\_CORRECTION, ErrorCorrectionLevel.H);

hints.put(EncodeHintType.CHARACTER\_SET, "utf-8");

hints.put(EncodeHintType.MARGIN, 0);

hints.put(EncodeHintType.MAX\_SIZE, 350);

hints.put(EncodeHintType.MIN\_SIZE, 100);

return hints;

}

public static void Dencode(String contents,String dest,int width,int height,int offset){

try{

contents=new String(contents.getBytes("UTF-8"),"ISO-8859-1");

BitMatrix matrix=new MultiFormatWriter().encode(contents,BarcodeFormat.CODE\_128,width-offset, height);

MatrixToImageWriter.writeToStream(matrix,FORMAT,new FileOutputStream(new File(dest)));

}catch(Exception e) {

e.printStackTrace();

}

}

public static void encode(String contents,String dest,int width,int height){

try{

contents=new String(contents.getBytes("UTF-8"),"ISO-8859-1");

QRCodeWriter writer=new QRCodeWriter();

BitMatrix matrix=writer.encode(contents, BarcodeFormat.QR\_CODE, width, height);

MatrixToImageWriter.writeToStream(matrix, FORMAT, new FileOutputStream(new File(dest)));

}catch(Exception e) {

e.printStackTrace();

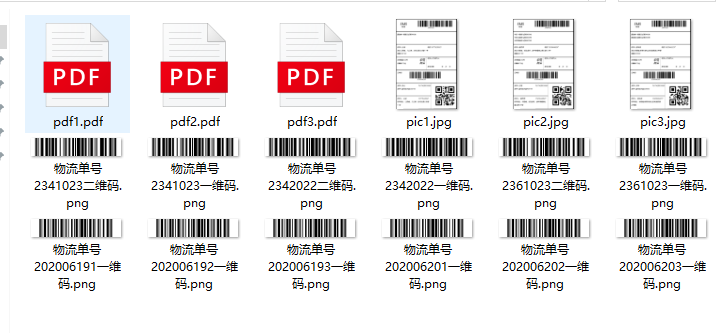
}

}

}

# **3 系统测试**

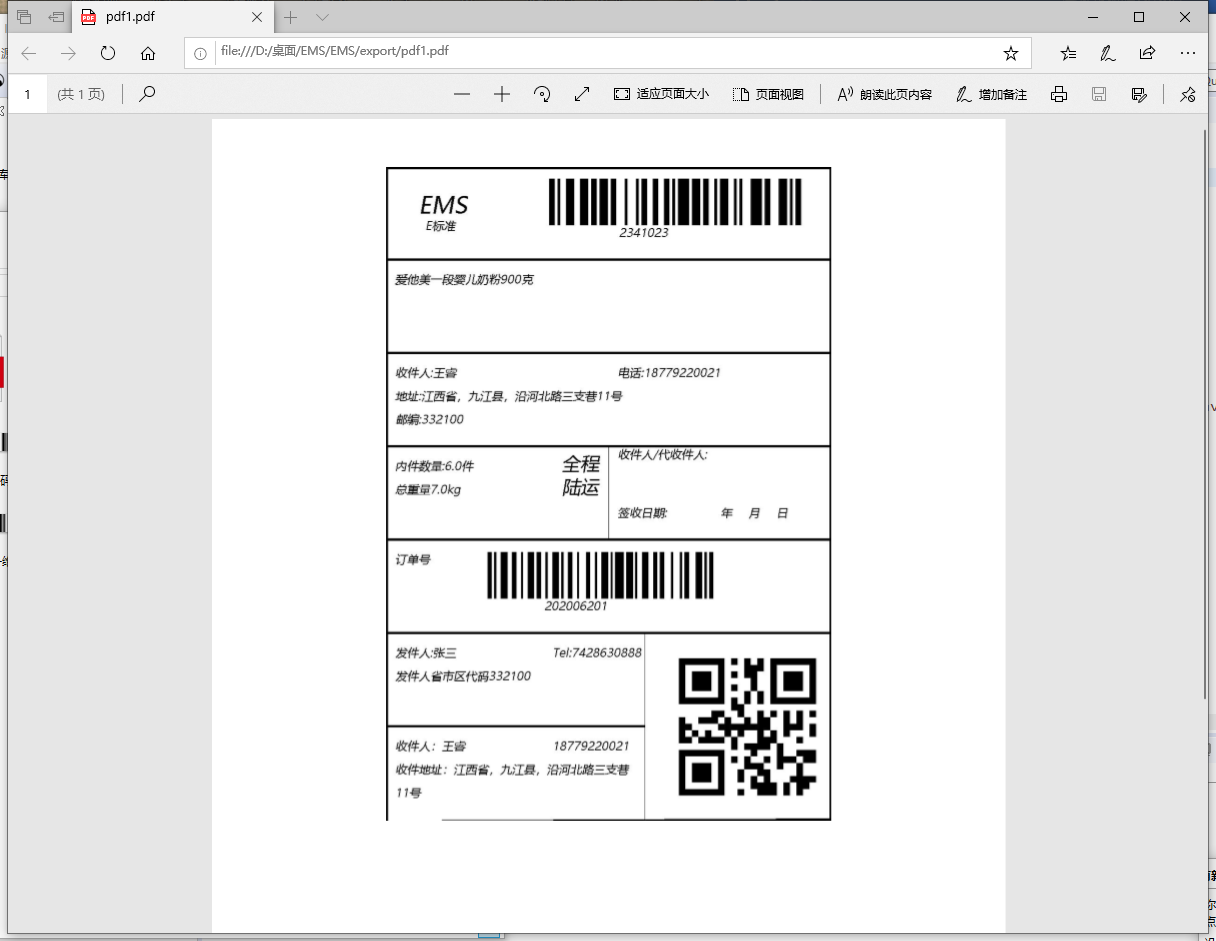
在export中生成的 EMS物流面单和pdf格式



4.1生成的 EMS物流面单



4.2转换为pdf格式



# **4 系统总结**

在生成物流面单的构造和一维码的生成遇到了问题 通过百度和教案查阅资料才勉强解决问题。通过这次实验我对Excel读取数据更加熟练了 对java的理解也更深刻了。但是还有很多不足：还可以有个选项对输出的数据选择图片文件或者是pdf文件 图片的格式也可以更多的选择。