

# 中北大学软件学院

## 实训说明书

实训名称： 面向对象与数据库课程设计实训

题目名称： 物流公司车队管理

专 业： 软件工程

班 级： 19130404

### 小组成员

学号： 1913040430 姓名： 侯博文 成绩：     

学号： 1913040411 姓名： 杨祎程 成绩：     

学号： 1913040416 姓名： 田忆南 成绩：     

学号： 1913040423 姓名： 方鹏 成绩：     

学号： 1913040450 姓名： 武凯琪 成绩：     

学号： 1913040444 姓名： 赵文龙 成绩：     

指导教师： 李玉蓉

2021 年 1 月

# 中北大学

## 实训任务书

实训名称:	面向对象与数据库课程设计		
学院、系:	软件学院		
专    业:	软件工程		
班        级:	19130404		
学 生 姓 名:	侯博文	学号:	1913040430
	杨祎程	学号:	1913040411
	田忆南	学号:	1913040416
	方鹏	学号:	1913040423
	武凯琪	学号:	1913040450
	赵文龙	学号:	1913040444
设计题目:	物流公司车队管理		
起迄日期:	2020年12月28日~2021年1月15日		
指导教师:	李玉蓉		

发任务书日期: 2020年12月28日

# 实训任务书

## 1. 实训的目的和要求：

通过实训，使学生进一步理解和巩固面向对象程序设计和数据库的基本概念、思想和核心技术。培养学生能够根据数据库应用系统的实际需求，综合运用关系型数据库的基本原理，确保可以合理组织数据及有效存储数据的条件下，进行数据库设计和实现并利用面向对象分析、设计和编程的思想，开发出可复用、满足需求的数据库管理系统的能力；同时要求学生能够分阶段撰写规范的文档，清晰表达应用系统的需求、概要设计、详细设计、编码实现和测试等课程设计成果。

### 基本要求：

- (1) 使用 Java 完成开发，数据使用数据库持久存储，界面必须是图形用户界面；
- (2) 按照软件工程的思想完成系统开发过程（包括需求分析、系统设计、编码实现和测试）；
- (3) 最终的软件系统要求数据充实、界面友好、使用方便。

## 2. 实训的具体工作内容及分工：

### 设计内容：

设计一个物流公司车辆管理系统。

### 主要功能及要求：

软件要求实现司机档案管理、车辆档案管理、车辆日常费用管理、油卡管理以及出（派）车管理，完成车辆出车情况查询，车辆费用统计及各种管理报表打印功能。

### 主要功能及要求：

- 1、司机档案管理
- 2、车辆档案管理
- 3、车辆日常费用管理（维修，保险，罚款等）
- 4、油卡管理
- 5、出（派）车管理
- 6、费用统计

# 实训任务书

## 3. 对实训成果的要求:

提交实训说明书和光盘（每班统一制作 1 张光盘，每组一个目录）。

每组的光盘目录内含：

- （1）源程序文件；
- （2）数据备份文件；
- （3）实训说明书文档（另需交打印文档一份）。

## 4. 实训工作进度计划:

起 迄 日 期	工 作 内 容
2020 年 12 月 28 日~2021 年 1 月 4 日	选择实训题目，明确实训任务，完成并提交需求分析设计报告；
2021 年	功能模块代码实现、调试并提交中期报告；
1 月 5 日 ~ 1 月 11 日	完善功能代码、完善系统数据、系统测试、撰写并提交实训说明书文档；
1 月 9 日 ~ 1 月 13 日	验收程序、答辩；
1 月 14 日	修改、上交说明书。
1 月 15 日	

## 任务分工情况说明

	姓名	分工
组长	侯博文	分析需求，撰写需求分析，登录功能、系统主界面和出（派）车管理面板的设计和代码实现、退出系统功能，司机和管理员个人信息修改界面，司机界面，数据库工具类，撰写说明书。
组员	杨祎程	中期报告的撰写，司机档案处理面板的设计和代码实现、定义司机档案类、添加司机档案的增删改查操作，超级管理员界面。
组员	田忆南	建立数据库，车辆档案处理面板的设计和代码实现、定义车辆档案类、添加车辆档案的增删改查操作。
组员	方鹏	车辆日常费用管理面板设计和代码实现、定义车辆日常费用类、添加车辆日常费用的增删改查操作。
组员	武凯琪	油卡管理面板设计和代码实现、定义油卡类、添加油卡的增删改查操作，系统用到的路径、屏幕工具类。
组员	赵文龙	费用统计面板的设计和代码实现、编写系统用到的时间工具类。

## 目 录

1 引言.....	1
2 需求分析.....	1
2.1 分析目的.....	1
2.1.1 编写目的.....	1
2.1.2 预读作者.....	1
2.1.3 项目背景.....	1
2.2 任务概述.....	2
2.2.1 任务目标.....	2
2.2.2 运行环境.....	2
2.2.3 条件与限制硬件配置要求.....	2
2.3 系统说明.....	2
2.3.1 系统描述.....	2
2.3.2 系统基本业务流程图.....	2
2.3.3 系统 E-R 图.....	3
2.3.4 数据字典.....	5
2.4 功能需求.....	6
2.4.1 功能介绍.....	6
2.4.2 信息描述.....	7
2.4.3 数据安全性与完整性需求.....	7
2.5 性能需求.....	8
2.6 与系统交互需求.....	8
2.7 其他需求.....	8
3 概要设计.....	9
3.1 模块结构图.....	9
3.2 技术选型.....	9
4 详细设计.....	10
4.1 登录模块的设计与实现.....	10

4.1.1 登录界面.....	10
4.1.2 持久层.....	13
4.1.3 实体类 User.....	13
4.1.4 操作层 Dao 层.....	13
4.1.5 业务逻辑层.....	14
4.1.6 界面层（窗体层）.....	15
4.2 司机档案处理模块的设计与实现.....	17
4.2.1 显示界面和功能界面.....	17
4.2.2 持久层.....	24
4.2.3 实体类 Driver.....	25
4.2.4 操作层 Dao 层.....	26
4.2.5 界面层（窗体层）.....	26
4.3 车辆档案处理模块的设计与实现.....	28
4.3.1 显示界面和功能界面.....	28
4.3.2 持久层.....	36
4.3.3 实体类 Car.....	37
4.3.4 操作层 Dao 层.....	37
4.3.5 界面层（窗体层）.....	38
4.4 车辆日常费用管理模块的设计与实现.....	40
4.4.1 显示界面和功能界面.....	40
4.4.2 持久层.....	47
4.4.3 实体类 Carcost.....	47
4.4.4 操作层 Dao 层.....	48
4.4.5 界面层（窗体层）.....	48
4.5 油卡管理模块的设计与实现.....	50
4.5.1 显示界面和功能界面.....	50
4.5.2 持久层.....	56
4.5.3 实体类 Oilcard.....	57
4.5.4 操作层 Dao 层.....	57

4.5.5 界面层（窗体层）.....	58
4.6 司机任务&&出派车辆管理的设计与实现.....	60
4.6.1 显示界面和功能界面.....	60
4.6.2 持久层.....	66
4.6.3 实体类 DispatchVan.....	67
4.6.4 操作层 Dao 层.....	68
4.6.5 界面层（窗体层）.....	68
4.7 管理员个人信息模块的设计与实现.....	70
4.7.1 显示界面和功能界面.....	70
4.7.2 持久层.....	73
4.7.3 实体类 User.....	74
4.7.4 操作层 Dao 层.....	74
4.7.5 界面层(窗体层).....	75
4.8 费用统计模块的设计与实现.....	76
4.8.1 显示界面和功能界面.....	76
4.8.2 操作层 Dao 层.....	79
4.8.3 界面层(窗体层).....	79
4.9 超级管理员模块的设计与实现.....	80
4.9.1 显示界面和功能界面.....	80
4.9.2 持久层.....	86
4.9.3 实体类 User.....	87
4.9.4 操作层 Dao 层.....	87
4.9.5 界面层(窗体层).....	88
4.10 核心工具类的说明.....	90
4.10.1 CalendarPannel 类.....	90
4.10.2 ExtTable 类.....	93
5 结论.....	95
心得体会.....	96
参考文献.....	97



## 1、绪论

随着移动互联网的兴起,为了满足人们对海量数据信息化的需求,软件工程蓬勃发展。于是“软件”的定义慢慢从“一种产品”转变为“一种服务”,用户的需求是不断变化 的,很难找到一个通用的模式去解决问题。当下国家号召高校信息化建设,积极参与其中更是每个大学生义不容辞的义务。

物流车辆管理系统,可以实现对物流车辆进行管理。随着计算机技术不断创新发展以及与国际市场对接,传统的物流业务给公司创造的利润越来越有限,因而物流管理系统需要向基于网络的车辆管理进行转型,依靠互联网技术和信息管理系统,实现高效的管理和优化的工作流程,为企业创造新的利润增长点和提高企业核心竞争力。

本系统使用了《面向对象程序设计》和《数据库概论》的知识,在开发过程中提高了动手能力,巩固了所学知识,提高了自身综合素质。

## 2、需求分析

### 2.1 分析目的

#### 2.1.1 编写目的

本需求说明书全面描述物流公司车辆管理系统的各种功能,针对物流公司车队管理过程中派车灵活性强、运输司机多、车辆维护、出派费用繁杂、工作量大等特点,设计开发物流公司车队管理系统,旨在现代的信息网络化管理技术解决物流公司车队管理中存在的问题,减少车队管理的难度,降低管理员的工作强度和难度。

#### 2.1.2 预读作者

预期读者: 物流公司车辆管理系统管理员, 项目开发人员等。

#### 2.1.3 项目背景

物流公司车队管理工作是物流管理工作的重要环节,具有工作量大、灵活性强等特点。

物流公司车队管理系统是一种以互联网为基础的管理模式,本系统最根本的目的是让管理人员在线进行车辆、司机、车辆使用、车辆费用、油卡信息的输入、查询、修改和删除,费用统计信息的查询,管理员可以在线管理信息,为车队的管理提供了便捷的 平台。具有操作性好、交互友好、响应快捷等优点。建立了高效的服务平台。

## 2.2 任务概述

### 2.2.1 任务目标

解决物流公司车队管理中存在的问题，减少车队管理的难度，降低管理员的工作强度和难度，使之跟的上时代的发展。

### 2.2.2 运行环境

数据库 MySql 8.0

Eclipse

Window 10

### 2.2.3 条件与限制硬件配置要求

硬件外部设备需要 GTX 1650 以上的 PC 机，内存需要 16M 以上。

软件要求操作人员具有初步的相关知识。

网络配置要求使用可靠性较高的相关网络硬件设施。

## 2.3 系统说明

### 2.3.1 系统描述

本系统主要为了降低物流公司车辆管理人员的管理难度和工作强度，该系统可以提供更快、更优质的服务，一定程度上提高工作效率。

### 2.3.2 系统基本业务流程图

系统基本业务流程图 2.1 如下。

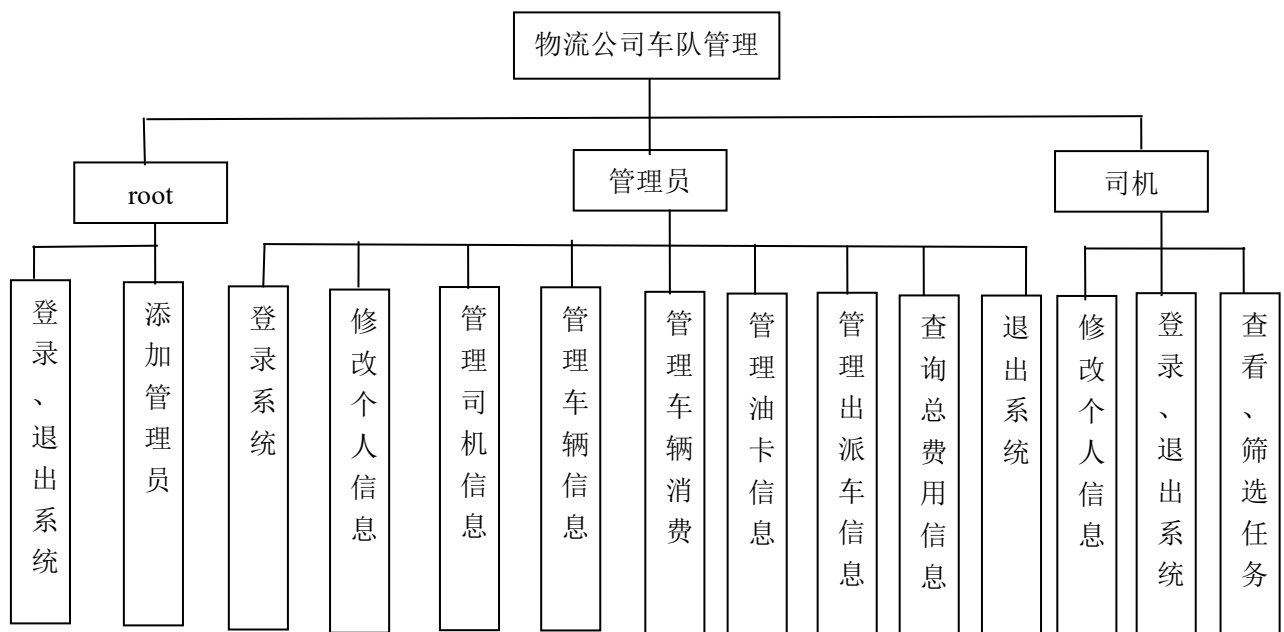


图 2.1 系统基本业务流程图

### 2.3.3 系统 E-R 图

1、系统 E-R 图如下图 2.2。

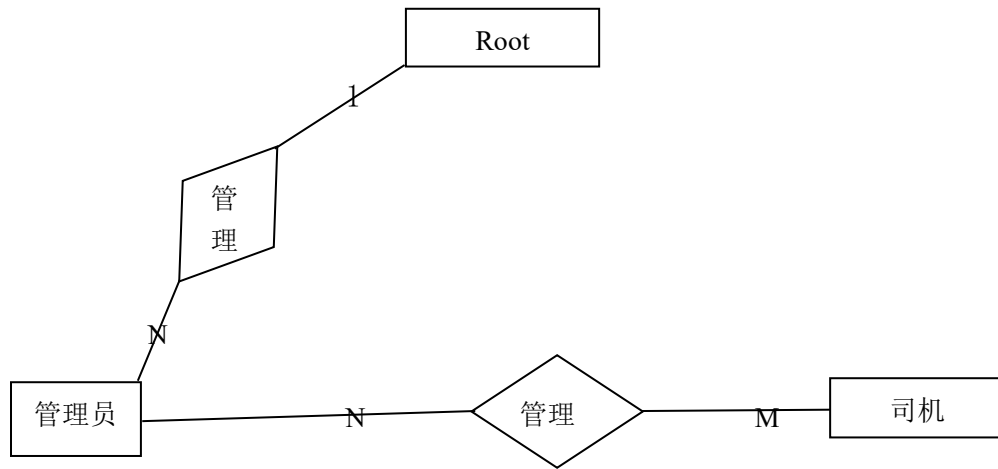


图 2.2 系统 E-R 图

2、RootE-R 图如下图 2.3。

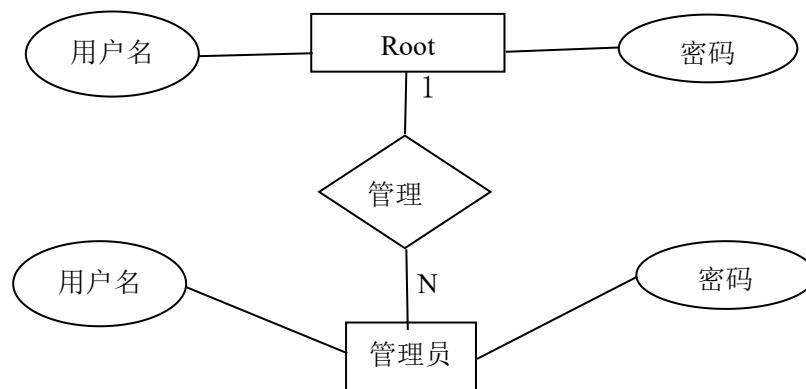


图 2.3RootE-R 图

3、司机 E-R 图如下图 2.4。

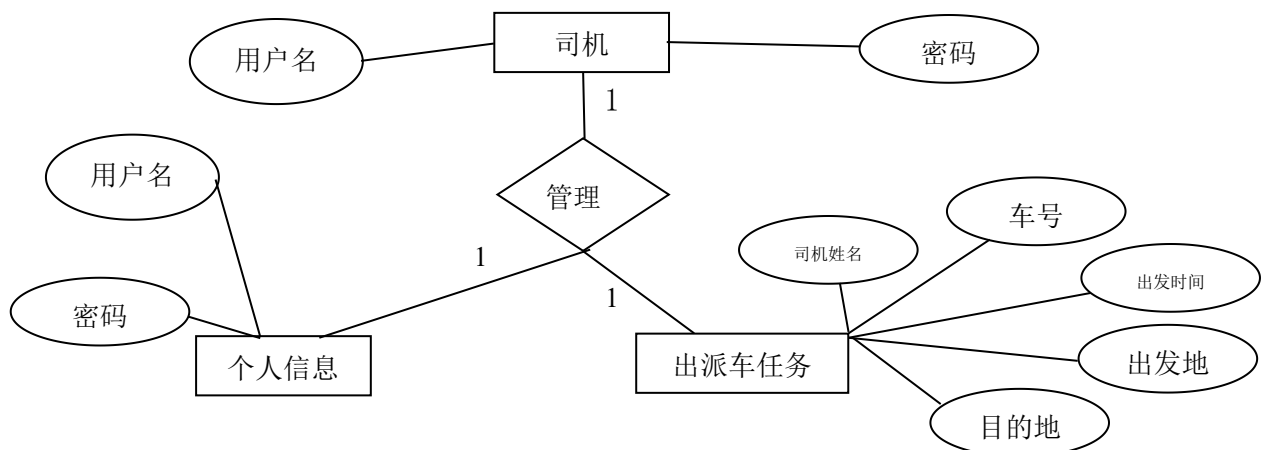


图 2.4 司机 E-R 图

4、管理员 E-R 图如下图 2.5。

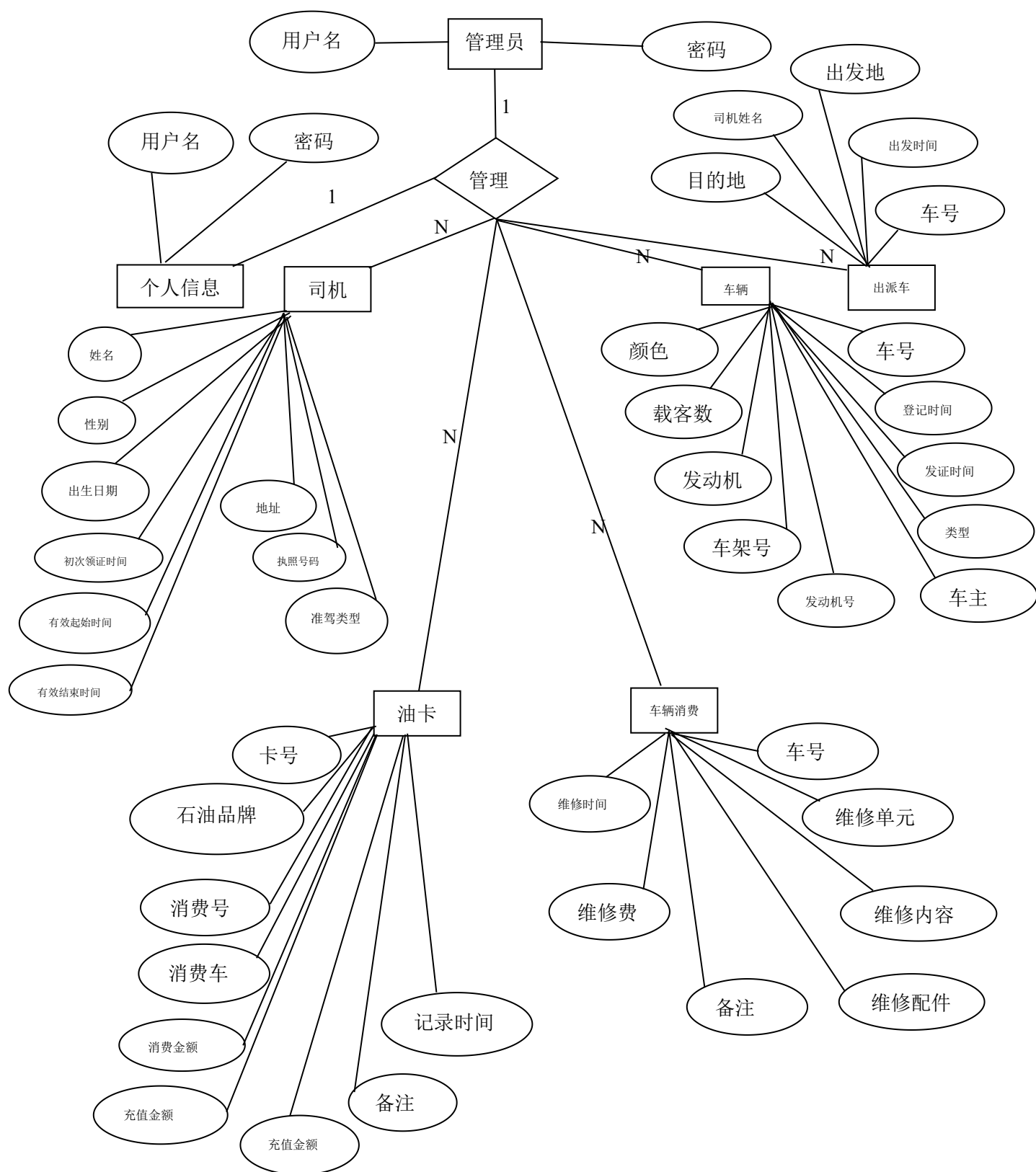


图 2.5 管理员 E-R 图

## 2.3.4 数据字典

表 3.1 司机信息数据字典

字段名	数据类型	默认值	允许为空	唯一	主/外键	含义
Name	char	null	no	唯一	主键, 外键	司机姓名
Sex	char	男	no			司机性别
Day	char	null	no			司机出生日期
FirstDay	char	null	no			初次领证时间
BeginDay	char	null	no			有效起始时间
Overday	char	null	no			有效结束时间
place	char	null	no			地址
PermitNum	char	null	no			执照号码
CarType	char	A3	no			准驾车型

表 3.2 车辆信息数据字典

字段名	数据类型	默认值	允许为空	唯一	主/外键	含义
CarNum	char	null	no	唯一	主键, 外键	车牌号
RegisterTime	char	null	no			登记时间
CertificateTime	char	null	no			发证时间
Type	char	null	no			类型
CarOwner	char	null	no			车主
EngineNum	char	null	no			发动机号
Num	char	null	no			车架号
CarBrand	char	null	no			汽车品牌
Carpeople	char	null	no			核定载客数
CarColor	char	null	no			颜色

表 3.3 车辆费用数据字典

字段名	数据类型	默认值	允许为空	唯一	主/外键	含义
CostNum	char	null	no	唯一	主键	车号
ServiceTime	char	null	no			维修时间
ServiceMany	char	null	no			维修费用（元）
ServiceUtil	char	null	no			维修单位
ServiceContent	char	null	no			维修内容
ServiceAccessories	char	null	no			维修配件
Remark	char	null	no			备注

## 中北大学面向对象与数据库课程设计实训说明书

表 3.4 为油卡信息数据字典

字段名	数据类型	默认值	允许为空	唯一	主/外键	含义
CardNum	char	null	no	唯一	主键	油卡编号
OilBrand	char	null	no			石油品牌
PayNo	char	null	no			消费编号
PayCarNO	char	null	no			消费车号
PayMoney	char	null	no			消费金额（元）
InNo	char	null	no			充值编号
InMoney	char	null	no			充值金额
InputTime	char	null	no			记录时间
OilRemark	char	null	no			备注

表 3.5 为出派车信息数据字典

字段名	数据类型	默认值	允许为空	唯一	主/外键	含义
DeliveryNum	char	null	no	唯一	主键	车号
DeliveryName	cahr	null	no			司机姓名
GoOff	char	null	no			出发时间
Start	char	null	no			出发地
Bourn	char	null	no			目的地

## 2.4 功能需求

### 2.4.1 功能介绍

#### 1、功能介绍

##### （1）登录/退出功能

在系统登录认证的时候，需要判别用户的身份，即判断身份密码是否符合权限。

##### （2）管理系统

- 1>录入、查询、修改、删除、打印司机信息。
- 2>录入、查询、修改、删除、打印车辆信息。
- 3>录入、查询、修改、删除、打印车辆日常消费信息。
- 4>录入、查询、修改、删除、打印油卡信息。
- 5>录入、查询、修改、删除、打印出派车信息。
- 6>查询、打印费用信息。
- 7>修改个人信息。

##### （3）Root 系统

1>录入、查询、修改、删除、打印管理员信息。

(4) 司机系统

1>查询任务信息。

2>修改个人信息。

## 2.4.2 信息描述

该系统主要记录车辆、司机、出派情况、车辆费用、油卡、管理员的基本信息。

1) 车辆基本信息

车辆基本信息包括车号、登记时间、发证时间、类型、车主、发动机号、车驾号、汽车品牌、核定载客数、颜色。

2) 司机基本信息

司机基本信息包括姓名、性别、出生日期、初次领证日期、有效起始时间、有效结束时间、地址、执照号码、准驾车型。

3) 车辆费用基本信息

车辆费用基本信息包括车号、维修时间、维修费用、维修单元、维修内容、维修配件、备注。

4) 油卡基本信息

油卡基本信息包括油卡编号、石油品牌、录入时间、充值金额、充值编号、消费金额、消费车号、消费编号、备注。

5) 出派车基本信息

出派车基本信息包括车号、司机姓名、出发时间、出发地、目的地。

6) 管理员基本信息

管理员基本信息包括用户名、密码。

7) 司机账户信息

司机账户信息包括用户名、密码。

## 2.4.3 数据安全性与完整性需求

1) 安全性需求

(1) 登录系统时，需要输入正确的用户名和密码，方可登录系统。

(2) 管理员由上级直接任命，所以本系统管理员用户需要上级管理直接添加，不存在注册通道。

## 2) 完整性需求

### (1) 实体完整性

车号、姓名、用户名、用户名为车辆、司机、管理员、司机账户的主码。

### (2) 参照完整性

出派车的车号与车辆信息中的车号相对应，车辆费用中的车牌号和车辆信息中的车号相对应，出派车的司机姓名与司机信息中的姓名相对应。

### (3) 用户定义的完整性

司机信息中的性别为男、女。

司机准驾车型信息中只有 A3、B1、B2、C1、C2。

油卡收支类型只有充值和消费两种。

## 2.5 性能需求

在进行向数据库文件提取数据时，要求数据记录定位准确，程序响应时间在人的感觉和视觉事件范围内；信息交换时间：要求在程序调用前调用后都与数据库保持同步更新，网络信息交换施加应该小于程序调用的时间，要求数据库具有很好的更新能力，由于本产品是实验性软件，故对磁盘和内存容量没有很高的要求。

## 2.6 与系统交互需求

网络硬件接口要求：现实中要求具有高速以太网组网以实现联网操作，但是在理论实验验证软件本身的目的来看，无需网络通讯接口。硬件接口无特别要求。Windows 标准接口，要求与其他软件无冲突错误发生。屏幕固定大小，通过改变面板实现各个功能的操作。

## 2.7 其他需求

### (1) 可用性

本软件也可以通过单步跟踪的操作进行检查处理。

### (2) 安全性

由于软件运行数据放在数据库中，所以参数不容易被错改、破坏，万一参数受到破坏也不会影响源程序。

### (3) 可维护性

本软件利用数据库进行编程，系统结构由程序基本确定，大量的参数及文本内容



全部放于数据库中。修改、更新数据只要在数据库进行修改添加，而不需要对系统结构进行修改，这样系统维护性、升级都十分方便。

### 3、概要设计

#### 3.1 模块结构图

本系统的模块结构图如图 3.1 所示。

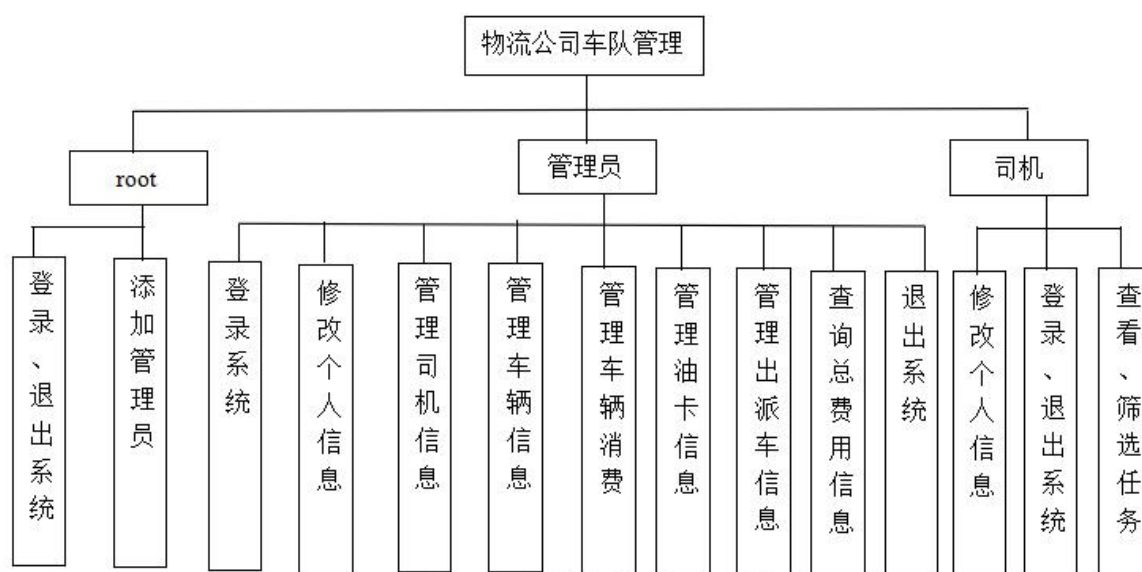


图 3.1 系统基本业务流程图

#### 3.2 技术选型

数据持久化到文件存储，存储格式选择了使用数据库存储，这是一种可以支撑高负荷查询压力，但是不能作为数据仓库使用的准企业级数据库系统。MySQL 是一种开放源代码的关系型数据库管理系统（RDBMS），使用最常用的数据库管理语言——结构化查询语言（SQL）进行数据库管理。MySQL 因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL 是管理内容最好的选择。

系统是图形化的，运用了 Swing，这是一个为 Java 设计的 GUI 工具包。

Swing 是 JAVA 基础类的一部分。Swing 包括了图形用户界面（GUI）器件如：文本框，按钮，分隔窗格和表。Swing 提供许多比 AWT 更好的屏幕显示元素。它们用纯 Java 写成，所以同 Java 本身一样可以跨平台运行，这一点不像 AWT。它们是 JFC 的一部分。它们支持可更换的面板和主题（各种操作系统默认的特有主题），然而不是真的使用原生平台提供的设备，而是仅仅在表面上模仿它们。这意味着你可以在任意平台上使用 JAVA 支持

的任意面板。轻量级组件的缺点则是执行速度较慢，优点就是可以在所有平台上采用统一的行为。在这个系统上用到了 JPanel、JTabbedPane、JMenuItem、JOptionPane、JTable、JLabel、JTextField、JButton、JRadioButton、ButtonGroup、还有自定义的时间选择器 CalendarPanel。

## 4、详细设计

### 4.1 登录模块的设计与实现

#### 4.1.1 登录界面

##### (1) 登录

用户在登录时，需输入账号、密码，然后点击“登录”按钮，如图 4.1.1，但由于身份的不同，登录后显示的界面也不同，超级管理员登录可以实现对普通管理员的信息的调整，如图 4.1.2，普通管理员登录后则显示该项目核心功能（也是着重进行详细说明），还可以完成对司机账号密码的注册，如图 4.1.3，司机登录后则可以查看自己任务，如图 4.1.4。

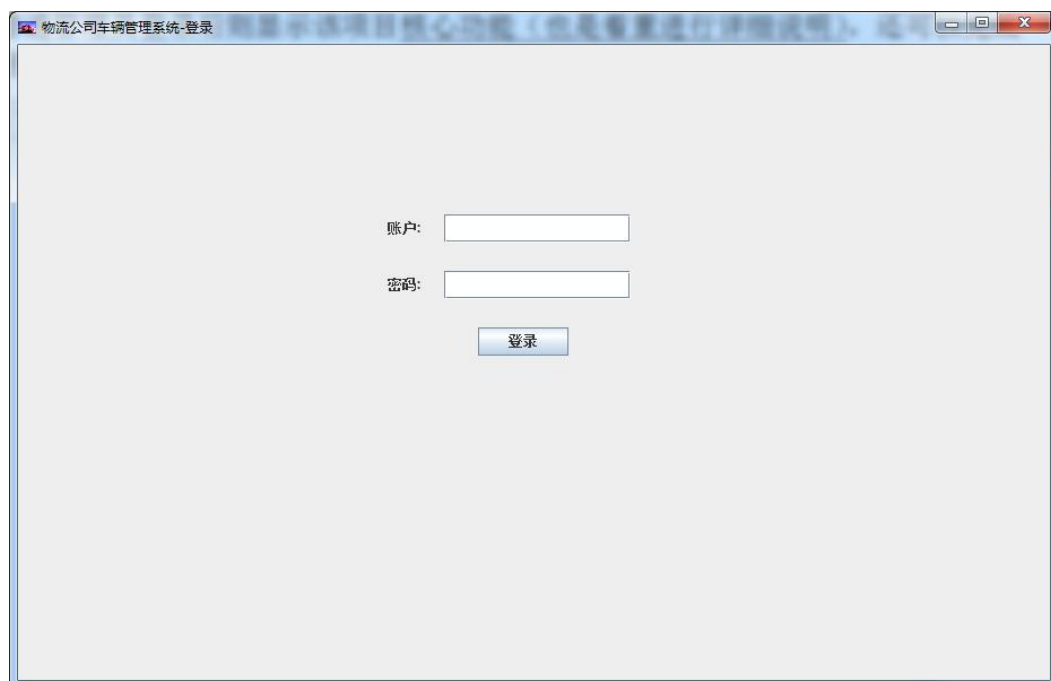


图 4.1.1 登录界面

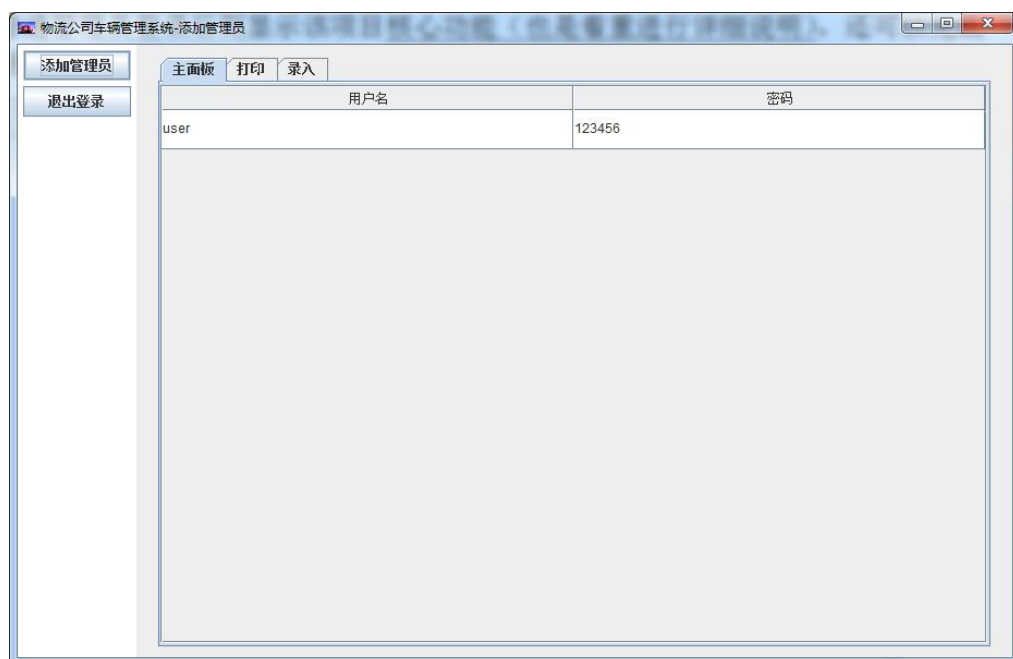


图 4.1.2 超级管理员登录的界面

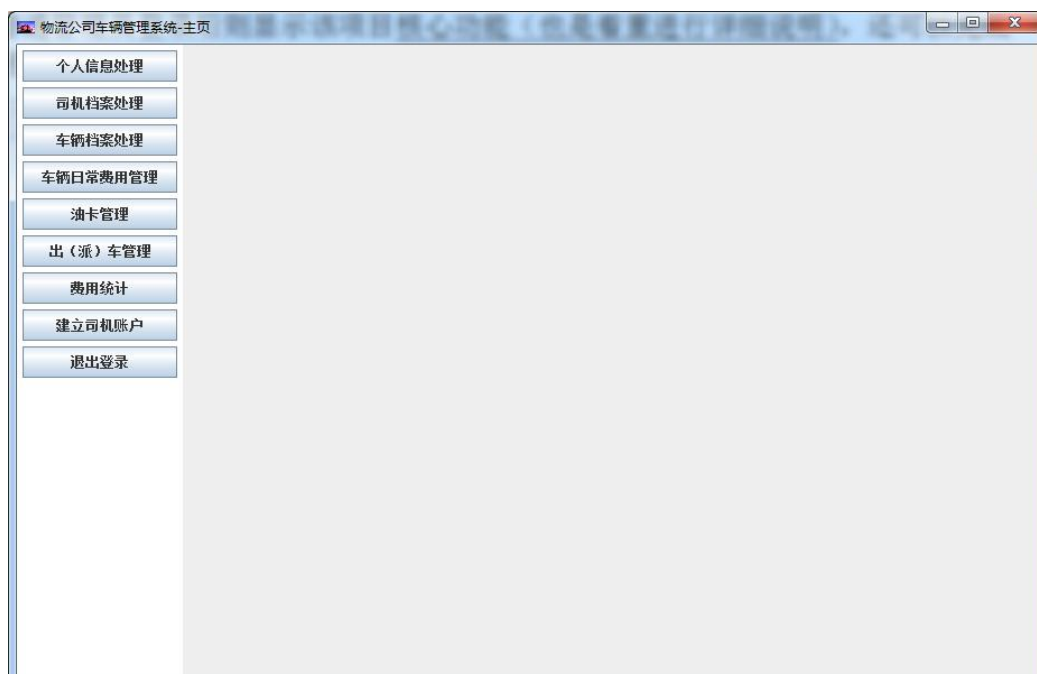


图 4.1.3 普通管理员登录的界面

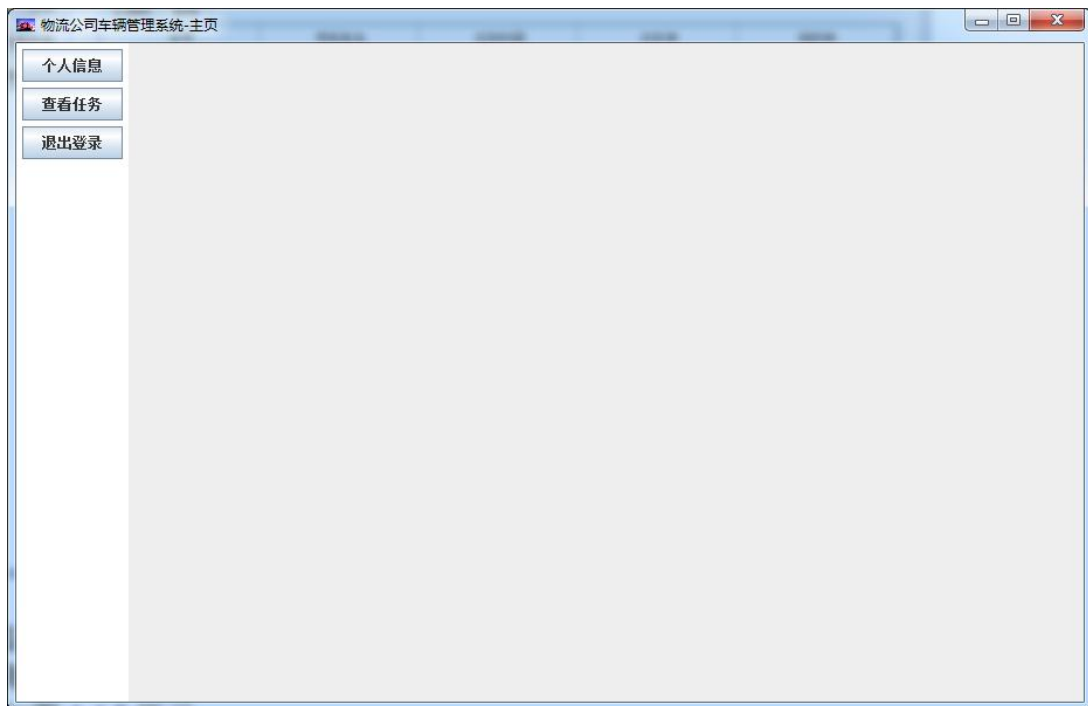


图 4.1.4 普通司机登录的界面

(2) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.1.5 所示。

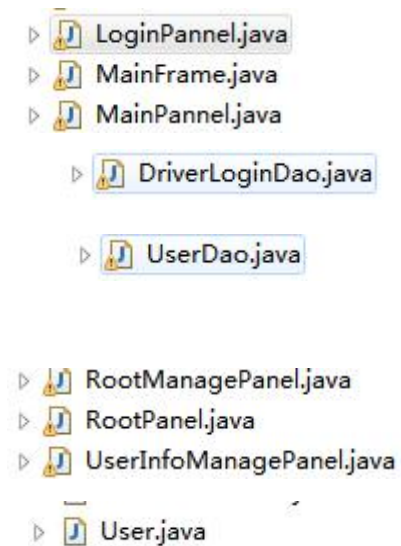


图 4.1.5 登录模块组织

### 4.1.2 持久层

使用数据库 mysql 中表 carcost 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

### 4.1.3 实体类 User

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.4.9 所示。

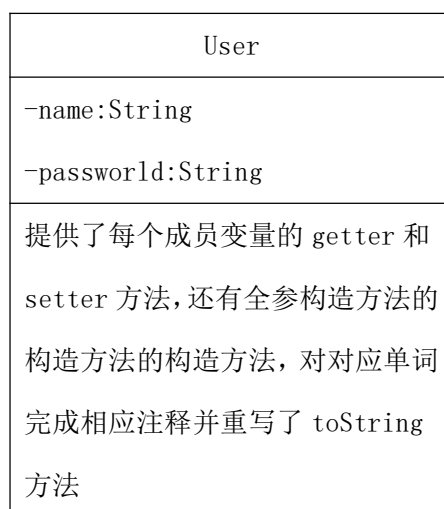


图 4.1.6 User 类图

### 4.1.4 操作层 Dao 层

该层（UserDao）主要完成对数据库中 User 的相关操作。

#### （1）获取密码的方法

```
public static String queryUserByName(String user) ;
```

该方法通过封装 user 实现登录过程账号密码匹配。

(2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中 User 全部信息的获取，返回 String 类型的集合，用于超级管理员对其操作。

(3) 获取某行用户用全部信息的方法

```
public static List<User> getAll();
```

返回 User 类型的集合。

(4) 更新信息（编辑）的方法

```
public static void remove(User user, User info);
```

定位某行信息对其进行更新。

(5) 删除某行信息的方法

```
public static void delete(User de);
```

定位某行对该行信息进行筛选。

(6) 录入（增加）的方法

```
public static void save(User user);
```

该方法通过封装 User 实现用户增加。

#### 4.1.5 业务逻辑层

从业务角度，在进行登录操作时，为了实现对不同用户的划分，需要合理合法，判断该登陆者的身份。否则，不许登录。

判断核心语句如下：

```
try {  
  
    if (passwordText.getText().equals(UserDao.queryUserByName(userText.getText()  
t())) {  
  
        MainPannel.placeComponents(panel);  
    }  
else if  
(passwordText.getText().equals("123456")&&userText.getText().equals("root"))
```

```
{
    RootPanel.placeComponents(panel);
} else if
(passwordText.getText().equals(DriverLoginDao.queryUserByName(userText.getText())) {
    users[0] = userText.getText();
    DriverLoginPanel.placeComponents(panel);
} else
{
    JOptionPane.showMessageDialog(panel, "账户名或者密码错误!!!", "登录异常", JOptionPane.WARNING_MESSAGE);
}
} catch (Exception e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
}
```

#### 4.1.6 界面层（窗体层）

本层包括三个窗体类：LoginPannel, MainPannel (普通管理员显示), 运行结果如图 4.1.1, 4.1.3。

\*登录窗体中登录响应流程和核心代码

登录过程中，若输入账号或者密码错误，则弹出对话框显示，如图 4.1.7 所示。



图 4.1.7 登录信息错误显示

若登录成功，则按照用户身份打开对应的窗体。

登录按钮事件处理核心代码如下：

```
loginButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        try {  
            if  
(passwordText.getText().equals(UserDao.queryUserByName(userText.getText())))  
{  
                MainPannel.placeComponents(panel);  
            } else if(passwordText.getText().equals("123456") &&  
userText.getText().equals("root")) {  
                RootPanel.placeComponents(panel);  
            } else  
if(passwordText.getText().equals(DriverLoginDao.queryUserByName(userText.get  
Text())))) {  
                users[0] = userText.getText();  
                DriverLoginPanel.placeComponents(panel);  
            } else {  
                JOptionPane.showMessageDialog(panel, "账户名或者密  
码错误!!!", "登录异常", JOptionPane.WARNING_MESSAGE);  
            }  
        } catch (Exception e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
    }  
});
```



## 4.2 司机档案处理模块的设计与实现

### 4.2.1 显示界面和功能界面

#### (1) 信息录入

管理员在进行司机档案录入时需输入姓名、性别、出生日期、初次领证时间、有效起止时间、有效结束时间、地址、执照号码、准假车型，录入界面如图 4.2.1 所示。

主面板打印录入

司机档案录入

姓名：

性别：☒男  
☐女

出生日期：

初次领证时间：

有效起始时间：

有效结束时间：

地址：

执照号码：

准驾车型：

确定

图 4.2.1 录入界面

#### (2) 司机档案主面板

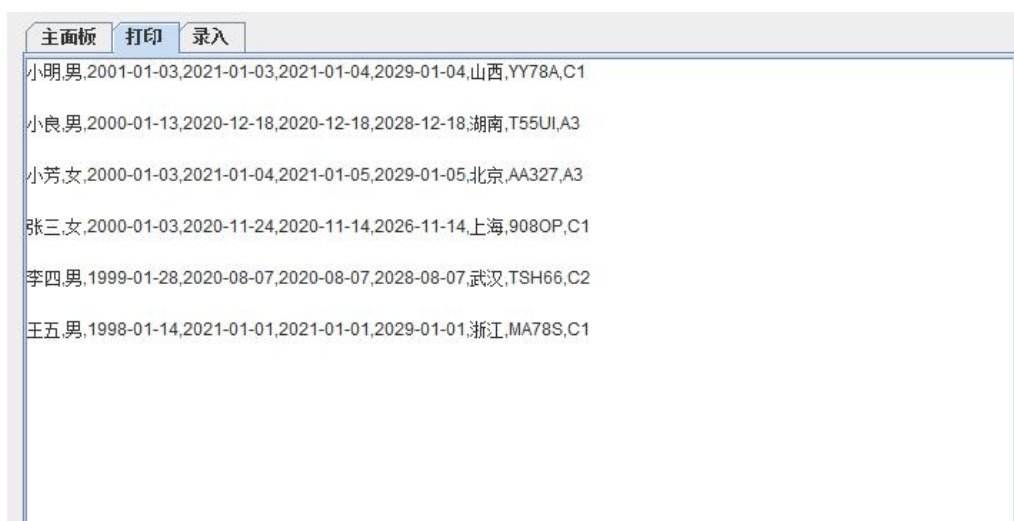
信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.2.2 所示。

姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小明	男	2001-01-03	2021-01-03	2021-01-04	2029-01-04	山西	YY78A	C1
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖南	T55UI	A3
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

图 4.2.2 主面板界面

### (3) 打印界面

打印界面显示出司机相关信息的语句，如图 4.2.3 所示。



4.2.3 打印界面

### (4) 删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.2.4 所示。

主面板 打印 录入									
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型	
小明	男	2001-01-03	2021-01-03	2021-01-04	2029-01-04	山西	YY78A	C1	
小良	<div>删除选定行</div> <div>刷新</div> <div>编辑</div> <div>筛选</div>	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖南	T55UI	A3	
小芳		2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3	
张三		2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1	
李四		1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2	
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1	

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖南	T55UI	A3
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

图 4.2.4 删除本行信息及自动刷新后主面板展示

## (5) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改司机信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.2.5 所示。

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖南	T55UI	A3
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

主面板打印录入编辑

司机档案录入

姓名: 小良

性别: ☒ 男  
☐ 女

出生日期: 2000-01-13

初次领证时间: 2020-12-18

有效起始时间: 2020-12-18

有效结束时间: 2028-12-18

地址: 湖南

执照号码: T55UI

准驾车型: C1

确定

主面板打印录入编辑

司机档案录入

姓名: 小良

性别: ☒ 男  
☐ 女

出生日期: 2000-01-13

初次领证时间: 2020-12-18

有效起始时间: 2020-12-18

有效结束时间: 2028-12-18

地址: 湖北

执照号码: T55UI

准驾车型: B2

确定

## 中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖北	T55UI	B2
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

图 4.2.5 编辑功能的过程及主面板展示

### (6) 筛选功能

点击筛选功能后弹出对话框，让管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4.2.6 所示。

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖北	T55UI	B2
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

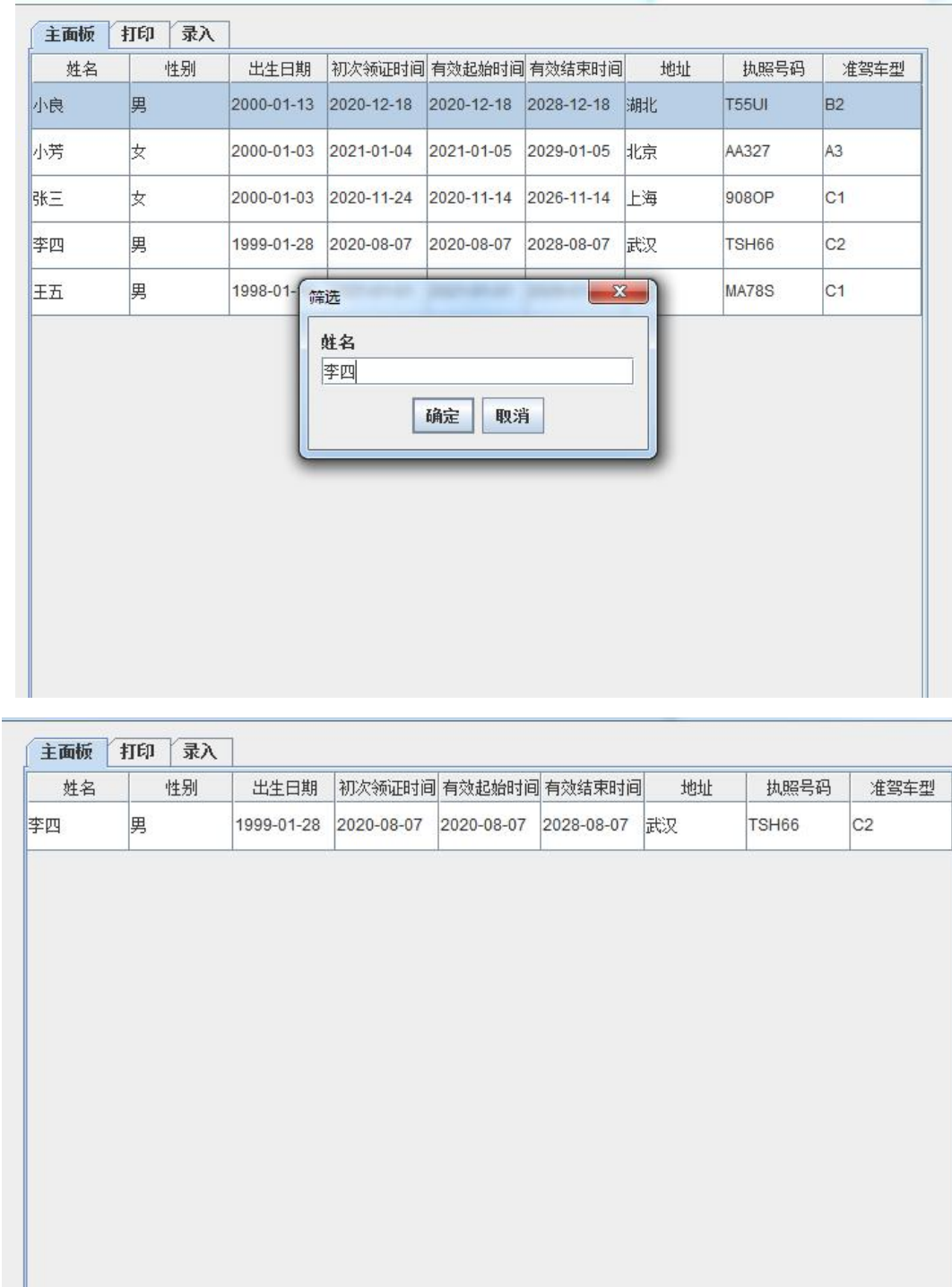


图 4.2.6 筛选功能及主面板展示

(7) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.2.7 所示。

中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
刷新 筛选 编辑 删除选定行								

主面板 打印 录入								
姓名	性别	出生日期	初次领证时间	有效起始时间	有效结束时间	地址	执照号码	准驾车型
小良	男	2000-01-13	2020-12-18	2020-12-18	2028-12-18	湖北	T55UI	B2
小芳	女	2000-01-03	2021-01-04	2021-01-05	2029-01-05	北京	AA327	A3
张三	女	2000-01-03	2020-11-24	2020-11-14	2026-11-14	上海	908OP	C1
李四	男	1999-01-28	2020-08-07	2020-08-07	2028-08-07	武汉	TSH66	C2
王五	男	1998-01-14	2021-01-01	2021-01-01	2029-01-01	浙江	MA78S	C1

图 4.2.7 刷新功能及主面板展示

(8) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.2.8 所示。





图 4.2.8 司机档案处理模块组织

#### 4.2.2 持久层

使用数据库 mysql 中表 driver 持久存储用户信息，所有用户的信息均以这样的格式



存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符 (url) 的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

### 4.2.3 实体类 Driver

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.2.9 所示。

Driver
-name:String -sex:Char -birthday:LocalDate -address:String -certificateTime:LocalDate -licenseNumber:String -drivingLicense:String -startTime:LocalDate -endTime:LocalDate
提供了每个成员变量的 getter 和 setter 方法,还有全参构造方法的构造方法,对对应单词完成相应注释并重写了 toString 方法

图 4.2.9 Driver 类图

#### 4.2.4 操作层 Dao 层

该层 (DriverDao) 主要完成对数据库中 Driver 的相关操作。

(1) 录入 (增加) 的方法

```
public static void save(Driver driver);
```

该方法通过封装 Driver 实现对数据库中增加司机档案。

(2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中司机全部信息的获取, 返回 String 类型的集合。

(3) 获取司机姓名的方法

```
public static Object[] getDriverNames();
```

(4) 通过司机姓名进行筛选的方法。

```
public static List<String> seek(String name);
```

通过筛选返回 String 类型的集合。

(5) 获取某行司机全部信息的方法

```
public static List<Driver> getAll();
```

返回 Driver 类型的集合。

(6) 更新信息 (编辑) 的方法

```
public static void remove(Driver driver, Driver info);
```

定位某行信息对其进行更新。

(7) 删除某行信息的方法

```
public static void delete(Driver de);
```

定位某行对该行信息进行筛选。

#### 4.2.5 界面层 (窗体层)

本层实现了对窗体内位置的布局和功能按钮的实现。

(1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置, 如图 4.2.10 所示。



图 4.2.10 顶部布局展示

核心代码如下：

//表格

```

JPanel qP = new JPanel(new BorderLayout(10, 0));
//表单录入司机信息
JPanel aP = new JPanel(null);
JPanel eP = new JPanel(null);
JPanel pP = new JPanel(null);
Map<String, Component> panelsMap = new HashMap<>();
panelsMap.put("主面板", qP);
panelsMap.put("录入", aP);
panelsMap.put("打印", pP);

JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);
panel.add(jTabbedPane, BorderLayout.CENTER);
    
```

```

String[] columns={"姓名","性别","出生日期","初次领证时间","有效起始时间",
    "有效结束时间","地址","执照号码","准驾车型"};
    
```

```

DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
    
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.2.11 所示。



图 4.2.11 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();  
    JMenuItem dMitem = new JMenuItem("删除选定行");  
    JMenuItem qMitem = new JMenuItem("筛选");  
    JMenuItem rMitem = new JMenuItem("刷新");  
    JMenuItem eMitem = new JMenuItem("编辑");  
    //鼠标监听器  
    MouseInputListener mouseInputListener =  
ExtTable.buildMouseInputMenuListener(mMap);  
    table.addMouseListener(mouseInputListener);
```

## 4.3 车辆档案处理模块的设计与实现

### 4.3.1 显示界面和功能界面

#### (1) 信息录入

管理员在进行车辆档案录入时需输入车号、登记时间、发证时间、类型、车主、发动机号、车驾号、汽车品牌、核定载客数、颜色录入界面如图 4.3.1 所示。

主面板打印录入

车辆档案录入

车号:

2

登记时间:

2020-01-13

发证时间:

2020-12-09

类型:

卡车

车主:

张三

发动机号:

34

车驾号:

90DSU

汽车品牌:

大众

核定载客数:

3

颜色:

绿

确定

图 4.3.1 录入界面

(2) 车辆档案主面板

信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.3.2 所示。

主面板打印录入

车号	登记时间	发证时间	类型	车主	发动机号	车驾号	汽车品牌	核定载客数	颜色
1	2021-01-07	2021-01-07	A3	小明	788	6788S	奥迪	4	red
2	2020-01-13	2020-12-09	卡车	张三	34	90DSU	大众	3	绿
3	2020-11-05	2021-01-21	大型卡车	李四	91	US920	大众	3	灰

图 4.3.2 主面板界面

## (3) 打印界面

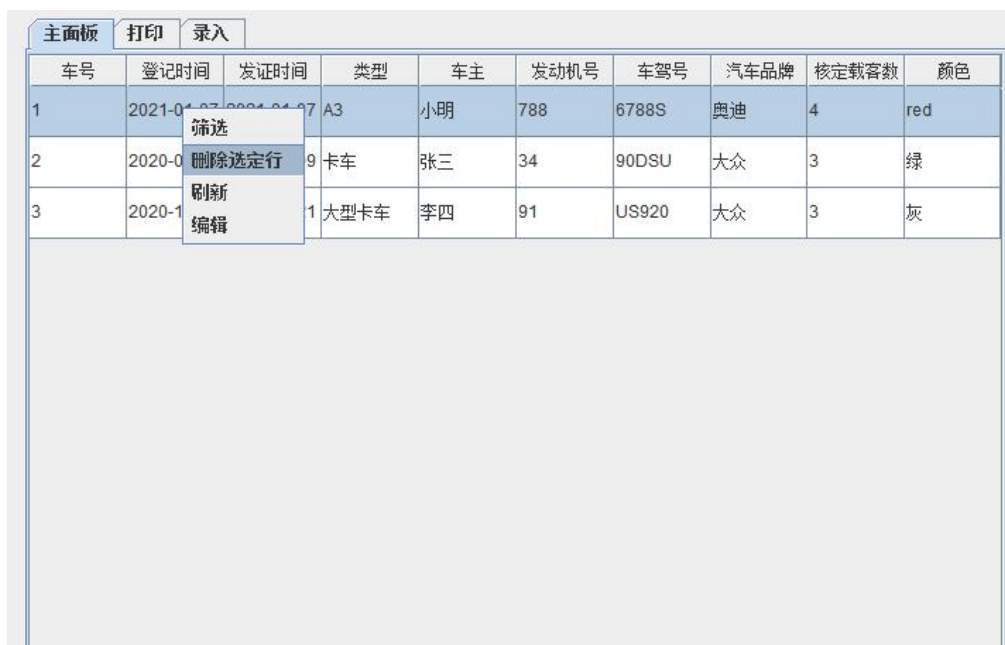
打印界面显示出车辆相关信息的语句，如图 4.3.3 所示。



图 4.3.3 打印界面

## (4) 删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.3.4 所示。



## 中北大学面向对象与数据库课程设计实训说明书

主面板		打印	录入							
车号	登记时间	发证时间	类型	车主	发动机号	车驾号	汽车品牌	核定载客数	颜色	
2	2020-01-13	2020-12-09	卡车	张三	34	90DSU	大众	3	绿	
3	2020-11-05	2021-01-21	大型卡车	李四	91	US920	大众	3	灰	

图 4.3.4 删除本行信息及自动刷新后主面板展示

### (5) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改车辆信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.3.5 所示。

主面板

打印

录入

车号	登记时间	发证时间	类型	车主	发动机号	车驾号	汽车品牌	核定载客数	颜色
2	2020-01-13	2020-12-09	卡车	张三	34	90DSU	大众	3	绿
3	2020-11-05	2021-01-21	大型卡车	李四	91	US920	大众	3	灰

刷新

删除

选定

行

筛选

编辑

主面板	打印	录入	编辑
-----	----	----	----

车辆档案录入

车号:	<input type="text" value="2"/>
登记时间:	<input type="text" value="2020-01-13"/>
发证时间:	<input type="text" value="2020-12-09"/>
类型:	<input type="text" value="卡车"/>
车主:	<input type="text" value="张三"/>
发动机号:	<input type="text" value="34"/>
车驾号:	<input type="text" value="90DSU"/>
汽车品牌:	<input type="text" value="大众"/>
核定载客数:	<input type="text" value="3"/>
颜色:	<input type="text" value="绿"/>

确定

主面板	打印	录入	编辑
-----	----	----	----

车辆档案录入

车号:	<input type="text" value="2"/>
登记时间:	<input type="text" value="2020-01-13"/>
发证时间:	<input type="text" value="2020-12-09"/>
类型:	<input type="text" value="卡车"/>
车主:	<input type="text" value="张三"/>
发动机号:	<input type="text" value="77"/>
车驾号:	<input type="text" value="90DSU"/>
汽车品牌:	<input type="text" value="大众"/>
核定载客数:	<input type="text" value="2"/>
颜色:	<input type="text" value="绿"/>

确定



主面板 打印 录入									
车号	登记时间	发证时间	类型	车主	发动机号	车驾号	汽车品牌	核定载客数	颜色
2	2020-01-13	2020-12-09	卡车	张三	77	90DSU	大众	2	绿
3	2020-11-05	2021-01-21	大型卡车	李四	91	US920	大众	3	灰

图 4.3.5 编辑功能的过程及主面板展示

#### (6) 筛选功能

点击筛选功能后弹出对话框，让管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4.3.6 所示。

主面板 打印 录入									
车号	登记时间	发证时间	类型	车主	发动机号	车驾号	汽车品牌	核定载客数	颜色
2	2020-01-13	2020-12-09	卡车	张三	77	90DSU	大众	2	绿
3	2020-11-05	2021-01-21	大型卡车	李四	91	US920	大众	3	灰

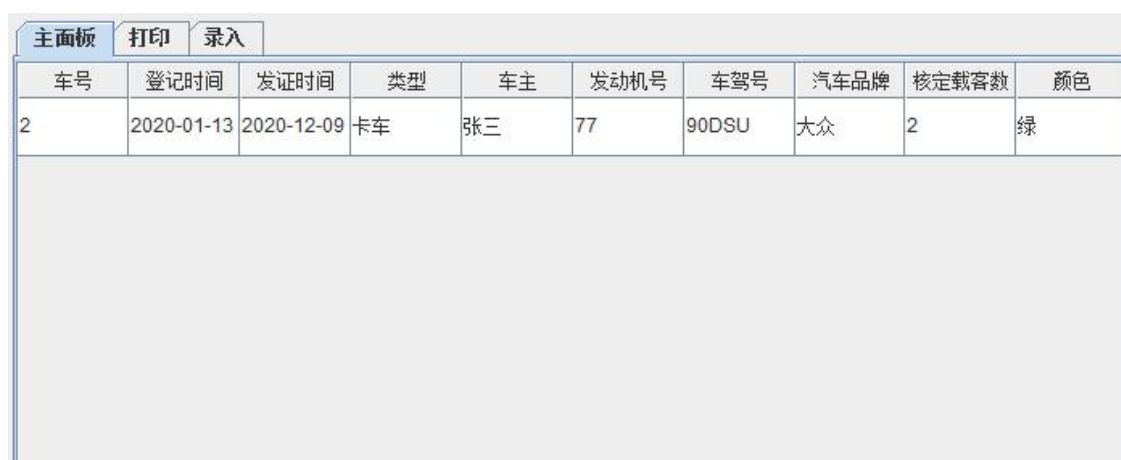


图 4.3.6 筛选功能及主面板展示

#### (7) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.3.7 所示。

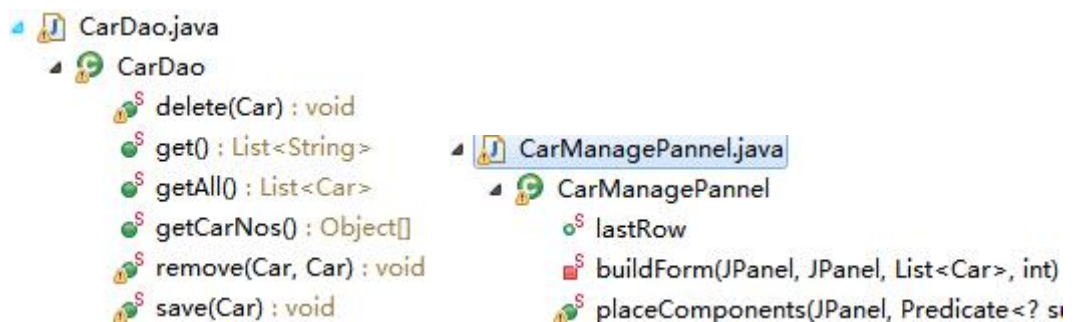
## 中北大学面向对象与数据库课程设计实训说明书



图 4.3.7 刷新功能及主面板展示

(8) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.3.8 所示。



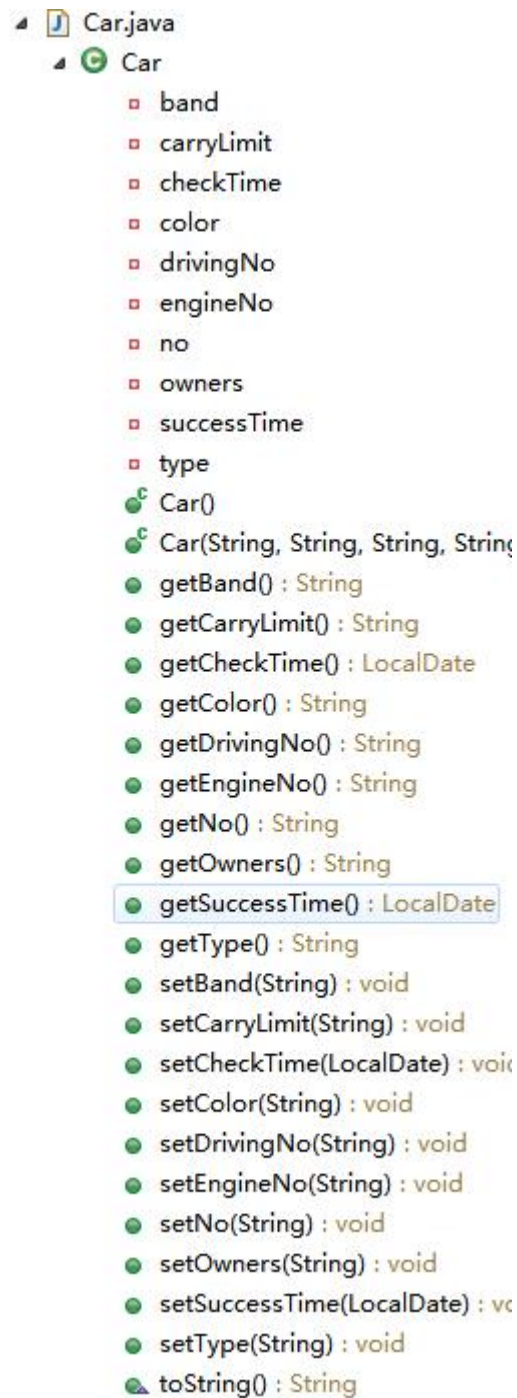


图 4.3.8 汽车案处理模块组织

#### 4.3.2 持久层

使用数据库 mysql 中表 car 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

### 4.3.3 实体类 Car

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.3.9 所示。

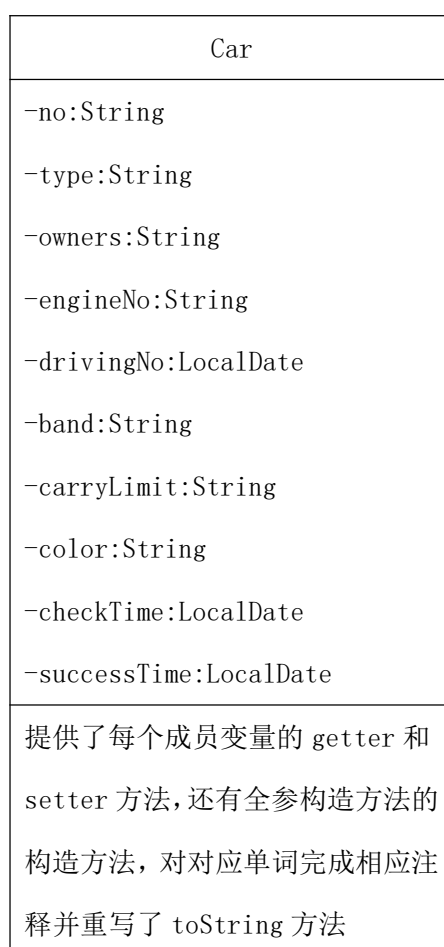


图 4.3.9 Car 类图

### 4.3.4 操作层 Dao 层

该层（CarDao）主要完成对数据库中 Car 的相关操作。

(1) 录入（增加）的方法

```
public static void save(Car car);
```

该方法通过封装 Car 实现对数据库中增加汽车档案。

(2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中汽车全部信息的获取，返回 String 类型的集合。

(3) 获取汽车车号的方法

```
public static Object[] getCarNos();
```

通过筛选返回 String 类型的集合。

(4) 获取某行汽车全部信息的方法

```
public static List<Car> getAll();
```

返回 Car 类型的集合。

(5) 更新信息（编辑）的方法

```
public static void remove(Car car, Car info);
```

定位某行信息对其进行更新。

(6) 删除某行信息的方法

```
public static void delete(Car cr);
```

定位某行对该行信息进行筛选。

### 4.3.5 界面层（窗体层）

本层实现了对窗体内位置的布局和功能按钮的实现。

(1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.3.10 所示。



图 4.3.10 顶部布局展示

核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));  
//表单录入车辆信息  
JPanel aP = new JPanel(null);  
JPanel eP = new JPanel(null);  
JPanel pP = new JPanel(null);  
Map<String, Component> panelsMap = new HashMap<>();  
panelsMap.put("主面板", qP);  
panelsMap.put("录入", aP);  
panelsMap.put("打印", pP);  
  
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);  
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"车号","登记时间","发证时间","类型","车主","发动机号",  
"车驾号","汽车品牌","核定载客数","颜色"};
```

```
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.3.11 所示。



图 4.3.11 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();

JMenuItem dMitem = new JMenuItem("删除选定行");
JMenuItem qMitem = new JMenuItem("筛选");
JMenuItem rMitem = new JMenuItem("刷新");
JMenuItem eMitem = new JMenuItem("编辑");

//鼠标监听器

MouseListener mouseInputListener =
ExtTable.buildMouseListener(mMap);

table.addMouseListener(mouseInputListener);
```

## 4.4 车辆日常费用管理模块的设计与实现

### 4.4.1 显示界面和功能界面

#### (1) 信息录入

管理员在进行车辆日常费用录入时需输入车号（已有汽车车号）、维修时间、维修费用（元）、维修单位、维修内容、维修配件、备注，界面如图 4.4.1 所示。



图 4.4.1 录入界面

## (2) 维修档案主面板

信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.4.2 所示。

主面板		打印	录入			
车牌号	维修时间	维修费用（元）	维修单位	维修内容	维修配件	备注
2	2021-01-05	400.0	卡车	对于远光灯的维修	远光灯	左侧远光灯
3	2021-01-04	3000.0	大型卡车	对车门的维修	车门	顺便检查下轮胎

图 4.4.2 主面板界面

## (3) 打印界面

打印界面显示出维修相关信息的语句，如图 4.4.3 所示。

## 中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入						
车牌号	维修时间	维修费用（元）	维修单位	维修内容	维修配件	备注
2	2021-01-05	400.0	卡车	对于远光灯的维修	远光灯	左侧远光灯
3	2021-01-04	3000.0	大型卡车	对车门的维修	车门	顺便检查下轮胎

图 4.4.3 打印界面

### （4）删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.4.4 所示。

主面板 打印 录入						
车牌号	维修时间	维修费用（元）	维修单位	维修内容	维修配件	备注
2	2021-01-05	400.0	卡车	对于近光灯的维修	近光灯	左侧近光灯
3	2021-01-04		大型卡车	对车门的维修	车门	顺便检查下轮胎

主面板 打印 录入						
车牌号	维修时间	维修费用（元）	维修单位	维修内容	维修配件	备注
3	2021-01-04	3000.0	大型卡车	对车门的维修	车门	顺便检查下轮胎

图 4.4.4 删除本行信息及自动刷新后主面板展示

(5) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改车辆信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.4.5 所示。

车牌号	维修时间	维修费用(元)	维修单位	维修内容	维修配件	备注
2	筛选	400.0	卡车	对于远光灯的维修	远光灯	左侧远光灯
3	编辑	3000.0	大型卡车	对车门的维修	车门	顺便检查下轮胎
	刷新					
	删除选定行					

**车辆费用录入**

车号: 2

维修时间: 2021-01-05

维修费用(元): 400.0

维修单位: 卡车

维修内容: 对于远光灯的维修

维修配件: 远光灯

备注: 左侧远光灯

确定

### 车辆费用录入

车号:

2

维修时间:

2021-01-05

维修费用 (元):

400.0

维修单位:

卡车

维修内容:

对于近光灯的维修

维修配件:

近光灯

备注:

左侧近光灯

确定

主面板
打印
录入

车牌号	维修时间	维修费用 (元)	维修单位	维修内容	维修配件	备注
2	2021-01-05	400.0	卡车	对于近光灯的维修	近光灯	左侧近光灯
3	2021-01-04	3000.0	大型卡车	对车门的维修	车门	顺便检查下轮胎

图 4.4.5 编辑功能的过程及主面板展示

#### (6) 筛选功能

点击筛选功能后弹出对话框，让管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4.4.6 所示。

## 中北大学面向对象与数据库课程设计实训说明书

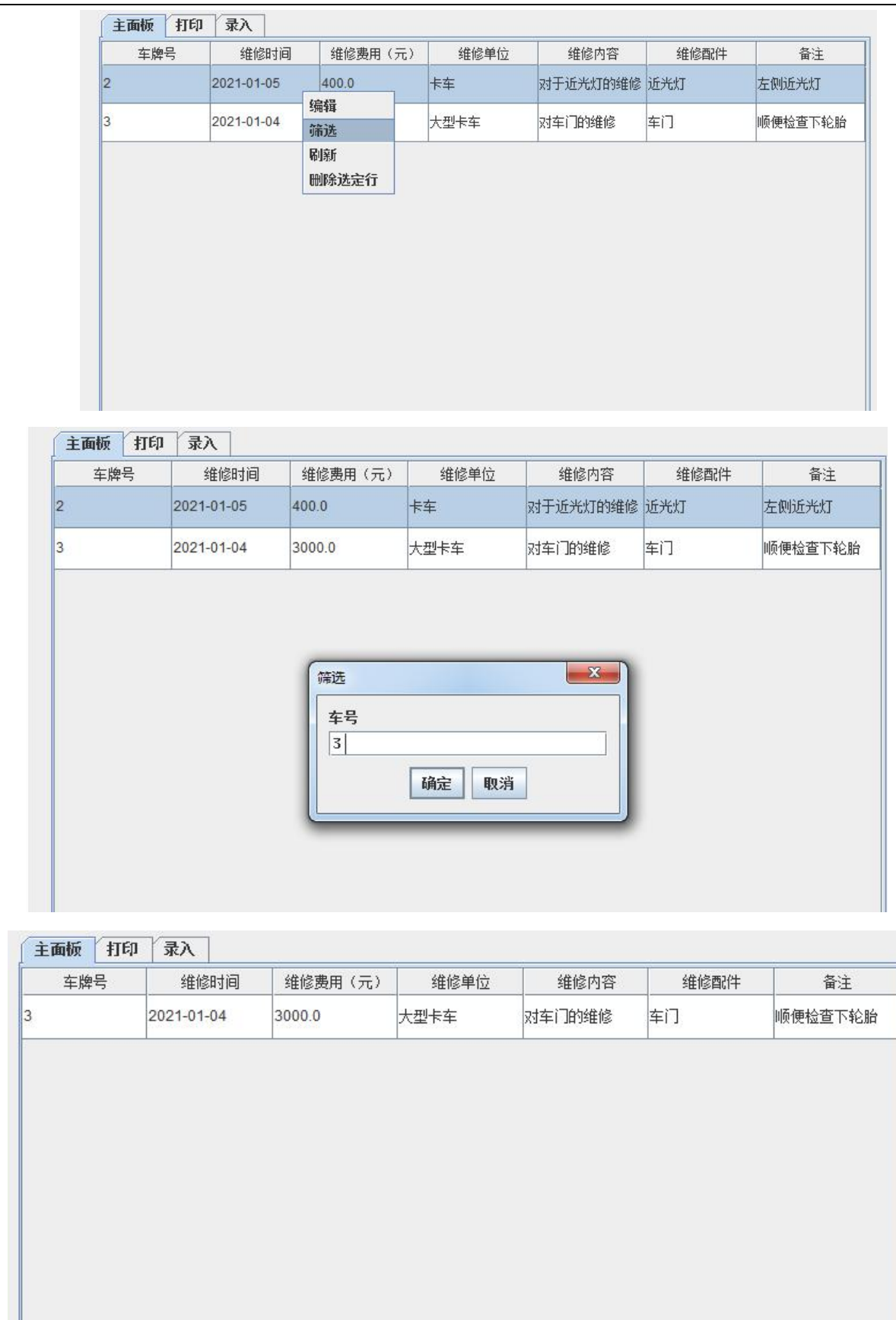


图 4.4.6 筛选功能及主面板展示

### (7) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.4.7 所示。



图 4.4.7 刷新功能及主面板展示

### (8) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.4.8 所示。

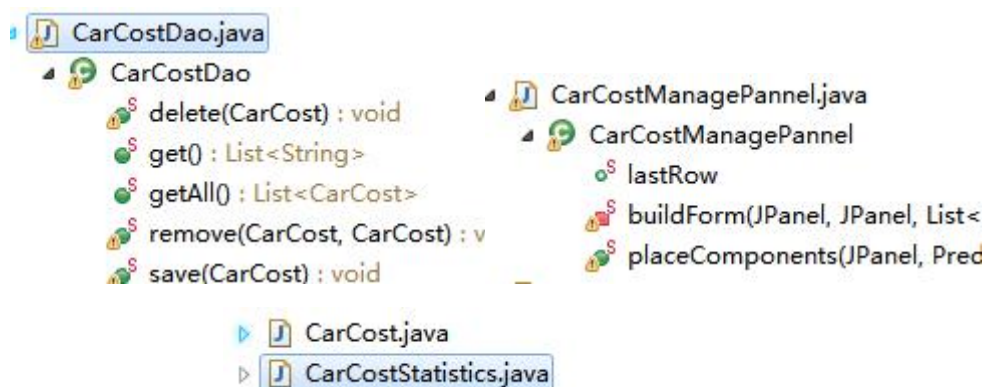


图 4.4.8 维修费用处理模块组织

#### 4.4.2 持久层

使用数据库 mysql 中表 carcost 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

#### 4.4.3 实体类 Carcost

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.4.9 所示。

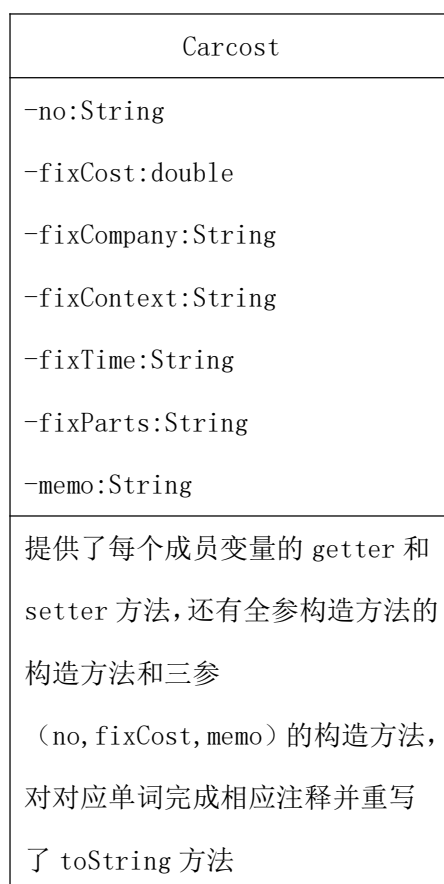


图 4.4.9 Carcost 类图

#### 4.4.4 操作层 Dao 层

该层（CarCostDao）主要完成对数据库中 Carcost 的相关操作。

##### （1）录入（增加）的方法

```
public static void save(Carcost carcost);
```

该方法通过封装 Carcost 实现对数据库中增加汽车档案。

##### （2）获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中维修全部信息的获取，返回 String 类型的集合。

##### （3）获取某行维修费用全部信息的方法

```
public static List<Carcost> getAll();
```

返回 Carcost 类型的集合。

##### （4）更新信息（编辑）的方法

```
public static void remove(Carcost carcost, Carcost info);
```

定位某行信息对其进行更新。

##### （5）删除某行信息的方法

```
public static void delete(Carcost cor);
```

定位某行对该行信息进行筛选。

#### 4.4.5 界面层（窗体层）

本层实现了对窗体内位置的布局和功能按钮的实现。

##### （1）对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.4.10 所示。

主面板 打印 录入						
车牌号	维修时间	维修费用（元）	维修单位	维修内容	维修配件	备注

图 4.4.10 顶部布局展示



核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));  
//表单录入费用信息  
JPanel aP = new JPanel(null);  
JPanel eP = new JPanel(null);  
JPanel pP = new JPanel(null);  
Map<String, Component> panelsMap = new HashMap<>();  
panelsMap.put("主面板", qP);  
panelsMap.put("录入", aP);  
panelsMap.put("打印", pP);  
  
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);  
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"车牌号","维修时间","维修费用（元）","维修单位","维修  
内容","维修配件","备注"};
```

```
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.4.11 所示。



图 4.4.11 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();

JMenuItem dMitem = new JMenuItem("删除选定行");
JMenuItem qMitem = new JMenuItem("筛选");
JMenuItem rMitem = new JMenuItem("刷新");
JMenuItem eMitem = new JMenuItem("编辑");

//鼠标监听器

MouseListener mouseInputListener =
ExtTable.buildMouseListener(mMap);

table.addMouseListener(mouseInputListener);
```

## 4.5 油卡管理模块的设计与实现

### 4.5.1 显示界面和功能界面

#### (1) 信息录入

管理员在进行油卡信息录入时需输入油卡编号、石油品牌、录入时间、收支类型（若选择充值则出现充值金额（元）文本框，若选择消费则弹出消费金额（元），和消费车号）、备注，界面如图 4.5.1 所示。

主面板 打印 录入

车辆费用录入

油卡编号: 26edfgq8

石油品牌: 95

录入时间: 2021-01-07

收支类型: ☒ 充值 ☐ 消费

充值金额(元): 6666

备注: 无

确定

图 4.5.1 录入界面

## (2) 油卡档案主面板

信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.5.2 所示。

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额(元)	充值编号	充值金额(元)	记录时间	备注
26edfgq8	95	null		0.0	3438dfc7-ac...	6666.0	2021-01-07	无
gdgser6326	97	bba1af47-4...	3	500.0	null	0.0	2021-01-10	无

图 4.5.2 主面板界面

## (3) 打印界面

打印界面显示出油卡相关信息的语句，如图 4.5.3 所示。

## 中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 (...	充值编号	充值金额 (...	记录时间	备注
26edfgq8	95	null		0.0	3438dfc7-ac...	6666.0	2021-01-07	无
gdgser6326	97	bba1af47-4...	3	500.0	null	0.0	2021-01-10	无

图 4.5.3 打印界面

### (4) 删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.5.4 所示。

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 (...	充值编号	充值金额 (...	记录时间	备注
26edfgq8	95	null		0.0	767a8427-5...	6666.0	2021-01-07	首次充值
gdgser6326	97	bba1af47-4...		500.0	null	0.0	2021-01-10	无

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 (...	充值编号	充值金额 (...	记录时间	备注
gdgser6326	97	bba1af47-4...	3	500.0	null	0.0	2021-01-10	无

图 4.5.4 删除本行信息及自动刷新后主面板展示

### (5) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改油卡信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.5.5 所示。

## 中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 (...	充值编号	充值金额 (...	记录时间	备注
26edfgq8	95			0.0	3438dfc7-ac...	6666.0	2021-01-07	无
gdgser6326	97		3	500.0	null	0.0	2021-01-10	无

编辑  
删除选定行  
刷新  
筛选

主面板	打印	录入	编辑
-----	----	----	----

车辆费用录入

油卡编号: 26edfgq8

石油品牌: 95

录入时间: 2021-01-07

收支类型: ☐ 充值  
☐ 消费

充值金额 (元): 6666.0

备注: 无

确定

主面板	打印	录入	编辑
-----	----	----	----

车辆费用录入

油卡编号: 26edfgq8

石油品牌: 95

录入时间: 2021-01-07

收支类型: ☐ 充值  
☐ 消费

充值金额 (元): 6666.0

备注: 首次充值

确定

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 ( ...	充值编号	充值金额 ( ...	记录时间	备注
26edfgq8	95	null		0.0	767a8427-5...	6666.0	2021-01-07	首次充值
gdgser6326	97	bba1af47-4...	3	500.0	null	0.0	2021-01-10	无

图 4. 5. 5 编辑功能的过程及主面板展示

#### (6) 筛选功能

点击筛选功能后弹出对话框，让管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4. 5. 6 所示。

主面板 打印 录入								
油卡编号	石油品牌	消费编号	消费车号	消费金额 ( ...	充值编号	充值金额 ( ...	记录时间	备注
26edfgq8	95	null		0.0	767a8427-5...	6666.0	2021-01-07	首次充值
gdgser6326	97	bba		500.0	null	0.0	2021-01-10	无

筛选  
 编辑  
 刷新  
 删除选定行

## 中北大学面向对象与数据库课程设计实训说明书



图 4.5.6 筛选功能及主面板展示

### (7) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.5.7 所示。



油卡编号	石油品牌	消费编号	消费车号	消费金额 (...)	充值编号	充值金额 (...)	记录时间	备注
26edfgq8	95	null		0.0	767a8427-5...	6666.0	2021-01-07	首次充值
gdgser6326	97	bba1af47-4...	3	500.0	null	0.0	2021-01-10	无

图 4.5.7 刷新功能及主面板展示

(8) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.5.8 所示。

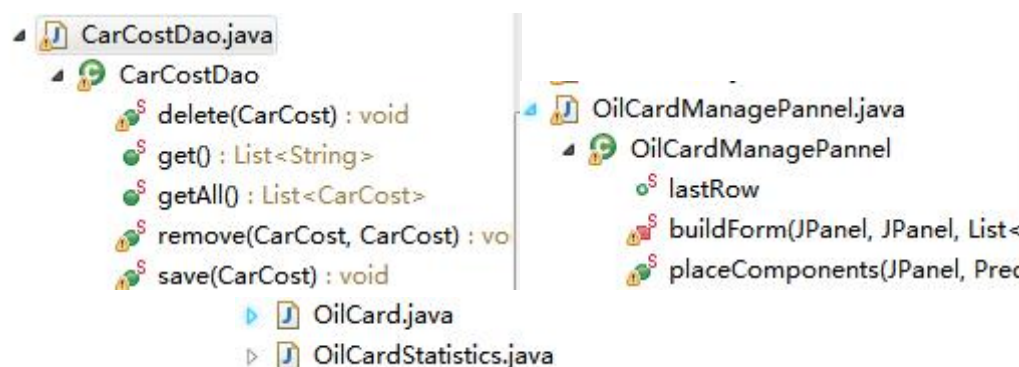


图 4.5.8 油卡处理模块组织

## 4.5.2 持久层

使用数据库 mysql 中表 oilcard 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符 (url) 的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
```



```
con = DriverManager.getConnection(url, user, password);
```

### 4.5.3 实体类 Oilcard

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.5.9 所示。

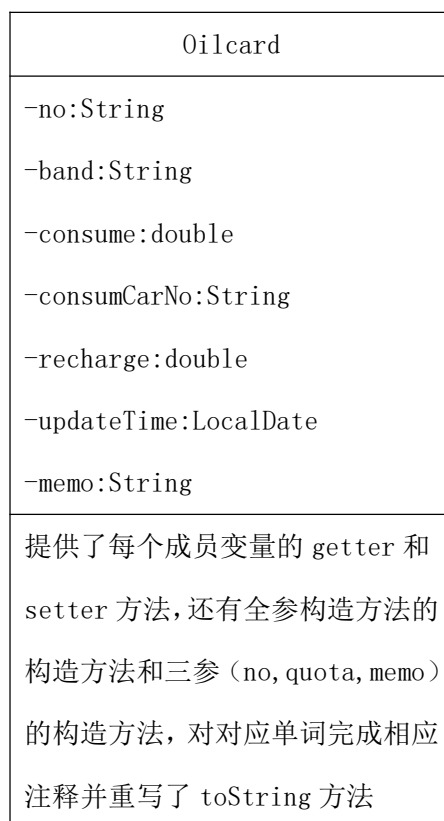


图 4.5.9 Oilcard 类图

### 4.5.4 操作层 Dao 层

该层 (OilcardDao) 主要完成对数据库中 Oilcard 的相关操作。

#### (1) 录入 (增加) 的方法

```
public static void save(Oilcard oilcard);
```

该方法通过封装 Oilcard 实现对数据库中增加汽车档案。

#### (2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中油卡全部信息的获取,返回 String 类型的集合。

#### (3) 获取某行油卡信息全部信息的方法

```
public static List<Oilcard> getAll();
```

返回 Oilcard 类型的集合。

#### (4) 更新信息 (编辑) 的方法

```
public static void remove(Oilcard oilcard, Oilcard info);
```

定位某行信息对其进行更新。

#### (5) 删除某行信息的方法

```
public static void delete(Oilcard oid);
```

定位某行对该行信息进行筛选。

### 4.5.5 界面层（窗体层）

本层实现了对窗体内位置的布局和功能按钮的实现。

#### (1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.5.10 所示。



图 4.5.10 顶部布局展示

核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));
//表单录入油卡信息
JPanel aP = new JPanel(null);
JPanel eP = new JPanel(null);
JPanel pP = new JPanel(null);
Map<String, Component> panelsMap = new HashMap<>();
panelsMap.put("主面板", qP);
panelsMap.put("录入", aP);
panelsMap.put("打印", pP);
```

```
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"油卡编号","石油品牌","消费编号","消费车号","消费金额  
(元)","充值编号","充值金额(元)","记录时间","备注"};
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.5.11 所示。

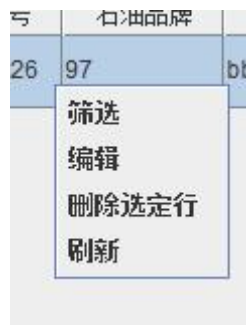


图 4.5.11 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();
JMenuItem dMitem = new JMenuItem("删除选定行");
JMenuItem qMitem = new JMenuItem("筛选");
JMenuItem rMitem = new JMenuItem("刷新");
JMenuItem eMitem = new JMenuItem("编辑");
//鼠标监听器
MouseListener mouseInputListener =
ExtTable.buildMouseListener(mMap);
table.addMouseListener(mouseInputListener);
```

## 4.6 司机任务&&出派车辆管理的设计与实现

\*因为这两个功能相似所以放在一起说明。

### 4.6.1 显示界面和功能界面

#### (1) 信息录入及显示

管理员在进行司机任务录入时需输入相应信息，界面如图 4.6.1 所示。

主面板 打印 录入

车辆费用录入

车号: 2

司机姓名: 小良

出发时间: 2021-01-04

出发地: 北京

目的地: 天津

确定

图 4.6.1 管理员录入信息界面

#### (2) 出派车辆管理主面板

信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.6.2 所示。

## 中北大学面向对象与数据库课程设计实训说明书

主面板 打印 录入				
车号	司机姓名	出发时间	出发地	目的地
3	小芳	2021-01-11	北京	武汉
2	李四	2021-01-04	西藏	新疆
2	小良	2021-01-04	北京	天津

图 4.6.2 主面板界面

### (3) 打印界面

打印界面显示出油卡相关信息的语句，如图 4.6.3 所示。

主面板 打印 录入				
车号	司机姓名	出发时间	出发地	目的地
3	小芳	2021-01-11	北京	武汉
2	李四	2021-01-04	西藏	新疆
2	小良	2021-01-04	北京	天津

图 4.6.3 打印界面

### (4) 删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.6.4 所示。

主面板 打印 录入				
车号	司机姓名	出发时间	出发地	目的地
3	小芳	2021-01-11	北京	武汉
2	李四	2021-01-04	西藏	新疆
2	小良	2021-01-04	北京	天津

主面板 打印 录入				
车号	司机姓名	出发时间	出发地	目的地
2	李四	2021-01-04	西藏	新疆
2	小良	2021-01-04	北京	天津

图 4.6.4 删除本行信息及自动刷新后主面板展示

(5) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改出派车辆信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.6.5 所示。

主面板 打印 录入

车号	司机姓名	出发时间	出发地	目的地
2	李四	2021-01-04	西藏	新疆
2	小良	2021-01-04	北京	天津

筛选

刷新

编辑

删除选定行

主面板 打印 录入 编辑

车辆费用录入

车号: 2

司机姓名: 李四

出发时间: 2021-01-04

出发地: 西藏

目的地: 新疆

确定

主面板
打印
录入
编辑

车辆费用录入

车号: 3

司机姓名: 李四

出发时间: 2021-01-04

出发地: 合肥

目的地: 公司本部

确定

主面板
打印
录入

车号	司机姓名	出发时间	出发地	目的地
3	李四	2021-01-04	合肥	公司本部
2	小良	2021-01-04	北京	天津

图 4.6.5 编辑功能的过程及主面板展示

## (6) 筛选功能

点击筛选功能后弹出对话框，让管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4.6.6 所示。

主面板
打印
录入

车号	司机姓名	出发时间	出发地	目的地
3	李四	2021-01-04	合肥	公司本部
2		2021-01-04	北京	天津

删除选定行
编辑
刷新
筛选

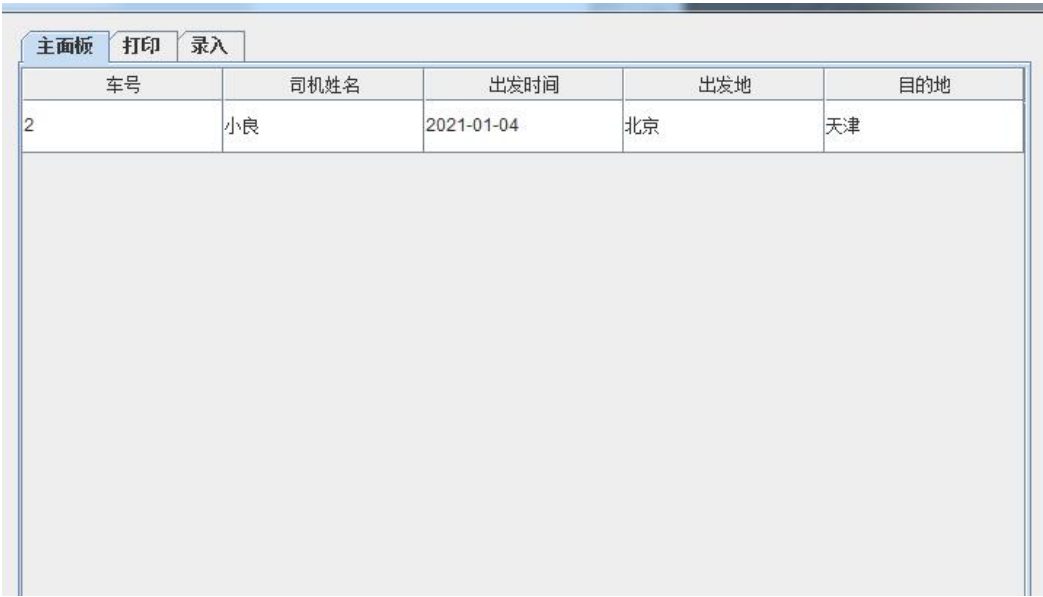
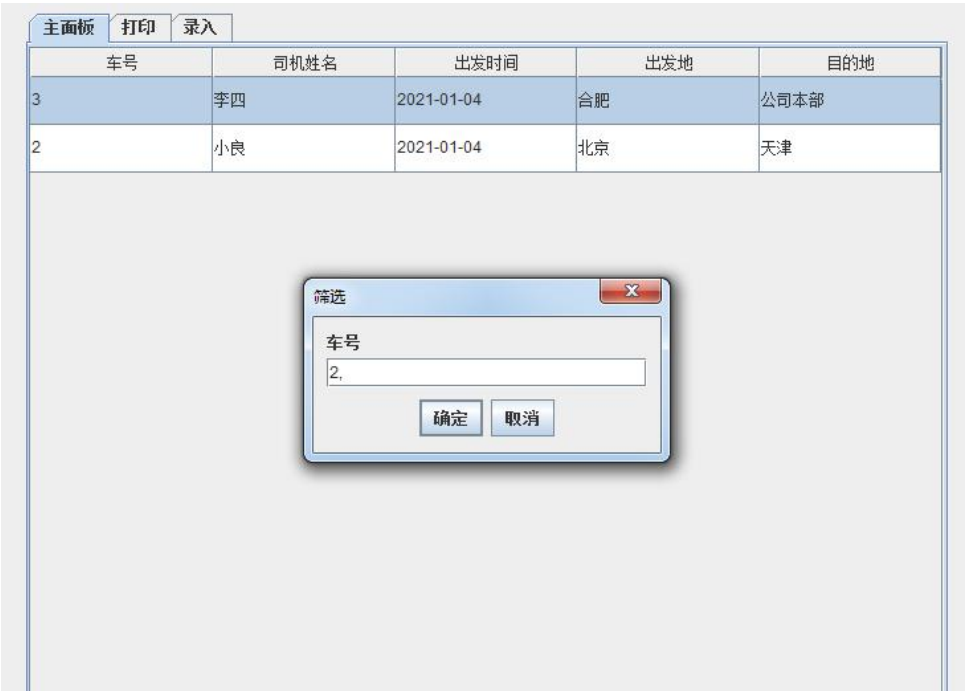


图 4.6.6 筛选功能及主面板展示

(7) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.6.7 所示。





图 4.6.7 刷新功能及主面板展示

(8) 司机登录任务面板展示

司机登录后点击“查看任务”即可显示自己的任务，但只有查看，筛选操作，如图 4.6 所示。

主面板 打印				
车号	司机姓名	出发时间	出发地	目的地
2	小良	2021-01-04	北京	天津

图 4.6.8 司机登录页面任务显示

(9) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.6.9 所示。

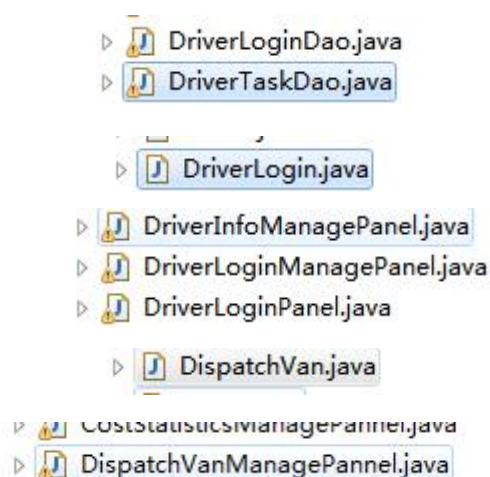


图 4.6.9 出派车辆处理模块组织

## 4.6.2 持久层

使用数据库 mysql 中表 delivery 持久存储出派车辆信息，deiverlogin 持久存储司机账户信息信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

#### 4.6.3 实体类 DispatchVan, DriverLogin

DispatchVan 类主要用于出派车辆的信息，类图如图 4.6.4 所示。

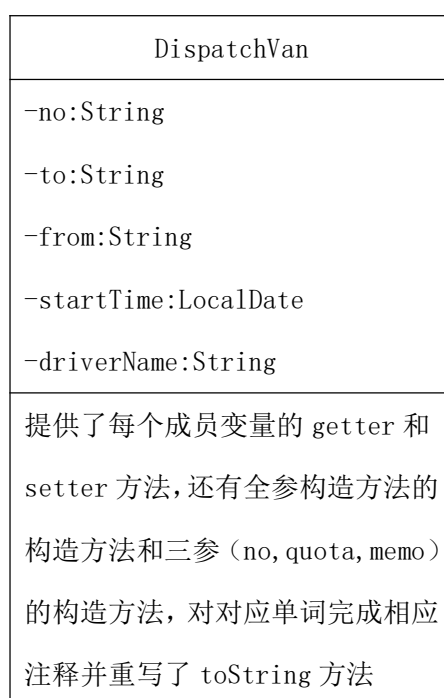
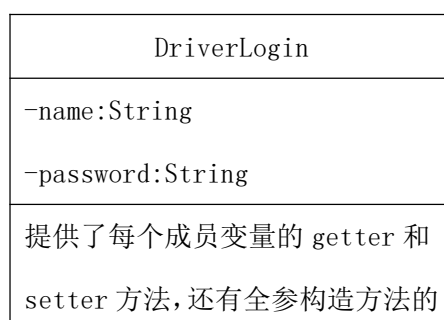


图 4.6.4 DispatchVan 类图

DriverLogin 类主要用于司机的信息，类图如图 4.6.5 所示。



构造方法和三参 (no, quota, memo) 的构造方法, 对对应单词完成相应注释并重写了 toString 方法

图 4.6.5 DriverLogin 类图

#### 4.6.4 操作层 Dao 层

该层 (DispatchVanDao) 主要完成对数据库中 delivery 的相关操作。

##### (1) 录入 (增加) 的方法

```
public static void save(DispatchVan dispatchVan);
```

该方法通过封装 DispatchVan 实现对数据库中增加汽车档案。

##### (2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中出派车辆全部信息的获取, 返回 String 类型的集合。

##### (3) 获取某行油卡信息全部信息的方法

```
public static List<DispatchVan> getAll();
```

返回 DispatchVan 类型的集合。

##### (4) 更新信息 (编辑) 的方法

```
public static void remove(DispatchVan dispatchVan, DispatchVan info);
```

定位某行信息对其进行更新。

##### (5) 删除某行信息的方法

```
public static void delete(DispatchVan dvd);
```

定位某行对该行信息进行筛选。

#### 4.6.5 界面层 (窗体层)

本层实现了对窗体内位置的布局和功能按钮的实现。

##### (1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置, 如图 4.6.6 所示。



图 4.6.6 顶部布局展示

核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));
```

//表单录入出派信息

```
JPanel aP = new JPanel(null);
```

```
JPanel eP = new JPanel(null);
```

```
JPanel pP = new JPanel(null);
```

```
Map<String, Component> panelsMap = new HashMap<>();
```

```
panelsMap.put("主面板", qP);
```

```
panelsMap.put("录入", aP);
```

```
panelsMap.put("打印", pP);
```

```
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);
```

```
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"车号","司机姓名","出发时间","出发地","目的地"};
```

```
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.6.7 所示。

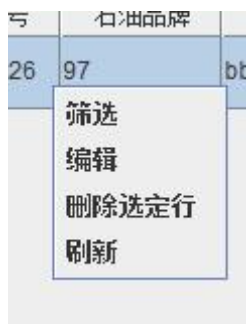


图 4.6.7 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();  
    JMenuItem dMitem = new JMenuItem("删除选定行");  
    JMenuItem qMitem = new JMenuItem("筛选");  
    JMenuItem rMitem = new JMenuItem("刷新");  
    JMenuItem eMitem = new JMenuItem("编辑");  
    //鼠标监听器  
    MouseInputListener mouseInputListener =  
ExtTable.buildMouseInputMenuListener(mMap);  
    table.addMouseListener(mouseInputListener);
```

## 4.7 管理员个人信息模块的设计与实现

司机个人信息管理和管理员相同。

### 4.7.1 显示界面和功能界面

#### (1) 管理员个人信息主面板

普通管理员登录后点击“个人信息处理”跳转到有关自己账户信息页面，如图 4.7.1 所示。

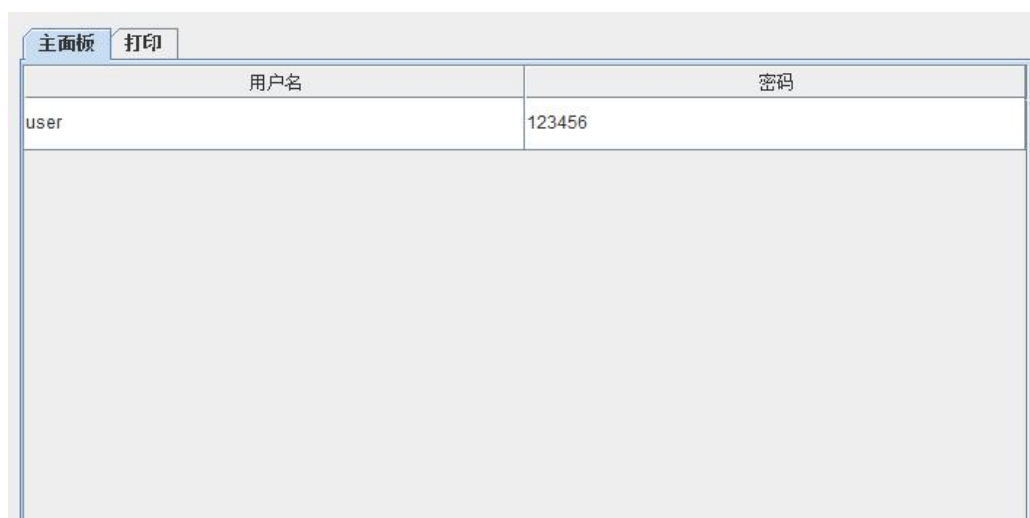


图 4.7.1 主面板界面

#### (2) 打印界面

打印界面显示出管理员相关信息的语句，如图 4.7.2 所示。

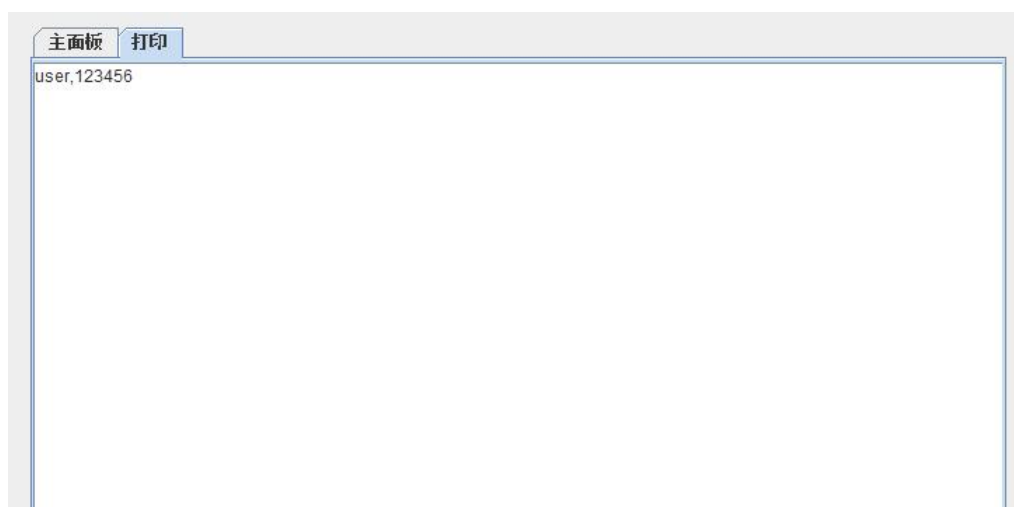


图 4.7.2 打印界面

### (3) 编辑功能

点击编辑键后自动跳转到编辑界面可以修改管理员自己的信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.7.3 所示。

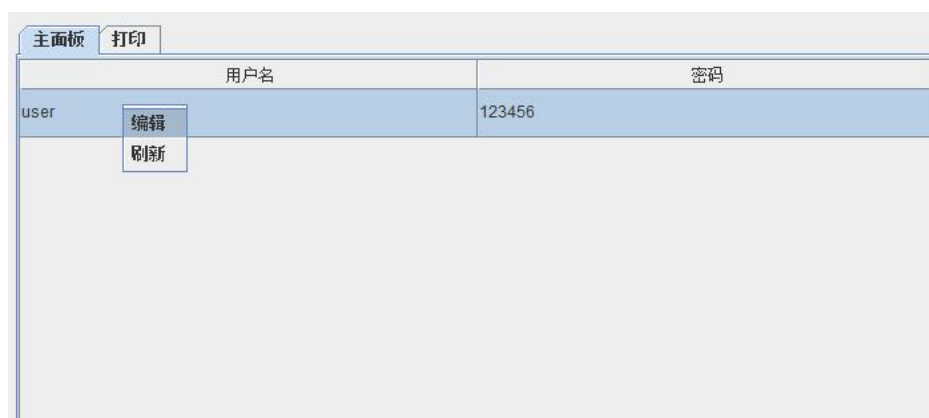




图 4.7.3 编辑功能的过程及主面板展示



#### (4) 刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.7.4 所示。

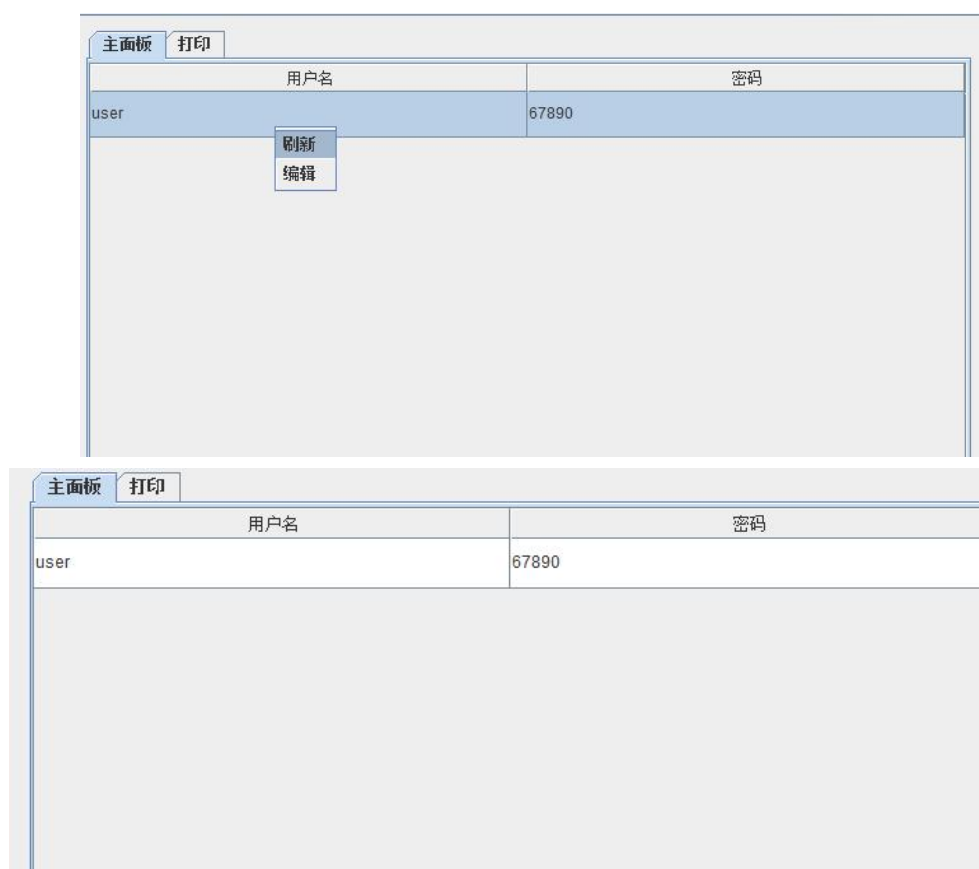


图 4.7.4 刷新功能及主面板展示

#### (5) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.7.5 所示。



图 4.7.5 个人信息处理模块组织

### 4.7.2 持久层

使用数据库 mysql 中表 User 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

### 4.7.3 实体类 User

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.7.6 所示。

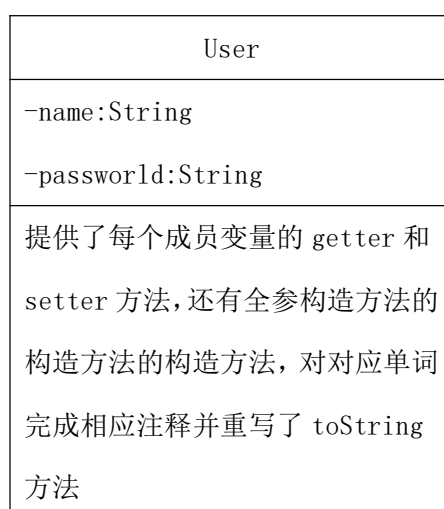


图 4.7.6 User 类图

### 4.7.4 操作层 Dao 层

#### (1) 获取密码的方法

```
public static String queryUserByName(String user) ;
```

该方法通过封装 user 实现登录过程账号密码匹配。

#### (2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中 User 全部信息的获取，返回 String 类型的集合，用于超级管理员对其操作。

(3) 更新信息（编辑）的方法

```
public static void remove(User user, User info);
```

定位某行信息对其进行更新。

#### 4.7.5 界面层（窗体层）

本层实现了对窗体内位置的布局和功能按钮的实现。

(1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.7.5 所示。

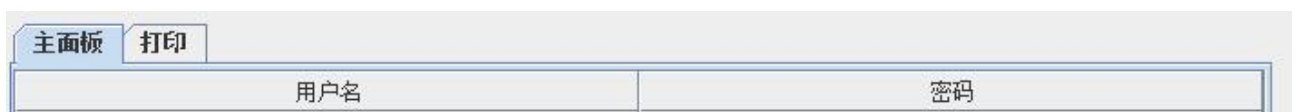


图 4.7.5 顶部布局展示

核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));
```

```
JPanel eP = new JPanel(null);
```

```
JPanel pP = new JPanel(null);
```

```
Map<String, Component> panelsMap = new HashMap<>();
```

```
panelsMap.put("主面板", qP);
```

```
panelsMap.put("打印", pP);
```

```
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);
```

```
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"用户名","密码"};
```

```
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.7.6 所示。

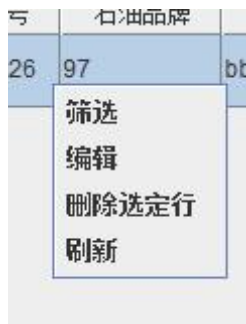


图 4.7.6 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();
    JMenuItem rMitem = new JMenuItem("刷新");
    JMenuItem eMitem = new JMenuItem("编辑");
    //鼠标监听器
    MouseInputListener mouseInputListener =
ExtTable.buildMouseInputMenuListener(mMap);
    table.addMouseListener(mouseInputListener);
```

## 4.8 费用统计模块的设计与实现

### 4.8.1 显示界面和功能界面

(1) 油卡收支统计界面

该显示来自油卡管理的信息，如图 4.8.1 所示。

油卡收支统计	打印油卡收支	打印车辆维修费用	车辆维修费用统计
油卡编号		余额	备注
gdgser6326		-500.0	系统统计

图 4.8.1 油卡收支统计界面

(2) 打印油卡收支界面

打印界面显示出油卡收支相关信息的语句，如图 4.8.2 所示。

油卡收支统计	打印油卡收支	打印车辆维修费用	车辆维修费用统计
gdgser6326,-500.0,系统统计			

图 4.8.2 打印界面

(3) 打印车辆维修费用界面

打印界面显示出车辆维修费用相关信息的语句，如图 4.8.3 所示。



图 4.8.3 打印界面

(4) 车辆维修费用统计界面

该显示来自车辆维修费用的信息，如图 4.8.4 所示。



图 4.8.4 油卡收支统计界面

(5) 该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.8.5 所示。

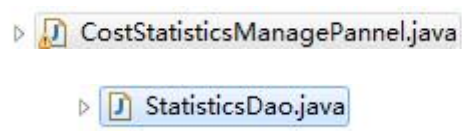


图 4.8.5 费用处理模块组织

## 4.8.2 操作层 Dao 层

该层（StatisticsDao）主要完成的相关操作。

（1）获取某行油卡信息全部信息的方法

```
public static List<String> getOil();
```

返回 String 类型的集合。

（2）获取某行车辆维修信息全部信息的方法

```
public static List<String> getCar();
```

返回 String 类型的集合。

## 4.8.3 界面层（窗体层）

本层实现了对窗体内位置的布局的实现。

（1）对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.8.3 所示。



图 4.8.3 顶部布局展示

核心代码如下：

//表格

```
JPanel oP = new JPanel(new BorderLayout(10, 0));
```

```
JPanel cP = new JPanel(new BorderLayout(10, 0));
```

```
JPanel oPP = new JPanel(null);
```

```
JPanel cPP = new JPanel(null);
```

```
Map<String, Component> panelsMap = new HashMap<>();
```

```
panelsMap.put("油卡收支统计", oP);
```

```
panelsMap.put("车辆维修费用统计", cP);
```

```
panelsMap.put("打印油卡收支", oPP);
```

```
panelsMap.put("打印车辆维修费用", cPP);
```

```
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);
panel.add(jTabbedPane, BorderLayout.CENTER);
```

(2) 顶部布局—油卡收支统计相应的窗体和核心代码

设置了对应信息的位置，如图 4.8.4 所示。

油卡收支统计	打印油卡收支	打印车辆维修费用	车辆维修费用统计
油卡编号	余额	备注	

图 4.8.4 顶部—油卡收支统计布局所示

核心代码如下：

```
String[] columns={"油卡编号","余额","备注"};
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(3) 顶部布局—车辆维修费用统计相应的窗体和核心代码

设置了对应信息的位置，如图 4.8.5 所示。

油卡收支统计	打印油卡收支	打印车辆维修费用	车辆维修费用统计
油卡编号	余额	备注	

图 4.8.5 顶部—车辆维修费用统计布局所示

核心代码如下：

```
String[] columns1={"车牌号","维修总费用（元）","备注"};
DefaultTableModel tableModel1 = new DefaultTableModel(columns1, 0);
```

## 4.9 超级管理员模块的设计与实现

### 4.9.1 显示界面和功能界面

(1) 信息录入



超级管理员在进行普通管理员信息录入时需输入其账号及密码，界面如图 4.9.1 所示。



主面板 打印 录入

管理员档案录入

姓名: jdk

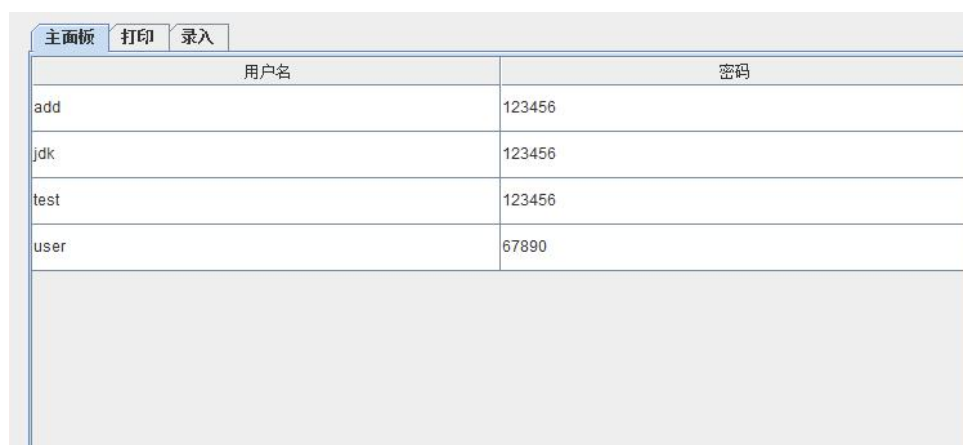
密码: 123456

确定

图 4.9.1 录入界面

## (2) 普通管理员档案主面板

信息正确输入后点击“确定”按钮自动跳转到主面板界面，显示刚录入信息和已有信息，主面板界面如图 4.9.2 所示。



用户名	密码
add	123456
jdk	123456
test	123456
user	67890

图 4.9.2 主面板界面

## (3) 打印界面

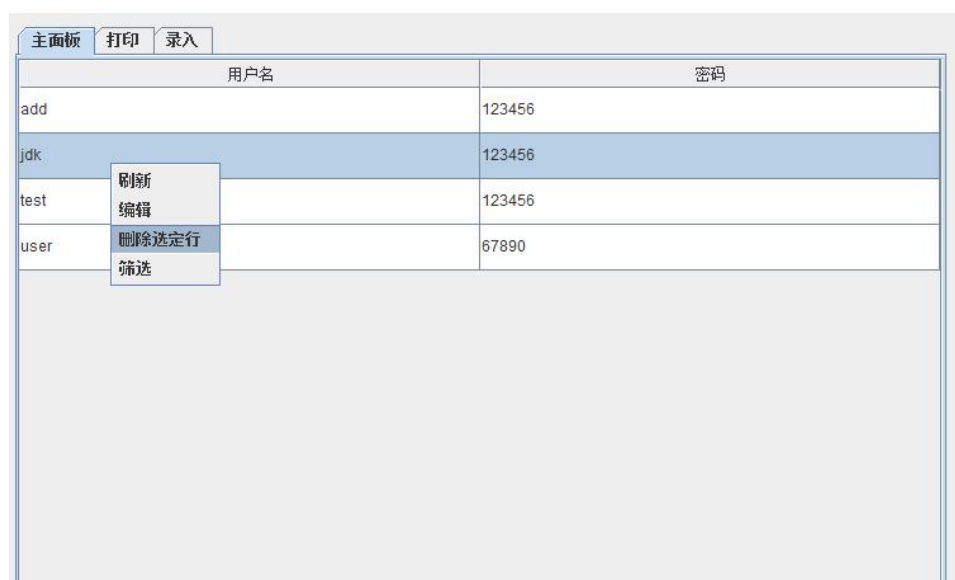
打印界面显示出油卡相关信息的语句，如图 4.9.3 所示。



图 4.9.3 打印界面

## (4) 删除选定行功能

对于主面板的信息，右键可以跳出四个功能选项，其中删除选定行功能可以删除本行信息并且自动刷新，在主面板显示其余信息，如图 4.9.4 所示。





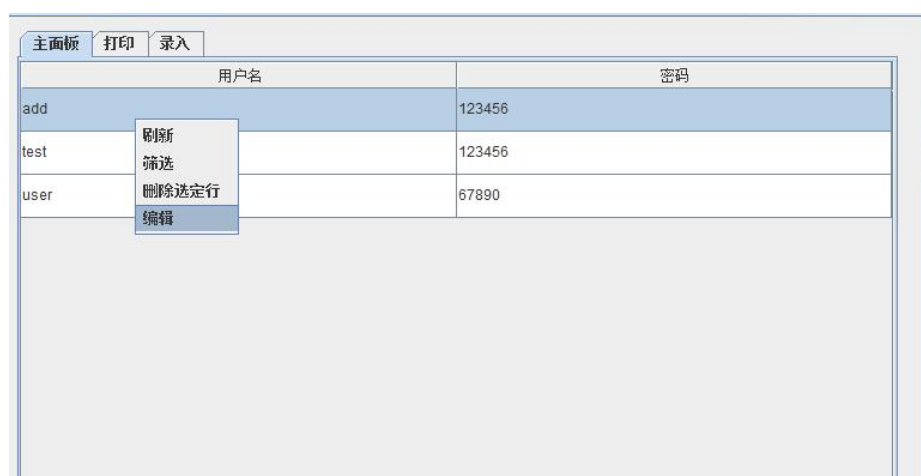
The screenshot shows a software interface with a main panel. At the top, there are three tabs: '主面板' (Main Panel), '打印' (Print), and '录入' (Input). The '主面板' tab is selected. Below the tabs is a table with two columns: '用户名' (Username) and '密码' (Password). The table contains three rows of data: 'add' with password '123456', 'test' with password '123456', and 'user' with password '67890'. Below the table is a large empty rectangular area.

用户名	密码
add	123456
test	123456
user	67890

图 4.9.4 删除本行信息及自动刷新后主面板展示

#### (5) 编辑功能

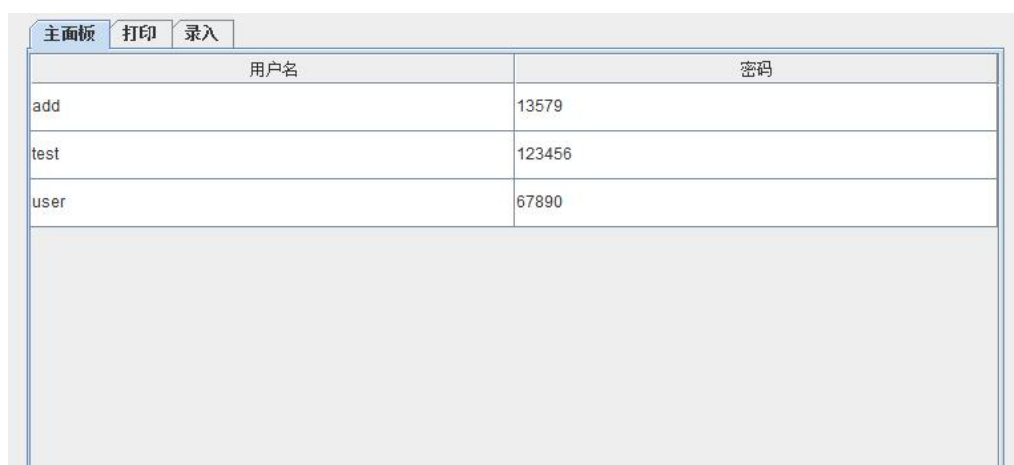
点击编辑键后自动跳转到编辑界面可以修改油卡信息，完成后点击确定实现信息更新并自动调回主面板，如图 4.9.5 所示。



The screenshot shows the '编辑' (Edit) form. At the top, there are four tabs: '主面板' (Main Panel), '打印' (Print), '录入' (Input), and '编辑' (Edit). The '编辑' tab is selected. The form is titled '管理员档案录入' (Administrator Record Entry). It contains two input fields: '姓名:' (Name) with the value 'add' and '密码:' (Password) with the value '123456'. Below these fields is a '确定' (Confirm) button.



The image shows a software window titled '管理员档案录入' (Administrator Record Entry). At the top, there are four tabs: '主面板' (Main Panel), '打印' (Print), '录入' (Entry), and '编辑' (Edit), with '编辑' being the active tab. The main area contains two input fields: '姓名:' (Name) with the value 'add' and '密码:' (Password) with the value '13579'. Below these fields is a blue button labeled '确定' (Confirm).



The image shows the '主面板' (Main Panel) tab of the software window. It displays a table with two columns: '用户名' (Username) and '密码' (Password). The table contains three rows of data.

用户名	密码
add	13579
test	123456
user	67890

图 4.9.5 编辑功能的过程及主面板展示

#### (6) 筛选功能

点击筛选功能后弹出对话框，让超级管理者输入筛选条件，再点击确定后按条件筛选，完毕后显示在主面板，如图 4.9.6 所示。

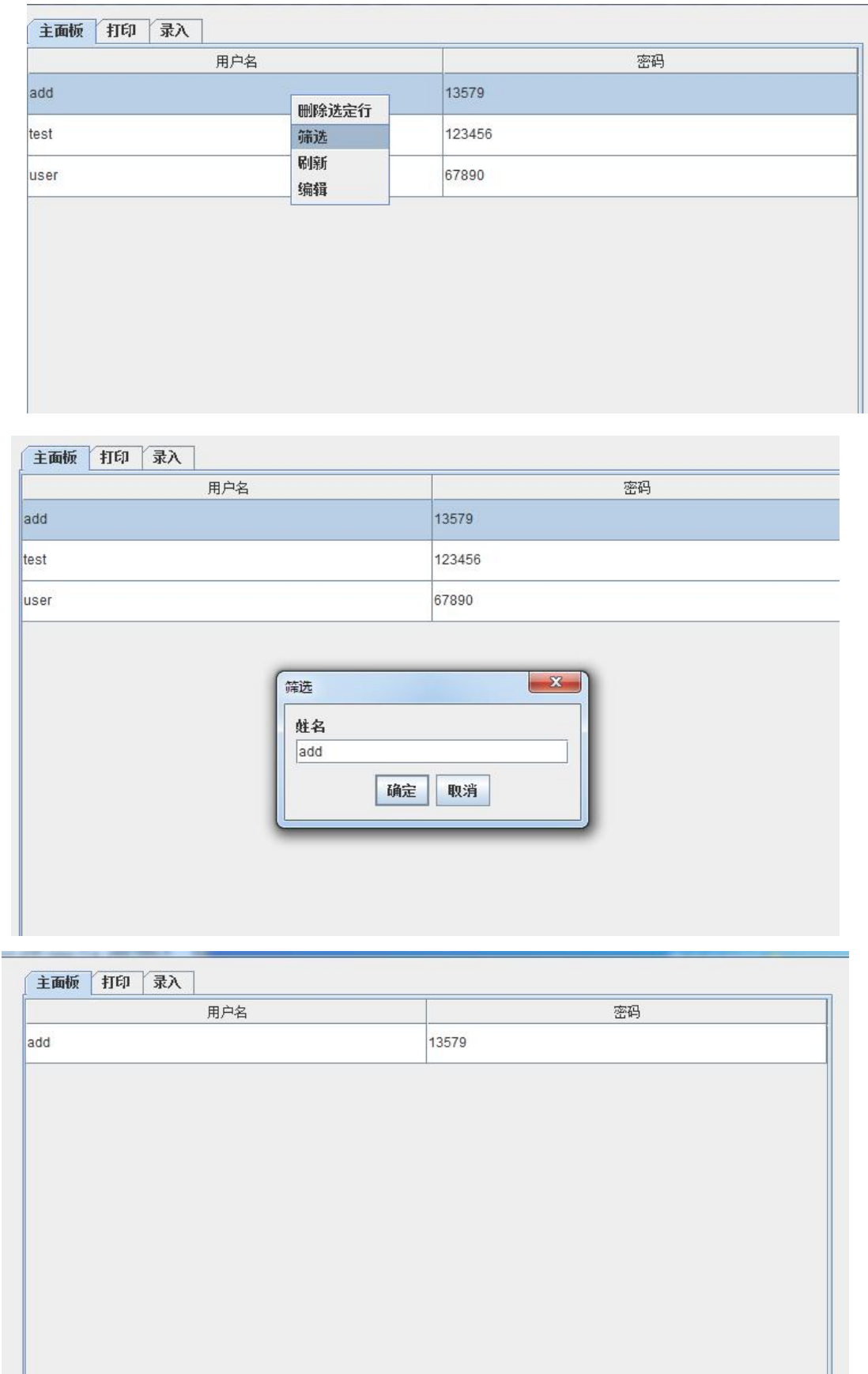


图 4.9.6 筛选功能及主面板展示

## （7）刷新功能

筛选完成后点击刷新按钮完成主面板刷新，如图 4.9.7 所示。

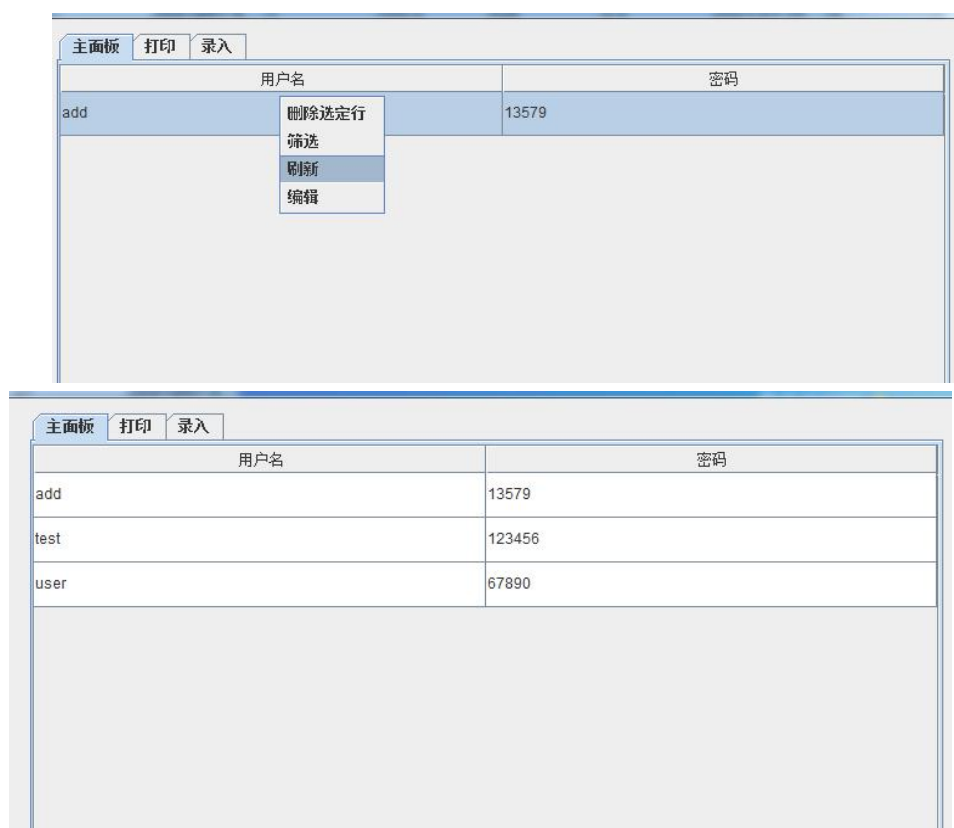


图 4.9.7 刷新功能及主面板展示

## （8）该模块涉及到的文件和类

通过详细的分析，该模块涉及到的文件和类在工程中的组织如图 4.9.8 所示。

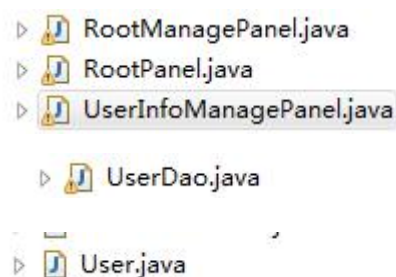


图 4.9.8 处理普通管理员信息模块组织

## 4.9.2 持久层

使用数据库 mysql 中表 User 持久存储用户信息，所有用户的信息均以这样的格式存储，且每个用户的信息在文件中占一行。

为了方便，在类 DBManager 中使用静态常量描述了连接数据库的详细方法。

```
String url =
"jdbc:mysql://localhost:3306/fleetManagement?serverTimezone=UTC&characterEnc
oding=utf8&useUnicode=true&useSSL=false"; // 数据库连接字符串，一般使用统一资
源定位符（url）的形式
```

```
String user = "root"; // 连接数据库时的用户
String password = "123456"; // 连接数据库时的密码
System.out.println("user="+user+" password="+password);
con = DriverManager.getConnection(url, user, password);
```

### 4.9.3 实体类 User

该类主要用于封装用户的信息：姓名和用户类型等信息，类图如图 4.9.9 所示。

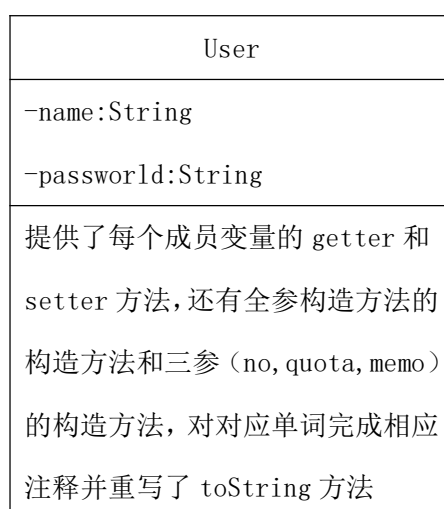


图 4.9.9 User 类图

### 4.9.4 操作层 Dao 层

该层（UserDao）主要完成对数据库中 User 的相关操作。

#### (1) 获取密码的方法

```
public static String queryUserByName(String user) ;
```

该方法通过封装 user 实现登录过程账号密码匹配。

#### (2) 获取全部信息的方法

```
public static List<String> get();
```

该方法实现对数据库中 User 全部信息的获取，返回 String 类型的集合，用于超级

管理员对其操作。

(3) 获取某行用户用全部信息的方法

```
public static List<User> getAll();
```

返回 User 类型的集合。

(4) 更新信息（编辑）的方法

```
public static void remove(User user, User info);
```

定位某行信息对其进行更新。

(5) 删除某行信息的方法

```
public static void delete(User de);
```

定位某行对该行信息进行筛选。

(6) 录入（增加）的方法

```
public static void save(User user);
```

该方法通过封装 User 实现用户增加。

#### 4.9.5 界面层（窗体层）

本层实现了对窗体内位置的布局和功能按钮的实现。

(1) 对窗体顶端布局窗体和核心代码

设置了对应信息的位置，如图 4.9.10 所示。



图 4.9.10 顶部布局展示

核心代码如下：

//表格

```
JPanel qP = new JPanel(new BorderLayout(10, 0));
```

```
JPanel aP = new JPanel(null);
```

```
JPanel eP = new JPanel(null);
```

```
JPanel pP = new JPanel(null);
```



```
Map<String, Component> panelsMap = new HashMap<>();  
panelsMap.put("主面板", qP);  
panelsMap.put("录入", aP);  
panelsMap.put("打印", pP);  
  
JTabbedPane jTabbedPane = ExtTable.buildJTabbedPane(panelsMap);  
panel.add(jTabbedPane, BorderLayout.CENTER);
```

```
String[] columns={"用户名", "密码"};  
DefaultTableModel tableModel = new DefaultTableModel(columns, 0);
```

(2) 对于点击对窗体内部信息进行相应的窗体和核心代码

使用了鼠标监听器，对列表中信息是用 HashMap，对右键点击显示进行响应，如图 4.5.11 所示。



图 4.5.11 右键点击所示

核心代码如下：

```
HashMap<JMenuItem, Function> mMap = new HashMap<>();  
JMenuItem dMitem = new JMenuItem("删除选定行");  
JMenuItem qMitem = new JMenuItem("筛选");  
JMenuItem rMitem = new JMenuItem("刷新");
```

```
JMenuItem eMitem = new JMenuItem("编辑");

//鼠标监听器

MouseListener mouseInputListener =
ExtTable.buildMouseListener(mMap);

table.addMouseListener(mouseInputListener);
```

## 4.10 核心工具类的说明

### 4.10.1 CalendarPanel 类

该类用于实现系统内关于日期的设置，实例如图 4.10.1 所示。



图 4.10.1 日期设置示意图

一些核心内容如下：

```
/**
 * 带参数的构造函数，该构造函数将构造一个设置了日历控件的载体组件的
CalendarPanel 对象实例
 *
 * @param component 日历控件的载体组件，可以是 TextField、JTextField、
Label、JLabel
 * @param pattern 日期格式，默认为 yyyy-MM-dd
 */
public CalendarPanel(Object component, String pattern)
```

/\*\*

\* 带参数的构造函数，该构造函数将构造一个 CalenderPanel 对象实例，该对象时候没有设置日历控件的载体组件

\* \* @param patern 日期格式，默认为 yyyy-MM-dd

\*/

public CalendarPanel(String patern)

/\*\*

\* 无参数的构造函数，该构造函数将构造一个 CalenderPanel 对象实例，该对象时候没有设置日历控件的载体组件

\*/

public CalendarPanel()

/\*\*

\* 获取日历控件的载体组件

\*

\* @return 返回该日历控件中的载体组件，如果没有设置，则返回 null;

\*/

public Object getComponent()

/\*\*

\* 设置日历控件的载体组件

\*

\* @param component

\*/

public void setComponent(Object component)

```
/**
 * 获取选择日期时候的日期字符串
 *
 * @return
 */
public String getReturnDateStr()
```

```
/**
 * 获取日历控件的回调接口对象
 *
 * @return
 */
public Callback getCallback()
```

```
/**
 * 设置日历控件的回调接口对象
 *
 * @param callBack
 */
public void setCallback(Callback callBack)
```

```
/**
 * 创建下拉选择日历控件的主体面板(即显示日期部分的面板)
 *
 * @param date 需要显示的日期
 */
private void createCalendarPanel(String date)
```

```
/**
 * 初始化所有控件
 */
private void init()

/**
 * 重新设置日历面板显示为系统当前日期
 */
public void reset()

/**
 * 改变当前日历控件面板的日期显示
 *
 * @param YEAR_OR_MONTH 是对 YEAR 还是 MONTH 操作，这里需要传入 Calendar
类对应的常量值：Calendar.YEAR, Calendar.MONTH
 * @param flag 对 YEAR_OR_MONTH 属性是进行加法操作还是减法操
作，如果 flag 为 true 则是加法操作，否则为减法操作
 */
public void changTheDate(int YEAR_OR_MONTH, boolean flag)

/**
 * 回调接口
 *
 * @author Administrator
 */
interface CallBack
```

#### 4.10.2 ExtTable 类

该类用于实现系统内关于窗体布局设置。

部分核心内容如下：

```
public static JTabbedPane buildJTabbedPane(Map<String, Component> panelsMap)
{
    // 选项卡面板
    JTabbedPane tabbedPane = new JTabbedPane();
    // 通过 BorderFactory 来设置边框的特性
    tabbedPane.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));
    panelsMap.forEach((s, jPanel) -> tabbedPane.add(s, jPanel));
    return tabbedPane;
}

public static JTable getDefaultTable(List<Integer> editCols) {
    JTable table = new JTable() {
        @Override
        public void updateUI() {
            // 刷新
            super.updateUI();
            // 表格行高
            setRowHeight(36);
        }

        @Override
        public boolean isCellEditable(int row, int column) {
            if (Objects.nonNull(editCols) && editCols.size() > 0 &&
editCols.contains(column)) {
                return true;
            }
        }
    }
}
```

```
        return false;
    }
};

// 设置表头不可移动
table.getTableHeader().setReorderingAllowed(false);
// 一次只能选择一项
table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
return table;
}

public static DefaultTableModel buildTableModel(Object[] columnNames,
Object[][] rowData) {
    return new DefaultTableModel(rowData, columnNames);
}
```

## 5、结论

通过此次课程设计，使我们更加扎实的掌握了有关 Java 编程方面的知识，在设计过程中虽然遇到了一些问题，但经过一次又一次的思考，一遍又一遍的检查终于找出了原因所在，也暴露出了前期我在这方面的知识欠缺和经验不足。实践出真知，通过亲自动手制作，使我们掌握的知识不再是纸上谈兵。过而能改，善莫大焉。在课程设计过程中，我们不断发现错误，不断改正，不断领悟，不断获取。最终的检测调试环节，本身就是在践行“过而能改，善莫大焉”的知行观。这次课程设计终于顺利完成了，在设计中遇到了很多问题，最后在老师的指导下，终于游逆而解。在今后社会的发展和学习实践过程中，一定要不懈努力，不能遇到问题就想到要退缩，一定要不厌其烦的发现问题所在，然后一一进行解决，只有这样，才能成功的做成想做的事，才能在今后的道路上劈荆斩棘，而不是知难而退，那样永远不可能收获成功，收获喜悦，也永远不可能得到社会及他人对你的认可！课程设计诚然是一门专业课，给我很多专业知识以及专业技能上的提升，同时又是一门讲道课，一门辩思课，给了我许多道，给了我很多思，给了

我莫大的空间。同时，设计让我感触很深。使我对抽象的理论有了具体的认识。通过这次课程设计，我熟悉了 Java 编程的各种方法以及如何提高程序的性能等等。我认为，在这学期的实验中，不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。更重要的是，在实验课上，我们学会了很多学习的方法。而这是日后最实用的，真的是受益匪浅。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。这对于我们的将来也有很大的帮助。以后，不管有多苦，我想我们都能变苦为乐，找寻有趣的事情，发现其中珍贵的事情。就像中国提倡的艰苦奋斗一样，我们都可以在实验结束之后变的更加成熟，会面对需要面对的事情。 回顾起此课程设计，至今我仍感慨颇多，从理论到实践，在这段日子里，可以说得是苦多于甜，但是可以学到很多很多东西，同时不仅可以巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在设计的过程中遇到问题，可以说得是困难重重，但可喜的是最终都得到了解决。 实验过程中，也对团队精神的进行了考察，让我们在合作起来更加默契，在成功后一起体会喜悦的心情。果然是团结就是力量，只有互相之间默契融洽的配合才能换来最终完美的结果。

## 心得体会

这次系统是使用 eclipse 开发，数据持久层是数据库存储。这个系统主要是我们自己开发的，但也得到了老师和同学的很大帮助。我正在做系统的过程中遇到了很多问题，有的是知识存储不足，有的是考虑不够周全，之所以能够顺利实现基本功能，离不开老师和同学的大力相助。事实证明，只靠自己是不能顺利完成一套功能完整的系统的，必须充分利用团队的力量。开发一套系统，最重要的是细心，并不是一定要做到面面俱到，但也要充分考虑到客户的需求和现实意义，不管什么系统，只用运用到实际应用中，才具有现实意义。所以在准备工作中要正确分析社会需求了解现实应用，画出流程图，把大体框架做好，然后再逐一细化。我们不可能做到面面俱到，但一定要做到步步扎实，作为一个程序编程人员，要保持清醒的头脑，以现实为依据，让自己的每一行代码都能实现自己的意义。通过这次课程设计，我收获的不仅仅是课程上的知识得到实际应用，还有编程的基本习惯和开发系统时应注意的流程。



## 参考文献

- [1]李兴华 . Java 从入门到项目实战（全程视频版）. 湖北：水利水电出版社, 2019. 619
- [2]高洪岩 . Java 编程思想 . 北京：机械工业出版社，2014. 900
- [3]耿祥义 张跃平等. Java 面向对象程序设计. 第三版. 北京：清华大学出版社，2010. 803
- [4]刘爽英等. 数据库原理及应用. 北京. 清华大学出版社, 2013. 447
- [5]爪哇®平台、标准版和 Java 开发工具包第 15 版 API 规范  
<https://docs.oracle.com/en/java/javase/15/docs/api/index.html>