《数据库系统原理》课程设计

系统实现报告

题目名称：“飞鸽”快递信息管理系统

13061168 仇栋民

学号及姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

13061162 史烨轩

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2015年12月27日

1. **系统功能需求分析**
2. **实现环境**

工程类型：C# Windows 窗体应用程序

IDE：Visual Studio Express 2013 for Windows Desktop

DBMS：MySQL 5.7.10

1. **功能需求分析**

**2.1注册功能**

使用本系统的人按其权限分为三种：客户、员工、管理员。其中只有用户是可以自行注册的，管理员账号有系统自动在建立数据库的时候初始化生成，员工由管理员在系统中通过增加员工功能来增加。

用户注册功能需求为：用户填写注册信息，包括用户账号、用户密码、用户姓名、用户性别、用户年龄、用户手机、用户邮箱，填写完成后，提交注册请求。其中用户账号为UNIQUE。由系统检验用户填写的注册信息是否符合要求，若填写正确，则将正确信息写入数据库。

**2.2登陆功能**

本系统的登陆功能分为用户登录和员工登陆（其中管理员登陆属于员工登陆）。

用户登录需求为：输入注册账号（用户账号为UNIQUE）和密码，提交登陆请求，系统验证用户身份，通过则登陆成功。

员工登陆需求为：输入员工号（即员工id）和密码，提交登陆请求，系统验证员工密码，通过则登陆成功。其中员工id为1的员工为系统管理员。

**2.3修改信息功能**

已经登陆系统的用户可以对自己的注册信息进行修改。

其需求为：从数据库中读取客户可修改信息（客户姓名、客户年龄、客户性别、客户手机、客户邮箱）或者员工可修改信息（员工姓名、员工性别、员工年龄、员工手机、员工邮箱、所属地点）显示到界面。用户可以对信息进行修改然后点击确认修改，系统检验信息的有效性，若正确则写入数据库。

**2.4修改密码功能**

对已经登陆系统的用户可以对自己的登陆密码进行修改。

其需求为：用户输入旧密码、新密码、确认新密码，提交请求，系统验证通过则写入数据库。

**2.5查询订单相关功能**

客户可以在系统中查询自己下的订单，以及订单的运输路径信息。

其需求为：客户点击查询按钮，系统查询出该客户的订单并显示在界面上。客户可以点击某个订单，系统查询该订单的运输路径信息并显示到界面上。

**2.6下单功能**

客户可以在系统中下订单。

其需求为：输入要快递的货物信息（货物名称、货物类别、货物重量）、出发地点以及目的地点，提交下单请求，系统检验信息有效性，若正确就写入数据库。

**2.7查询任务相关功能**

员工可以在系统中查询分配给自己的运输任务。

其需求为：员工点击查询任务，系统将分配给该员工的任务，分未完成和已完成两类进行显示。对未完成的任务，任务负责员工可以点击后，可以开始执行该任务或者完成该任务。

**2.8分配任务相关功能**

管理员可以查询系统中未分配的运输任务，并分配这些任务。

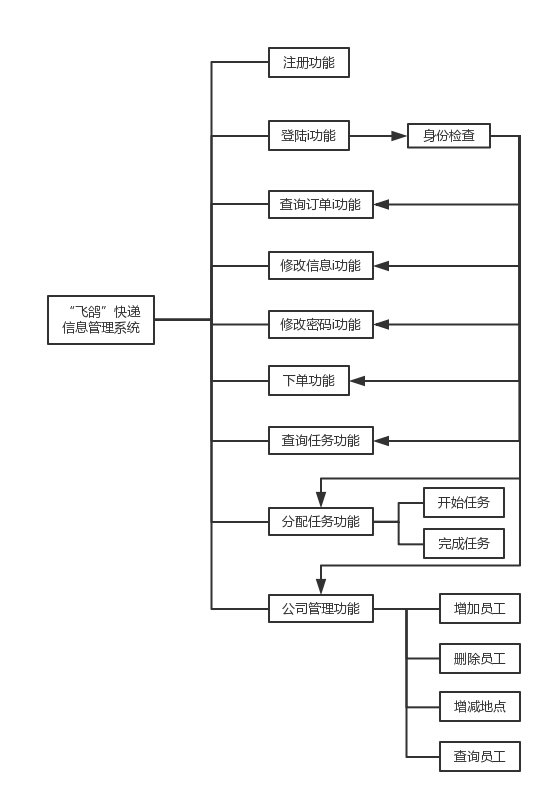
其需求为：管理员自己不会分配到任务；点击查询为分配的任务，系统会将当前未分配的任务显示到界面；点击分配任务，系统会将这些任务分配给员工，可能由多个员工完成（其中有一个负责人）。

**2.9公司管理相关功能**

管理员可以查询系统中所有员工、增加员工、删除员工、增加地点。

其需求为：管理员点击查询员工，系统将所有员工信息显示到界面上；点击增加员工，管理员在界面上填写完员工姓名、员工性别、员工年龄、员工手机、员工地点，提交请求，系统检验通过后写入数据库；点击增加地点，管理员填写完地点信息后，提交请求，系统验证通过写入数据库；管理员选中某个员工，点击删除员工，系统将删除该员工相关的记录。

## 二．系统功能结构设计

****

**三、数据库基本表的定义**

**1.用户信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| User\_id  客户编号 | Int | 10 | 客户编号，不能为空 |
| User\_account  客户账号 | Varchar | 45 | 客户账号（登陆用），不能为空 |
| User\_password  客户密码 | Varchar | 45 | 客户密码，不能为空 |
| User\_name  客户姓名 | Varchar | 45 | 客户姓名（显示用），不能为空 |
| User\_gender  客户性别 | Tinyint | 1 | 客户性别，仅取0（女），1（男） |
| User\_age  客户年龄 | Int | 10 | 客户年龄 |
| User\_phone  客户手机 | Varchar | 20 | 客户手机，不能为空 |
| User\_email  客户邮箱 | Varchar | 45 | 客户邮箱 |

**2.员工信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Employee\_id  员工编号 | Int | 10 | 员工编号，不能为空 |
| Password  员工密码 | Varchar | 45 | 员工密码，不能为空 |
| Name  员工姓名 | Varchar | 45 | 员工姓名，不能为空 |
| Gender  员工性别 | Tinyint | 1 | 员工性别，仅取0（女），1（男） |
| Age  员工年龄 | Int | 10 | 员工年龄 |
| Phone  员工手机 | Varchar | 20 | 员工手机，不能为空 |
| Workingtime  工作时长 | Time |  | 员工工作时间 |
| Location  所属地点 | Int | 10 | 员工所属地点编号 |

**3.订单信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Order\_id  订单编号 | Int | 10 | 订单编号，不能为空 |
| userid  发货人编号 | Int | 10 | 发货人编号，不能为空 |
| Startplace  发货地编号 | Int | 10 | 发货地编号，不能为空 |
| Endplace  收货地编号 | Int | 10 | 收货地编号，不能为空 |
| Order\_state  订单状态 | Int | 2 | 有三种状态0（未配送）、1（配送中）、2（已送达） |
| Order\_time  下单时间 | DateTime |  | 客户与订单的下单联系的属性 |
| Orser\_count  还需运输数 | Int | 10 | 订单送到目标地点还需的运输数 |

**4.货物信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Goods\_id  货物编号 | Int | 10 | 货物编号，不能为空 |
| Weight  重量 | Int | 10 | 货物重量，单位为kg，不能为空 |
| Type  货物类型 | Varchar | 45 | 货物类型，不能为空 |
| Orderid  所属订单编号 | Int | 10 | 货物所属订单编号 |
| Order\_name  货物名称 | Varchar | 45 | 货物名称 |

**5.运输信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Transport\_id  运输编号 | Int | 10 | 运输编号，不能为空 |
| Load\_weight  载重量 | Int | 10 | 运输的最大载重量，单位为kg |
| Startplace  出发地编号 | Int | 10 | 发货地编号，不能为空 |
| Ednplace  目的地编号 | Int | 10 | 收货地编号，不能为空 |
| Vehicle  运输工具 | Varchar | 45 | 运输工具 |
| Starttime  开始时间 | DateTime |  | 运输开始的时间 |
| Endtime  结束时间 | DateTime |  | 运输结束的时间 |
| Person\_id  负责员工编号 | Int | 10 | 运输主负责员工编号 |

**6.地点信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Place\_id  地点编号 | Int | 10 | 地点编号，不能为空 |
| Province  省名 | Varchar | 45 | 所在省 |
| City  市 | Varchar | 45 | 所在市 |
| District  区/镇 | Varchar | 45 | 所在区/镇 |
| Level  级别 | Int | 3 | 地点所处的级别（省级、市级、区级） |

**7.员工-运输联系表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Employee\_transport\_id  联系编号 | Int | 10 | 员工运输联系编号，不能为空 |
| Employee\_id  员工编号 | Int | 10 | 员工编号 |
| Transport\_id  运输编号 | Int | 10 | 运输编号 |
| Charge  是否负责 | Tinyint | 1 | 标识这个员工是否是该运输的负责人 |

**8.运输-货物联系表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 说明及约束 |
| Transport\_goods\_id  联系编号 | Int | 10 | 运输货物联系编号，不能为空 |
| Transport\_id  运输编号 | Int | 10 | 运输编号 |
| Goods\_id  货物编号 | Int | 10 | 货物编号 |

**四、触发器的定义与实现**

**1.employee\_BEFORE\_DELETE**

* 设置触发器，在员工信息表执行删除操作前，扫描员工-运输联系表，将与待删除的员工有关的记录中员工编号更新为系统默认的管理员的编号；同时更新运输信息表，由待删除的员工负责的运输信息记录的负责人编号更新为系统默认的管理员的编号。（目的是为了能够继续执行完该运输）

CREATE TRIGGER `expressdata`.`employee\_BEFORE\_DELETE`

BEFORE DELETE ON `employee`

FOR EACH ROW

BEGIN

UPDATE `transport`

SET `person\_id` = 1

WHERE `person\_id` IN

(SELECT `employee\_id`

FROM `employee\_transport`

WHERE `charge` = 1 AND old.employee\_id = `employee\_transport`.`employee\_id`);

UPDATE `employee\_transport`

SET `employee\_transport`.`employee\_id` = 1

WHERE old.employee\_id = `employee\_transport`.`employee\_id`;

END

**五、存储过程的定义与实现**

**1.ADD\_TRANSPORT()**

* 创建新运输时，同时会将运输任务分配给员工，并生成订单的运输路径。所以需要将该次运输所运送的货物相关的订单信息记录的“所需运输次数”的字段值+1；同时，对该次运输所运送的货物相关的订单，假如其订单状态为“未配送”，则修改为“正在配送”。

CREATE PROCEDURE `ADD\_TRANSPORT`(IN id INT)

BEGIN

UPDATE `order`

SET `order\_count` = `order\_count` + 1

WHERE `order\_id` IN

(SELECT dISTINCT `orderid`

FROM `goods`

WHERE `goods\_id` IN

(SELECT `goods\_id`

FROM `transport\_goods`

WHERE `transport\_id` = id));

UPDATE `order`

SET `order\_state` = 1

WHERE `order\_state` = 0 AND `order\_id` IN

(SELECT dISTINCT `orderid`

FROM `goods`

WHERE `goods\_id` IN

(SELECT `goods\_id`

FROM `transport\_goods`

WHERE `transport\_id` = id));

END

1. **FINISH\_TRANSPORT**

* 当某一个运输过程完成时，需要将该运输所运送货物相关的订单的所需运输次数-1；同时，扫描一遍订单信息表，对“所需运输次数”字段值为0并且订单状态为“正在配送”的记录，将其订单状态更新为“已送到”

CREATE PROCEDURE `FINISH\_TRANSPORT`(IN id INT)

BEGIN

UPDATE `order`

SET `order\_count` = `order\_count` - 1

WHERE `order\_id` IN

(SELECT DISTINCT `orderid`

FROM `goods`

WHERE `goods\_id` IN

(SELECT `goods\_id`

FROM `transport\_goods`

WHERE `transport\_id` = id));

UPDATE `order`

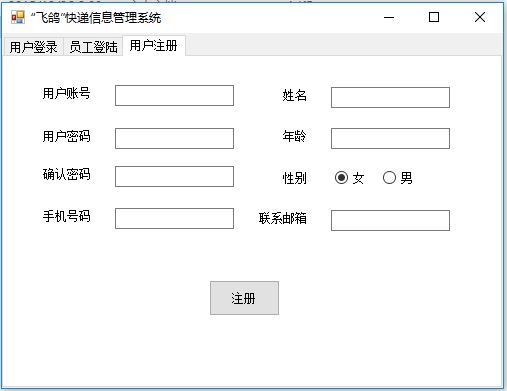
SET `order\_state` = 2

WHERE `order\_count` = 0;

END

**六、系统实现结果**

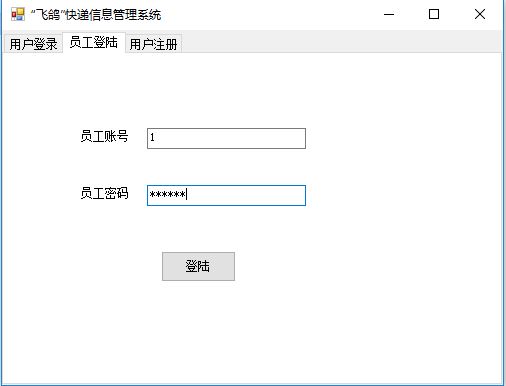
**1.用户注册**

****

1. **用户登录**

****

1. **员工登陆**

****

1. **用户修改信息**

****

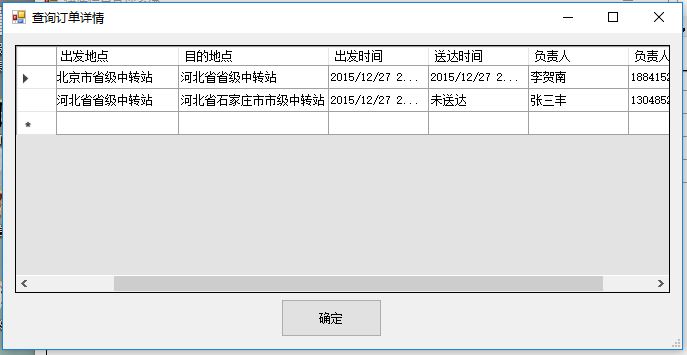
1. **员工修改信息**

****

1. **查询订单**

****

1. **查询订单详情**

****

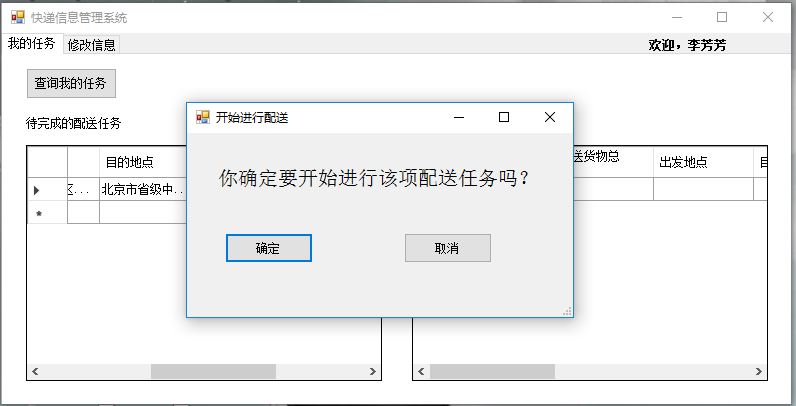
1. **新建订单**

****

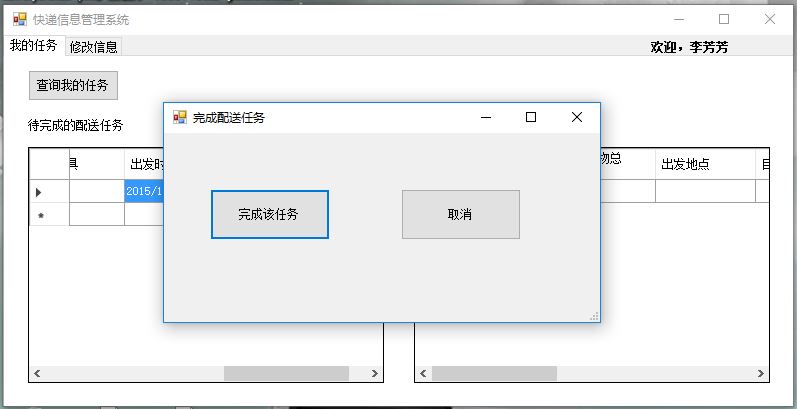
1. **查询任务**

****

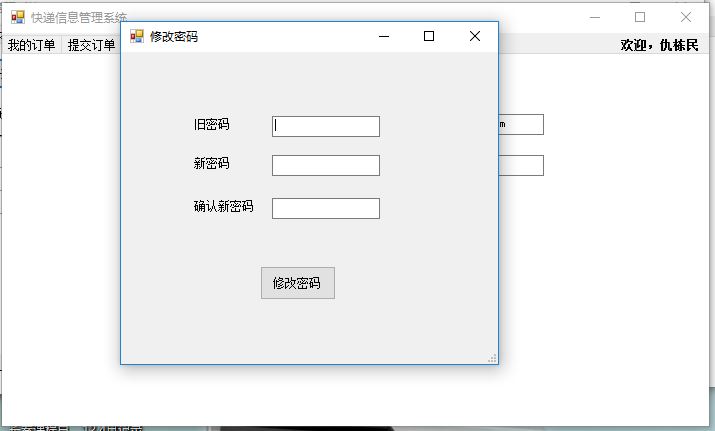
1. **开始执行配送任务**

****

1. **完成配送任务**

****

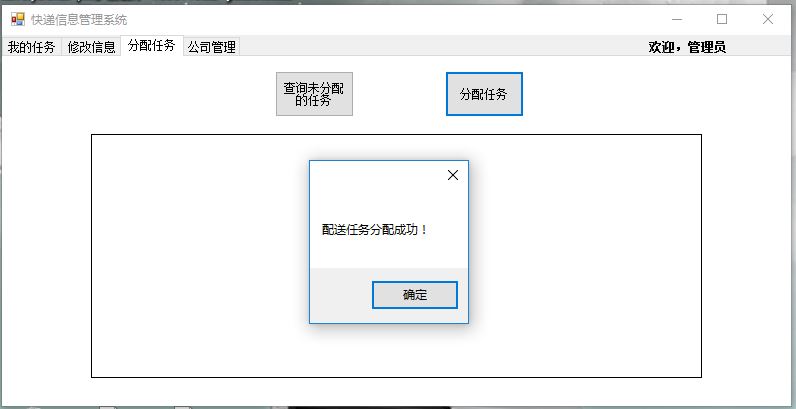
1. **修改密码**

****

1. **查询未分配的订单**

****

1. **分配任务**

****

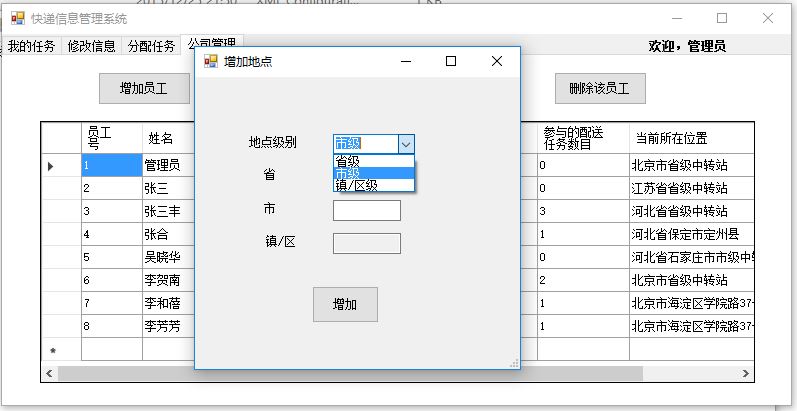
1. **查询员工**

****

1. **增加员工**

****

1. **增加地点**

****

1. **总结**

“飞鸽”快递信息管理系统是一个模拟一个快递公司的信息以及业务流程的信息管理系统，我们的系统还有很多不完善的地方， 只是简单的模拟了一下快递公司的配送流程，还有待进一步改进。但是经过这次的大作业过程我们收获了许多知识和体验。

* 数据库设计经验

在最初的数据库设计的时候，我们感觉自己的想法已经没有什么遗漏了，但是在实现的时候才发现了许许多多的问题，比如订单完成的判定条件、货物的名字等等。不过设计的过程还是让我明白了数据库的设计要至少满足3NF。并学会了如何对数据库进行优化。

* 高级数据库编程

在实现的时候，我发现触发器和存储过程可以很大程度上减少编程复杂度，将一些本来需要频繁和数据库进行交互的操作化简为一个操作，或者变为触发器让DBMS自己进行相关操作，这样就减少了许多与DBMS交互的过程，极大的优化了系统性能和编程复杂度。

* 团队协作经验

从最开始的选题，到最后的提交，我们进行了无数次讨论，在每一次讨论后，我们的思路都变得更清晰，意见更加一致。我们体会到团队合作的不简单，但也同时感受到了团队协作带来的极高开发效率

* 项目开发经验

虽然我们的大作业只是一个很小的应用，但是我们通过使用git版本管理工具，获得了多人同时快速开发的代码整合经验。提高了我们多人开发的经验。