**上海电力学院**

**.NET 课程设计**



**题 目:**  物流快递系统

**姓 名：** 黄海琨

**学 号：** 20131809

**院 系：** 计算机科学与技术学院

**专业年级：**计算机科学与技术（电企方向）/2013级

**2016年6月29日**

**目录**

[一、设计题目 1](#_Toc455016709)

[1.1实验题目 1](#_Toc455016710)

[1.2问题描述 1](#_Toc455016711)

[1.3基本要求 1](#_Toc455016712)

[二、需求分析 3](#_Toc455016713)

[2.1功能描述 3](#_Toc455016714)

[2.2 功能模块 3](#_Toc455016715)

[2.3数据流图 4](#_Toc455016716)

[三、概要设计 6](#_Toc455016717)

[3.1 系统功能架构 6](#_Toc455016718)

[3.2 数据字典 6](#_Toc455016719)

[四、系统设计 8](#_Toc455016720)

[4.1 系统总体设计 8](#_Toc455016721)

[4.2 系统数据模型设计 8](#_Toc455016722)

[4.3 系统实体联系图 8](#_Toc455016723)

[4.4 关系模型 9](#_Toc455016724)

[4.5 数据库表的设计 9](#_Toc455016725)

[五、详细设计 13](#_Toc455016726)

[5.1 连接数据库语句 13](#_Toc455016727)

[5.2 预约功能 13](#_Toc455016728)

[5.3 揽件功能 14](#_Toc455016729)

[5.4 配送安排功能 17](#_Toc455016730)

[5.5 派件签收功能 19](#_Toc455016731)

[六、测试 23](#_Toc455016732)

[6.1 测试内容 23](#_Toc455016733)

[6.2 测试用例设计 24](#_Toc455016734)

[七、总结 28](#_Toc455016735)

## 一、设计题目

### 1.1实验题目

物流快递系统

### 1.2问题描述

物流信息有着不同地方和数据量大的特点，里面有着业务流转，所以需要系统来支持物流的整个过程。

### 1.3基本要求

系统主要功能:

1. 电话预约
   * 用户打进电话留下相应的快递信息
   * 跟用户所属地区的业务员联系，确认取件时间
2. 物品信息管理

* 业务员上门取货，给物品标识一个流水号
* 填写详细物流信息

1. 发货管理

* 记录物品的发货时间、班次
* 支持查询发货信息

1. 仓库管理

* 记录不同地区的仓库每天入库出库物品信息
* 支持列出同一地区仓库的物品

1. 派送管理

* 记录业务员派送物品的具体信息

1. 跟踪管理

* 支持用户根据流水号查询物品所处状态

1. 投诉管理

* 记录用户的反馈信息及处理结果

## 二、需求分析

### 2.1功能描述

2.1.1电话预约

2.1.2物品信息管理

2.1.3发货管理

2.1.4仓库管理

2.1.5派送管理

2.1.6跟踪管理

2.1.7投诉管理

### 2.2 功能模块

#### 2.2.1 电话预约

* + 用户打进电话留下相应的快递信息
  + 跟用户所属地区的业务员联系，确认取件时间

#### 2.2.2 物品信息管理

* 业务员上门取货，给物品标识一个流水号
* 填写详细物流信息

#### 2.2.3 发货管理

* 记录物品的发货时间、班次
* 支持查询发货信息

#### 2.2.4 仓库管理

* 记录不同地区的仓库每天入库出库物品信息
* 支持列出同一地区仓库的物品

#### 2.2.5 派送管理

* 记录业务员派送物品的具体信息

#### 2.2.6 跟踪管理

* 支持用户根据流水号查询物品所处状态

#### 2.2.7 投诉管理

* 记录用户的反馈信息及处理结果

### 2.3数据流图

系统分为用户、业务员、网点、仓库、快递员四种，如图2-1所示

用户通过电话与业务员取得联系，业务员填写预约信息，系统自动发送信息给网点，通知快递员上门取件。快递员上门取件完毕后录入快递详情信息，生成快递单号给寄件人，网点集中发货到仓库，仓库确认收件，系统自动录入仓库信息，等待出库时间。出库后系统通知网点接收，网点接收后系统自动安排配送快递员。快递员接收派件任务，上门派件，确认收货或者退件。



图2-1 物流快递系统需求分析顶层图

如图2-2所示，用户在取得快递单号后可以随时查询快递信息

图2-2 物流快递系统需求分析0层图

## 三、概要设计

### 3.1 系统功能架构

物流快递系统分为预约、揽件、发货、仓储、派件、查询和登录七大块

* 预约功能：预约信息填写
* 揽件功能：上门揽件；填写货品基本信息
* 留言功能：揽件确认；发货确认；发货查询
* 仓储功能：入库；出库
* 派件功能：核实收件人信息；上门派件；
* 查询功能：查询快递信息；投诉；
* 登录功能：身份核实；权限分配；



图3-1 物流快递系统结构图

### 3.2 数据字典

数据字典是系统分析阶段的重要文档，它清楚地定义与详细地解释了数据流程图上不能详细表达的内容，同时它也是同用户交谈的工具。

下面详细列出本系统中的数据字典：

表3-1货品基本信息数据流条目

|  |  |
| --- | --- |
| 数据流名称 | 货品基本信息 |
| 别名 | 快递单信息 |
| 简述 | 货品的所有基本信息 |
| 数据流来源 | 网点 |
| 数据流去向 | 加工3录入 |
| 数据流组成 | 快递单号+始发地国家+始发地省份+始发地城市+始发地区县+始发地乡镇街道+始发地地址+始发地邮编+寄件人姓名+寄件人地址+寄件人电话+目的地国家+目的地省份+目的地城市+目的地区县+目的地乡镇街道+目的地地址+目的地邮编+收件人姓名+收件人地址+收件人电话+付款方式+品名+重量+保险+配送方式+运费+创建时间 |
| 流通量 | 10000/天 |
| 表3-2寄件预约单数据流条目 | |
| 数据流名称 | 寄件预约单 |
| 别名 | 快递预约揽件单 |
| 简述 | 寄快递时的预约信息 |
| 数据流来源 | 寄件人 |
| 数据流去向 | 加工1预约 |
| 数据流组成 | 预约单号+预约人姓名+预约人电话+预约人地址+货品类型+目的地+揽件时间+创建时间 |
| 流通量 | 10000/天 |
| 表3-3物流信息数据流条目 | |
| 数据流名称 | 物流信息 |
| 别名 | 无 |
| 简述 | 根据快递单号查询物流信息 |
| 数据流来源 | 加工11实时查询 |
| 数据流去向 | 寄件人、收件人 |
| 数据流组成 | 快递单号+揽件网点+揽件员工号+一级分拣中转站+运输点+二级分拣中转站+派件网点+派件员工号+签收状态 |
| 流通量 | 90000/天 |
| 表3-4快递单号数据流条目 | |
| 数据流名称 | 快递单号 |
| 别名 | 无 |
| 简述 | 根据快递单号查询物流信息的快递单号 |
| 数据流来源 | 寄件人、收件人 |
| 数据流去向 | 加工11实时查询 |
| 数据流组成 | 快递单号+揽件网点+揽件员工号+一级分拣中转站+运输点+二级分拣中转站+派件网点+派件员工号+签收状态 |
| 流通量 | 90000/天 |

## 四、系统设计

### 4.1 系统总体设计

采用ASP.NET来创建这个网站，采用MVC架构来进行业务逻辑，Controller，View，Model和其他Bootstrap的空间搭建网页的后台和前端。

### 4.2 系统数据模型设计

系统主要设计的实体有：预约单、详细物流信息、发货信息、仓储信息、库存信息、派送信息、流程单、投诉单、用户表

### 4.3 系统实体联系图

系统中的实体关系如图4-1所示

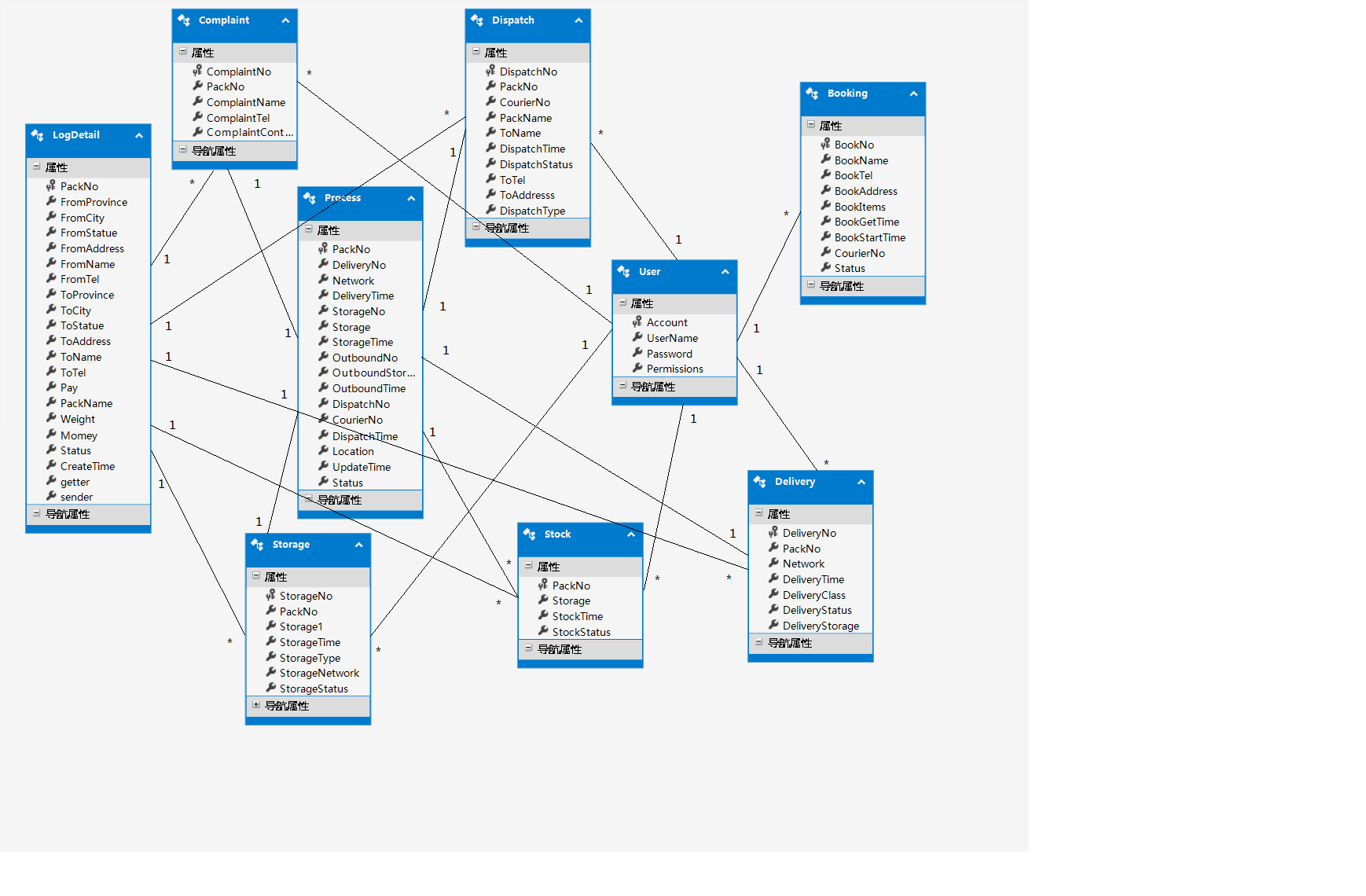


图4-1 实体联系ER图

### 4.4 关系模型

* 预约单（预约单号、预约人姓名、预约人电话、预约人地址、货品内容、揽件时间、创建时间）
* 详细物流信息（快递单号、始发地省份、始发地城市、始发地区县、始发地地址、寄件人姓名、寄件人电话、目的地省份、目的地城市、目的地区县、目的地地址、收件人姓名、收件人电话、付款方式、品名、重量、运费、状态、创建时间、揽件人、派送人）
* 发货信息（发货单号、快递单号、网点号、发货时间、发货班次、发往仓库、状态）
* 仓储信息（仓储单号、快递单号、仓库号、发往网点、创建时间、发送状态、仓储单类型）
* 库存信息（快递单号、仓库号、创建时间、状态）
* 派送信息（派送单号、快递单号、快递员工号、物品名、联系电话、收件人、派送地址、派送时间、派送类型、派送状态）
* 流程单（快递单号、发货单号、发货网点、发货时间、入库单号、入库仓库、入库时间、出库单号、出库仓库、出库时间、派送单号、派送人员、派送时间、当前位置、更新时间、状态）
* 投诉单（投诉单号、投诉快递单号、投诉人姓名、投诉人电话、投诉内容）
* 用户表（用户名、密码、用户名称、权限）

### 4.5 数据库表的设计

表4-1 预约单数据库表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 预约单 | Booking | BookNo | 预约单号 | int |  | Y |
| BookName | 预约人姓名 | nvarchar | 30 | N |
| BookTel | 预约人电话 | int |  | N |
| BookAddress | 预约人地址 | nvarchar | 30 | N |
| BookItems | 货品内容 | nvarchar | 150 | N |
| BookGetTime | 揽件时间 | datetime |  | N |
| BookStartTime | 创建时间 | datetime |  | N |
| 表4-2详细物流信息数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 详细物流信息 | LogDetail | PackNo | 快递单号 | int |  | Y |
| FromProvince | 始发地省份 | nvarchar | 10 | N |
| FromCity | 始发地城市 | nvarchar | 10 | N |
| FromStatue | 始发地区县 | nvarchar | 10 | N |
| FromAddress | 始发地地址 | nvarchar | 50 | N |
| FromName | 寄件人姓名 | nvarchar | 10 | N |
| FromTel | 寄件人电话 | int |  | N |
| ToProvince | 目的地省份 | nvarchar | 10 | N |
| ToCity | 目的地城市 | nvarchar | 10 | N |
| ToStatue | 目的地区县 | nvarchar | 10 | N |
| ToAddress | 目的地地址 | nvarchar | 50 | N |
| ToName | 收件人姓名 | nvarchar | 10 | N |
| ToTel | 收件人电话 | int |  | N |
| Pay | 付款方式 | nvarchar | 10 | N |
| PackName | 品名 | nvarchar | 10 | N |
| Weight | 重量 | double |  | N |
| Momey | 运费 | double |  | N |
| Status | 状态 | int |  | N |
| CreateTime | 创建时间 | datetime |  | N |
| getter | 揽件人 | int |  | N |
| sender | 派送人 | int |  | N |
| 表4-3 发货信息数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 发货信息 | Delivery | DeliveryNo | 发货单号 | int |  | Y |
| PackNo | 快递单号 | int |  | N |
| Network | 网点号 | int |  | N |
| DeliveryTime | 发货时间 | datetime |  | N |
| DeliveryClass | 发货班次 | int |  | N |
| DeliveryStorage | 发往仓库 | int |  | N |
| DeliveryStatus | 状态 | int |  | N |
| 表4-4 仓储信息数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 仓储信息 | Storage | StorageNo | 仓储单号 | int |  | Y |
| PackNo | 快递单号 | int |  | N |
| Storage | 仓库号 | int |  | N |
| StorageNetwork | 发往网点 | int |  | N |
| StorageTime | 创建时间 | datetime |  | N |
| StorageStatus | 发送状态 | int |  | N |
| StorageType | 仓储单类型 | int |  | N |
| 表4-5 库存信息数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 库存信息 | Stock | PackNo | 快递单号 | int |  | Y |
| Storage | 仓库号 | int |  | N |
| StockTime | 创建时间 | datetime |  | N |
| StockStatus | 状态 | int |  | N |
| 表4-6 派送信息数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 派送信息 | Dispatch | DispatchNo | 派送单号 | int |  | Y |
| PackNo | 快递单号 | int |  | N |
| CourierNo | 快递员工号 | int |  | N |
| PackName | 物品名 | nvarchar | 30 | N |
| ToTel | 联系电话 | int |  | N |
| ToName | 收件人 | nvarchar | 20 | N |
| ToAddress | 派送地址 | nvarchar | 50 | N |
| DispatchTime | 派送时间 | datetime |  | N |
| DispatchType | 派送类型 | int |  | N |
| DispatchStatus | 派送状态 | int |  | N |
| 表4-7 流程单数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 流程单 | Process | PackNo | 快递单号 | int |  | Y |
| DeliveryNo | 发货单号 | int |  | N |
| Network | 发货网点 | nvarchar | 50 | N |
| DeliveryTime | 发货时间 | datetime |  | N |
| StorageNo | 入库单号 | int |  | N |
| Storage | 入库仓库 | nvarchar | 50 | N |
| StorageTime | 入库时间 | datetime |  | N |
| OutboundNo | 出库单号 | int |  | N |
| OutboundStorage | 出库仓库 | nvarchar | 50 | N |
| OutboundTime | 出库时间 | datetime |  | N |
| DispatchNo | 派送单号 | int |  | N |
| CourierNo | 派送人员 | nvarchar | 20 | N |
| DispatchTime | 派送时间 | datetime |  | N |
| Location | 当前位置 | nvarchar | 50 | N |
| UpdateTime | 更新时间 | datetime |  | N |
| Status | 状态 | int |  | N |
| 表4-8 投诉单数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 投诉单 | Complaint | ComplaintNo | 投诉单号 | int |  | Y |
| PackNo | 投诉快递单号 | int |  | N |
| ComplaintName | 投诉人姓名 | nvarchar | 20 | N |
| ComplaintTel | 投诉人电话 | int |  | N |
| ComplaintContent | 投诉内容 | nvarchar | 300 | N |
| 表4-9 用户表数据库表 | | | | | | |
| 表名 |  | 属性名 | 属性描述 | 属性类型 | 长度 | 主键 |
| 用户表 | User | Account | 用户名 | int |  | Y |
| Password | 密码 | nvarchar | 20 | N |
| UserName | 用户名称 | nvarchar | 30 | N |
| Permissions | 权限 | int |  | N |

## 五、详细设计

### 5.1 连接数据库语句

<connectionStrings>

<add name="ModelContext" connectionString="server=121.42.197.31;user id=hhk;password=1234;database=Logistics" providerName="MySql.Data.MySqlClient" />

</connectionStrings>

### 5.2 预约功能

在物流系统中，寄件人通过系统进行预约，填写自己的个人信息，系统会安排快递员上门来揽件，在预约的时候可以修改自己的预约要求，约定上门时间。如图5-1所示算法。



图5-1 预约揽件过程流程图

**【代码】**

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult CreateBooking(CommunicateModel model)

{

model.BookingModel.BookStartTime = System.DateTime.Now;

model.BookingModel.CourierNo = 2;

model.BookingModel.Status = 0;

db.Booking.Add(model.BookingModel);

db.SaveChanges();

return RedirectToAction("../Home/ManagerIndex");

}

**【运行页面】**

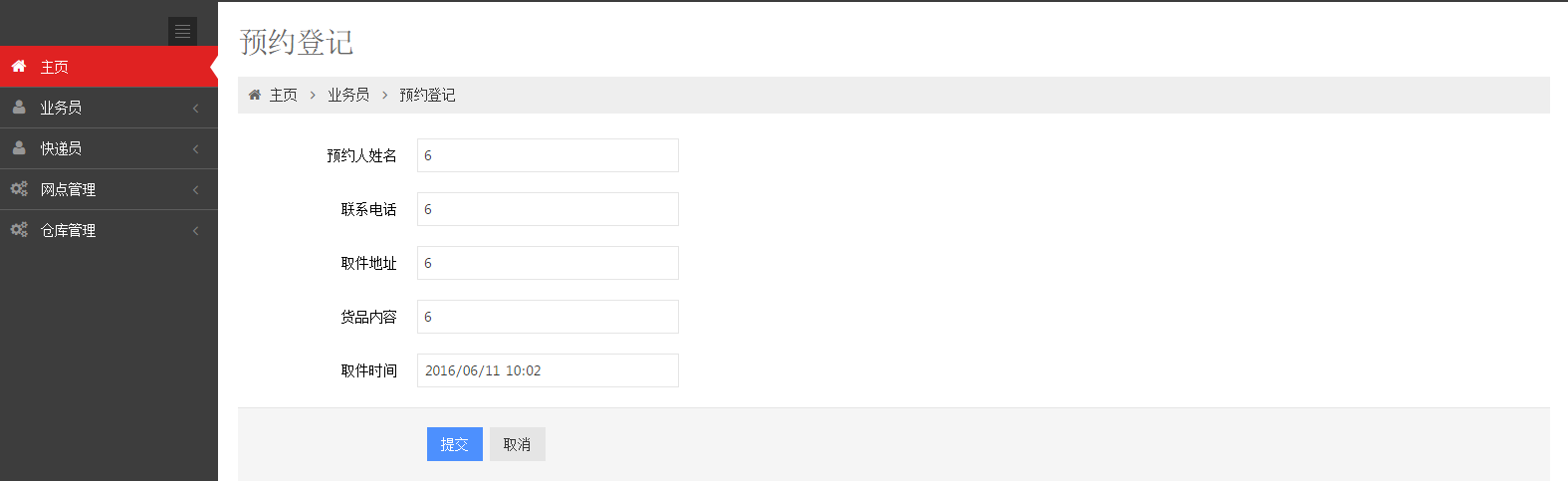


图5-2 预约功能页面

### 5.3 揽件功能

在物流系统中，等待预约过程完成后，需要进行揽件，系统发送揽件通知给网点，网点根据快递员的职能范围派发任务给快递员进行揽件，快递员在揽件的时候先核实预约的信息，然后上面进行揽件，确定快件的类型是否符合要求，如符合进行揽件，登入系统，系统把生成的快递单号发给寄件人和收件人。如图5-3所示算法。



图5-3 揽件功能流程图

**【代码】**

public ActionResult GetPackage()

{

int account = (int)Session["Account"];

var list = from a in db.Booking where a.CourierNo == account orderby a.Status, a.BookGetTime ascending select a;

CourierModel model = new CourierModel();

model.BookingModel = list.ToList();

return View(model);

}

public ActionResult GetPackageMethod(int? id)

{

Booking booking = db.Booking.Find(id);

booking.Status = 1;

db.Entry(booking).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("GetPackage");

}

[HttpGet]

public ActionResult AddPackageInfo()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult AddPackageInfo(CourierModel model)

{

String fromer = model.LogDetailModel.FromName;

DateTime datetime = System.DateTime.Now;

model.LogDetailModel.CreateTime = datetime;

model.LogDetailModel.Status = 0;

model.LogDetailModel.getter = (int)Session["Account"];

db.LogDetail.Add(model.LogDetailModel);

db.SaveChanges();

int packno = model.LogDetailModel.PackNo;

return Content("<script>alert('快递单号为：" + packno + " 请牢记！');location.href='../CourierModels/AddPackageInfo';</script>");

}

**【运行页面】**

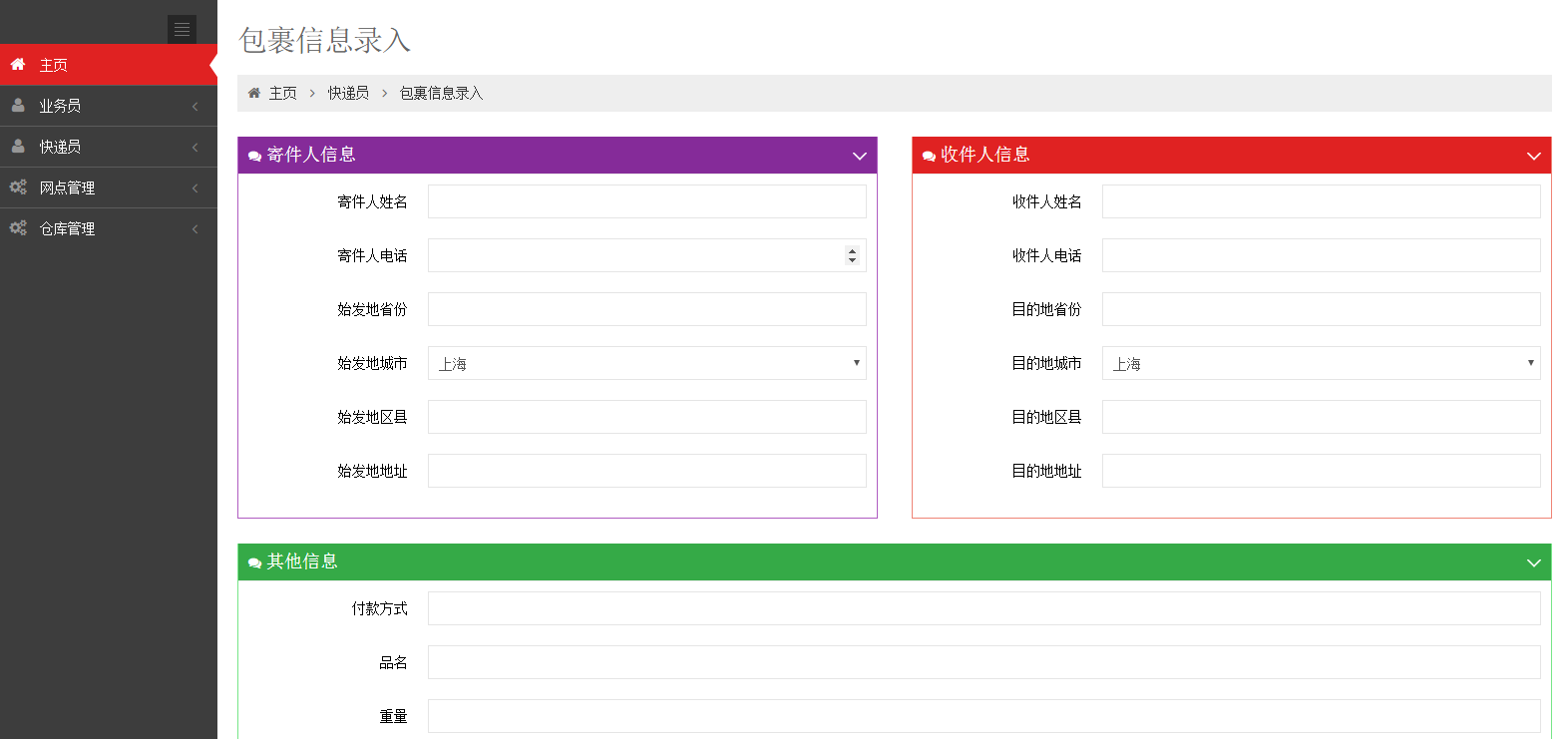


图5-4 揽件功能页面

### 5.4 配送安排功能

在仓库中的货品出库后，网点可以等待货品到达后，点击确定，然后系统自动安排快递员进行配送。



图5-5 配送安排功能流程图

**【代码】**

public ActionResult SendManager()

{

int account = (int)Session["Account"];

var list = from a in db.Storage where a.StorageNetwork == account orderby a.StorageStatus, a.StorageTime ascending select a;

NetWorkModel model = new NetWorkModel();

model.StorageModel = list.ToList();

return View(model);

}

public ActionResult SendManagerMethod(int? id)

{

int RandKey = 0;

int account = (int)Session["Account"];

User user = db.User.Find(account);

if(account == 6)

{

Random ran = new Random();

RandKey = ran.Next(2, 3);

}

if (account == 7)

{

Random ran = new Random();

RandKey = ran.Next(4, 5);

}

Storage storage = db.Storage.Find(id);

LogDetail logDetail = db.LogDetail.Find(storage.PackNo);

//快递单信息更改

logDetail.sender = RandKey;

db.Entry(logDetail).State = EntityState.Modified;

db.SaveChanges();

//发货信息状态更改

storage.StorageStatus = 1;

db.Entry(storage).State = EntityState.Modified;

db.SaveChanges();

//派送信息填写

Dispatch dispatch = new Dispatch();

dispatch.PackNo = logDetail.PackNo;

dispatch.PackName = logDetail.PackName;

dispatch.ToName = logDetail.ToName;

dispatch.ToTel = logDetail.ToTel;

dispatch.ToAddresss = logDetail.ToAddress;

dispatch.CourierNo = RandKey;

dispatch.DispatchTime = System.DateTime.Now;

dispatch.DispatchStatus = 0;

dispatch.DispatchType = logDetail.Status;

db.Dispatch.Add(dispatch);

db.SaveChanges();

int dispatchNo = dispatch.DispatchNo;

//流程信息填写

Process process = db.Process.Find(logDetail.PackNo);

process.DispatchNo = dispatchNo;

process.DispatchTime = dispatch.DispatchTime;

process.CourierNo = dispatch.CourierNo.ToString();

process.Location = user.UserName;

process.UpdateTime = System.DateTime.Now;

db.Entry(process).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("../NetWorkModels/SendManager");

}

**【运行页面】**



图5-6 配送安排页面

### 5.5 派件签收功能

在物流系统中，等待运输过程完成后，根据货品和快递单号，快递员核实收件人信息，如果信息正确上门派件，如果不正确进行退件处理。查看货品是否正确完整，如果正确完整查看是否要付费，如果需要就付费然后签收。如果货品损坏做退件保险处理。



图5-7 派件签收功能流程图

**【代码】**

public ActionResult SendPackage()

{

int account = (int)Session["Account"];

var list = from a in db.Dispatch where a.CourierNo == account orderby a.DispatchStatus, a.DispatchTime ascending select a;

CourierModel model = new CourierModel();

model.DispatchtModel = list.ToList();

return View(model);

}

public ActionResult AccpetSendPackageMethod(int? id)

{

//派送单信息修改

Dispatch dispatch = db.Dispatch.Find(id);

dispatch.DispatchStatus = 1;

db.Entry(dispatch).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("../CourierModels/SendPackage");

}

public ActionResult FinishSendPackageMethod(int? id)

{

//派送单信息修改

Dispatch dispatch = db.Dispatch.Find(id);

dispatch.DispatchStatus = 2;

db.Entry(dispatch).State = EntityState.Modified;

db.SaveChanges();

//流程信息单修改

Process process = db.Process.Find(dispatch.PackNo);

if(process.Status == 0)

{

process.Status = 2;

process.UpdateTime = System.DateTime.Now;

db.Entry(process).State = EntityState.Modified;

db.SaveChanges();

}

if (process.Status == 1)

{

process.Status = 3;

process.UpdateTime = System.DateTime.Now;

db.Entry(process).State = EntityState.Modified;

db.SaveChanges();

}

return RedirectToAction("../CourierModels/SendPackage");

}

public ActionResult BackSendPackageMethod(int? id)

{

//派送单信息修改

Dispatch dispatch = db.Dispatch.Find(id);

dispatch.DispatchStatus = 3;

db.Entry(dispatch).State = EntityState.Modified;

db.SaveChanges();

//快递信息单修改

LogDetail logDetail = db.LogDetail.Find(dispatch.PackNo);

logDetail.Status = 1;

db.Entry(logDetail).State = EntityState.Modified;

db.SaveChanges();

//流程信息单修改

Process process = db.Process.Find(dispatch.PackNo);

process.Status = 1;

process.UpdateTime = System.DateTime.Now;

db.Entry(process).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("../CourierModels/SendPackage");

}

**【运行页面】**



图5-8 派件签收功能页面

## 六、测试

在程序完成后，进行程序调试是很重要的。好的程序是要对于键盘上输入的任何值都不会有出错，而测试的目的就是尽可能的发现程序中的错误，要考虑到尽可能多的情况。

### 6.1 测试内容

#### 6.1.1 预约功能测试

本模块测试目的：检测是否按正确的字段进行录入。预约人姓名、预约人电话、预约人地址、货品类型、目的地、揽件时间输入是否正确，数据库写入是否正确。

#### 6.1.2 揽件功能测试

本模块测试目的：检测是否按正确的字段进行录入。始发地国家、始发地省份、始发地城市、始发地区县、始发地乡镇街道、始发地地址、始发地邮编、寄件人姓名、寄件人地址、寄件人电话、目的地国家、目的地省份、目的地城市、目的地区县、目的地乡镇街道、目的地地址、目的地邮编、收件人姓名、收件人地址、收件人电话、付款方式、品名、重量、保险、配送方式、运费输入是否正确，数据库写入是否正确。

#### 6.1.3 查询功能测试

本模块测试目的：检测是否按正确的字段进行录入。快递单号输入是否正确，数据库查询是否正确、货品物流信息是否显示正确。

#### 6.1.4 登录功能测试

本模块测试目的：检测是否按正确的字段进行录入。用户名、密码输入是否正确，数据库写入是否正确，员工界面显示是否正确。

### 6.2 测试用例设计

#### 6.2.1预约功能测试

**a. 预约人姓名：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 姓名 | 英文字母、中文 | 1 | 英文字母中文混合 | 2 |
| 数字 | 3 |
| 符号 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 姓名 | 等价类 |
| 张三丰 | 1 |
| 黄kl | 2 |
| 12 | 3 |
| ！@ | 4 |

**b. 预约揽件日期：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 日期和时间 |  | 1 | Year<1985 | 2 |
|  | Year>2200 | 3 |
| Year：1985~2200 | Month<01 | 4 |
| Month： 01~12 | Month>12 | 5 |
| Day： 01~31 | Day<01 | 6 |
| Hour：00~23 | Day>31 | 7 |
| Minute：00~59 | Hour<00 | 8 |
|  | Hour>23 | 9 |
|  | Minute<00 | 10 |
|  | Minute>59 | 11 |

**测试用例：**

|  |  |
| --- | --- |
| 预约时间 | 等价类 |
| 2015.01.15 15:00 | 1 |
| 1984.01.13 15:00 | 2 |
| 2201.01.13 15:00 | 3 |
| 2015.00.15 15:00 | 4 |
| 2015.13.15 15:00 | 5 |
| 2015.01.00 15:00 | 6 |
| 2015.01.32 15:00 | 7 |
| 2015.01.15 -01:00 | 8 |
| 2015.01.15 25:00 | 9 |
| 2015.01.15 15:-10 | 10 |
| 2015.01.15 15:65 | 11 |

**c. 预约人电话：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 11位电话号码 | 11位数字 | 1 | 非数字 | 2 |
| <11位 | 3 |
| >11位 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 电话号码 | 等价类 |
| 18917022222 | 1 |
| aaaaa | 2 |
| 2222 | 3 |
| 189170298090 | 4 |

#### 6.2.2揽件功能测试

**a. 收件人姓名：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 姓名 | 英文字母、中文 | 1 | 英文字母中文混合 | 2 |
| 数字 | 3 |
| 符号 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 姓名 | 等价类 |
| 张三丰 | 1 |
| 黄kl | 2 |
| 12 | 3 |
| ！@ | 4 |

**b. 收件人电话：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 11位电话号码 | 11位数字 | 1 | 非数字 | 2 |
| <11位 | 3 |
| >11位 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 电话号码 | 等价类 |
| 18917022222 | 1 |
| aaaaa | 2 |
| 2222 | 3 |
| 189170298090 | 4 |

**c．目的地邮编：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 邮编 | 6位数字 | 1 | <6位 | 2 |
| >6位 | 3 |
| 非数字 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 邮编 | 等价类 |
| 201315 | 1 |
| 23 | 2 |
| 1234567 | 3 |
| #12 | 4 |

#### 6.2.3查询功能测试

**a. 快递单号：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 快递单号 | 8位数字 | 1 | 非数字字符 | 2 |
| <8位 | 3 |
| >8位 | 4 |

**测试用例：**

|  |  |
| --- | --- |
| 快递单号 | 等价类 |
| 12345678 | 1 |
| #111 | 2 |
| 1234 | 3 |
| 123456789086 | 4 |

#### 6.2.4登录功能测试

**a. 用户名：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 用户名 | 6~20位字母数字组合 | 1 | <6位 | 2 |
| >20位 | 3 |

**测试用例：**

|  |  |
| --- | --- |
| 用户名 | 等价类 |
| Time45 | 1 |
| 4e | 2 |
| mdddjdjdiwjdjsksifjendd | 3 |

**b. 密码：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 合理的等价类 | 编号 | 不合理的等价类 | 编号 |
| 密码 | 6~20位字母数字组合 | 1 | <6位 | 2 |
| >20位 | 3 |

**测试用例：**

|  |  |
| --- | --- |
| 密码 | 等价类 |
| Time45 | 1 |
| 4e | 2 |
| mdddjdjdiwjdjsksifjendd | 3 |

## 七、总结

在本次课程设计中，MVC的框架结果帮助了我了解MVC体系的业务逻辑处理和页面呈现。采用CodeFirst建立Model类型和使用Linq语句查询数据库的方法都是之前没有用过的。在页面呈现上，我采用了Bootstrap的二次开发工具实现页面的呈现。

在业务逻辑上，我采用Session的方法来判断用户的登录状态，在每个页面上我都采用了Filter的方法来判断用户的权限。在页面视图要采用多种Model类型的时候，我使用了ViewModel类型来再次定义Model使得在页面上可以实现两种以上Model类型的内容。

本系统的各个环节都有着业务逻辑，每一个环节都互相依赖。