**快递物流管理系统（LEMS）**

**课程设计报告**

**班级：计科一班**

**学号：20140710**

**姓名：刘立东**

# 引言

## 目的

该文档是关于快递物流管理系统(LEMS)的功能和性能的描述，重点描述了系统的功能需求，并作为系统设计阶段的主要输入。

本文档的预期读者是:需求分析人员;设计人员；开发人员；项目管理：人员；测试人员；用户。

## 项目背景

项目名称：快递物流管理系统(LEMS)

项目提出者：南师大计算机学院

开发单位：南师大计算机学院

用户：快递物流管理人员

项目实施单位：南师大计算机学院

与其他系统的关系：本系统独立运行

## 缩写说明

1. 南师大：南京师范大学
2. LEMS：Logistics and Express Management System

## 术语定义

## 参考资料

[1] 窦万峰，软件工程方法与实践，北京：机械工业出版社，2009

[2] 贾志城 王云，JSP程序设计慕课版，人民邮电出版社，2016

[3] 明日科技，Java Web：典型模块与项目实战大全，电子工业出版社，2012

[4] 吉根林 顾韵华，Web程序设计，电子工业出版社，2015

## 版本信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 1 | 2017-6-10 | 1.0 | 全部 | 完成第一次编写 |

# 任务概述

## 系统定义

### 项目来源及背景

该项目是南师大计算机学院提出的，要求开发一个快递物流管理系统，供物流管理人员使用。

### 项目要达到的目标

该软件是为了更加方便和科学地管理快递物流而研发的，目标就是提高物流管理的效率和可靠性。

### 系统整体结构

图A-1给出了系统的物理组成结构。



## 运行环境

### 设备环境

普通PC机：处理器4核以上，内存2GB以上。

### 硬件环境

对计算机配置要求不高，能正常运行该程序即可。

### 软件环境

该软件适用于目前主流的操作系统，支持Win7、Win10两种系统。

### 网络环境

因特网，物流公司局域网

### 操作环境

计算机桌面操作

### 应用环境

系统工作流程如图A-2所示.



### 条件限制

### 软件的开发工作的假定和约束

该项目要求在Eclipse JavaEE上开发，需要Java jdk1.8、Tomcat 8.5和MySQL 5.1支持。

### 软件的最终用户、用户的教育水平和技术专长

最终用户一般是没有学历要求的物流管理人员。细心，熟悉计算机操作者为佳。

### 软件的预期使用频度等

本软件用于物流管理，使用频率较高。

# 数据描述

## 静态数据

1. 车辆信息
2. 管理员信息

## 动态数据

1. 用户登录信息
2. 车辆状态

## 数据库描述

本软件使用MySQL 5.1数据库。数据关系E-R图如下。

1. admin表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键 | 备注 |
| admin\_id | int | 是 | 管理员ID |
| admin\_name | varchar(255) | 否 | 管理员姓名 |
| admin\_password | varchar(255) | 否 | 密码 |

1. car表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键 | 备注 |
| car\_id | int | 是 | 车辆编号 |
| car\_name | varchar(255) | 否 | 车主姓名 |
| car\_phone | varchar(255) | 否 | 车主手机 |
| car\_free | boolean | 否 | 车辆空闲标记 |

1. orders表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否主键 | 备注 |
| orders\_id | int | 是 | 订单编号 |
| orders\_sname | varchar(255) | 否 | 发件人姓名 |
| orders\_sphone | varchar(255) | 否 | 发件人手机 |
| orders\_saddress | varchar(255) | 否 | 发件人地址 |
| orders\_rname | varchar(255) | 否 | 收件人姓名 |
| orders\_rphone | varchar(255) | 否 | 收件人手机 |
| orders\_raddress | varchar(255) | 否 | 收件人地址 |
| orders\_process | boolean | 否 | 订单状态 |
| admin\_id | int | 外键 | 管理员ID |
| car\_id | int | 外键 | 车辆ID |

## 数据字典

数据字典描述系统实体关系图中的实体和关系，以及数据流中的过程、数据流、数据存储的部分内容。



## 数据采集

系统启动时，需要读取数据库中用户的个人信息(admin表)，订单信息(order表)，车辆信息(car表)等。

# 功能需求

## 功能划分

### 系统功能组成



### 功能编号和优先级

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 描述 | 主要发起者 |
| 1 | 用户登陆注册 | 重要 | 用户登陆 | 管理员 |
| 2 | 订单生成和修改 | 重要 | 订单管理 | 管理员 |
| 3 | 车辆增加和修改 | 重要 | 车辆管理 | 管理员 |

### 功能定义

1. 用户注册
2. 用户登陆
3. 订单生成
4. 订单修改
5. 订单确认
6. 车辆增加
7. 车辆修改

## 功能描述

### 功能说明

1. 用户登陆注册

管理员填写用户名和密码注册账号，系统自动分配给管理员一个ID

1. 订单生成和修改
2. 管理员单击增加订单按钮就会弹出一个快递单表格，需要用户填写收件人姓名，收件人手机，收件人地址，发件人姓名，发件人手机，发件人地址以及系统给该订单分配的车辆编号，车辆状态变为在途。
3. 管理员修改订单：修改订单界面会列出在数据库中的所有订单，每个订单后面都有两个按钮，修改和确认订单，单击修改按钮，能够修改订单的所有信息。
4. 管理员确认订单：管理员确认收货后，订单状态变为已完成，车辆状态变为空闲。
5. 车辆增加和修改
6. 管理员增加车辆信息

单击增加车辆信息，弹出表格，管理员填入车辆编号，司机姓名，司机手机，车辆状态。默认为空闲。

1. 管理员修改车辆信息
2. 管理员可以修改车辆编号，司机姓名，司机手机，车辆状态。

### 详细描述

1. 用户登陆注册



1. 订单生成和修改



1. 车辆增加和修改



# 性能需求

## 数据精确度

1. 时钟设置以秒为单位；
2. 系统计时误差在10秒以内；

## 时间特性

用户登陆状态不超过12小时；

订单管理提交响应时间不超过5秒；

## 适应性

1. 适应Windows操作系统的不同版本；
2. 支持不同类型的PC兼容机和笔记本电脑；

# 运行需求

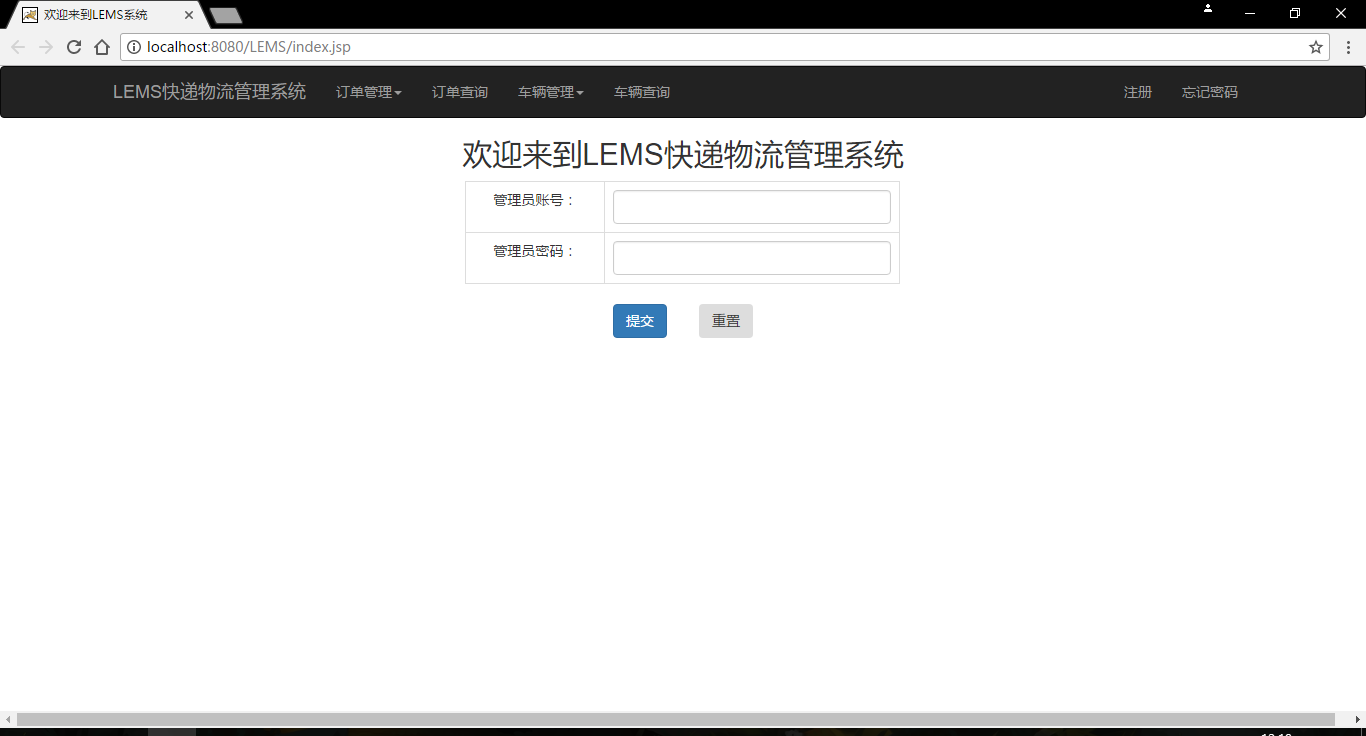
## 用户界面

### 界面风格

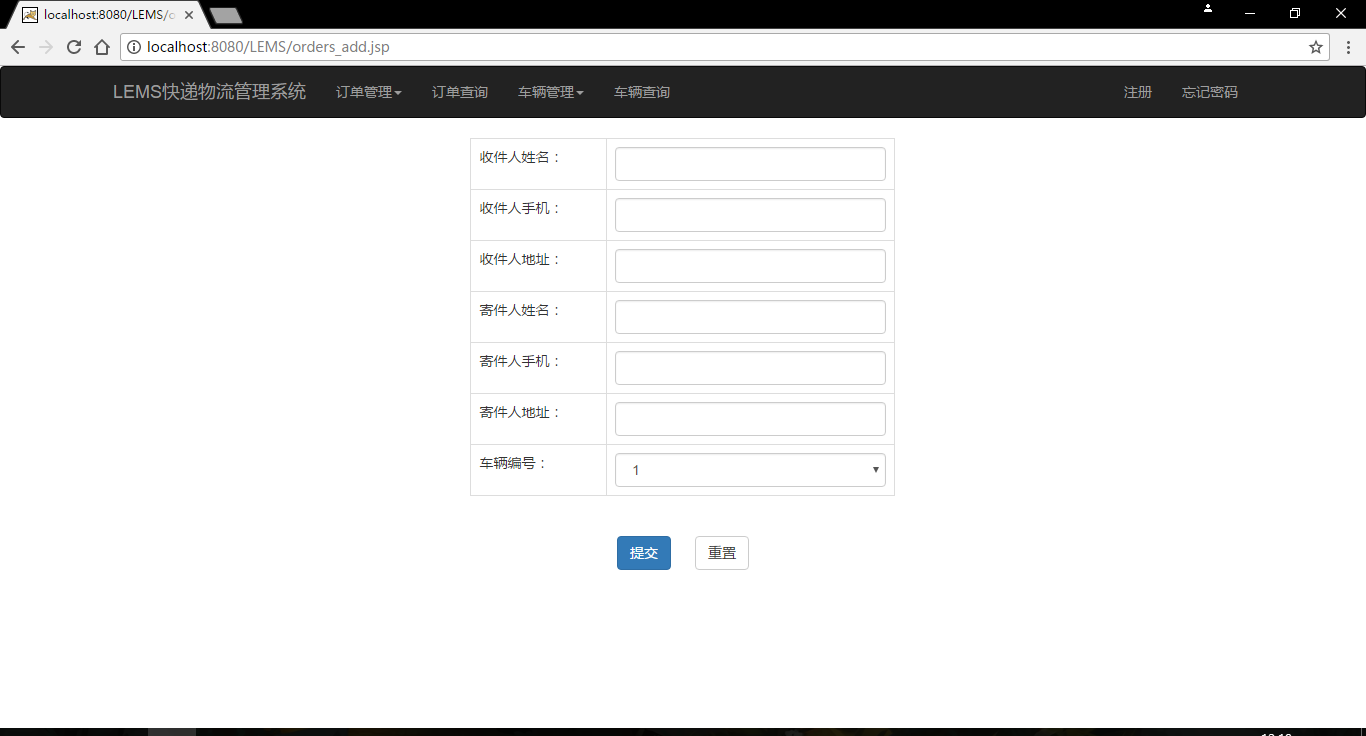
遵守Windows风格

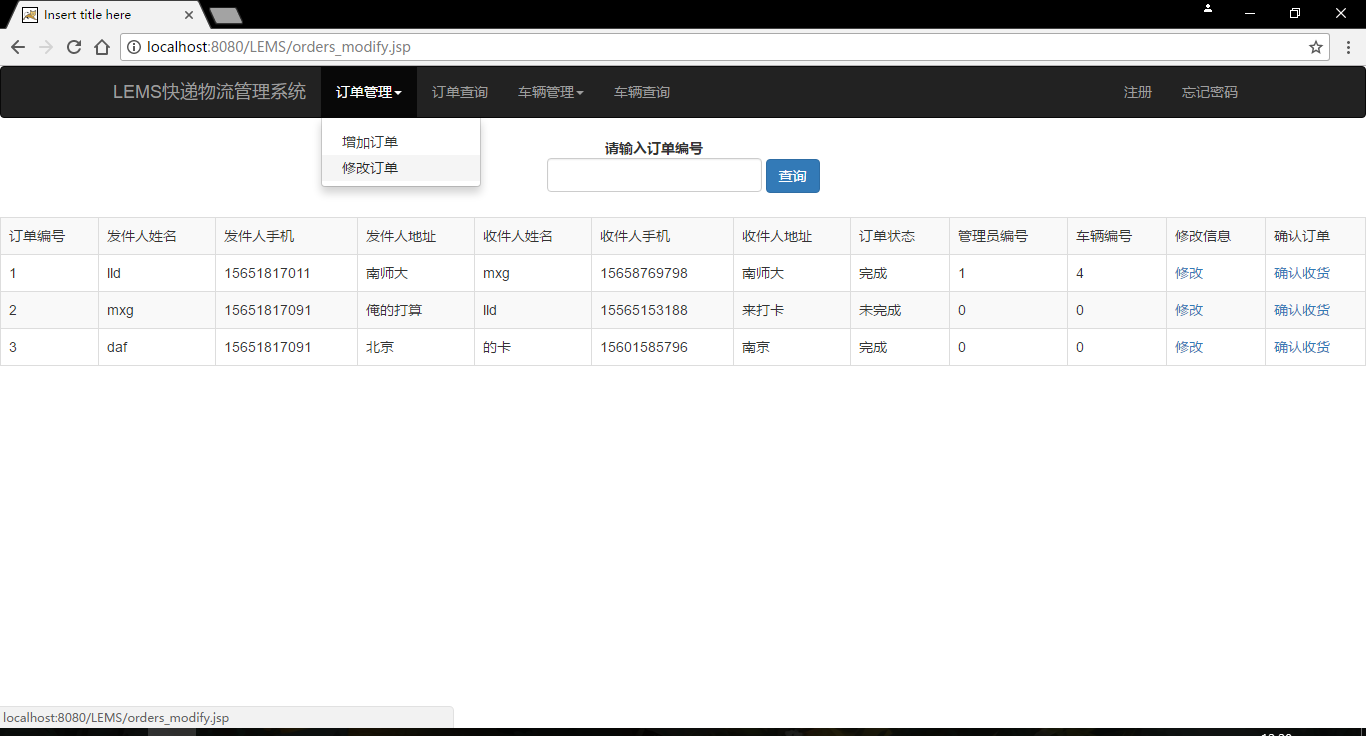
### 界面描述和样式

1. 登陆界面

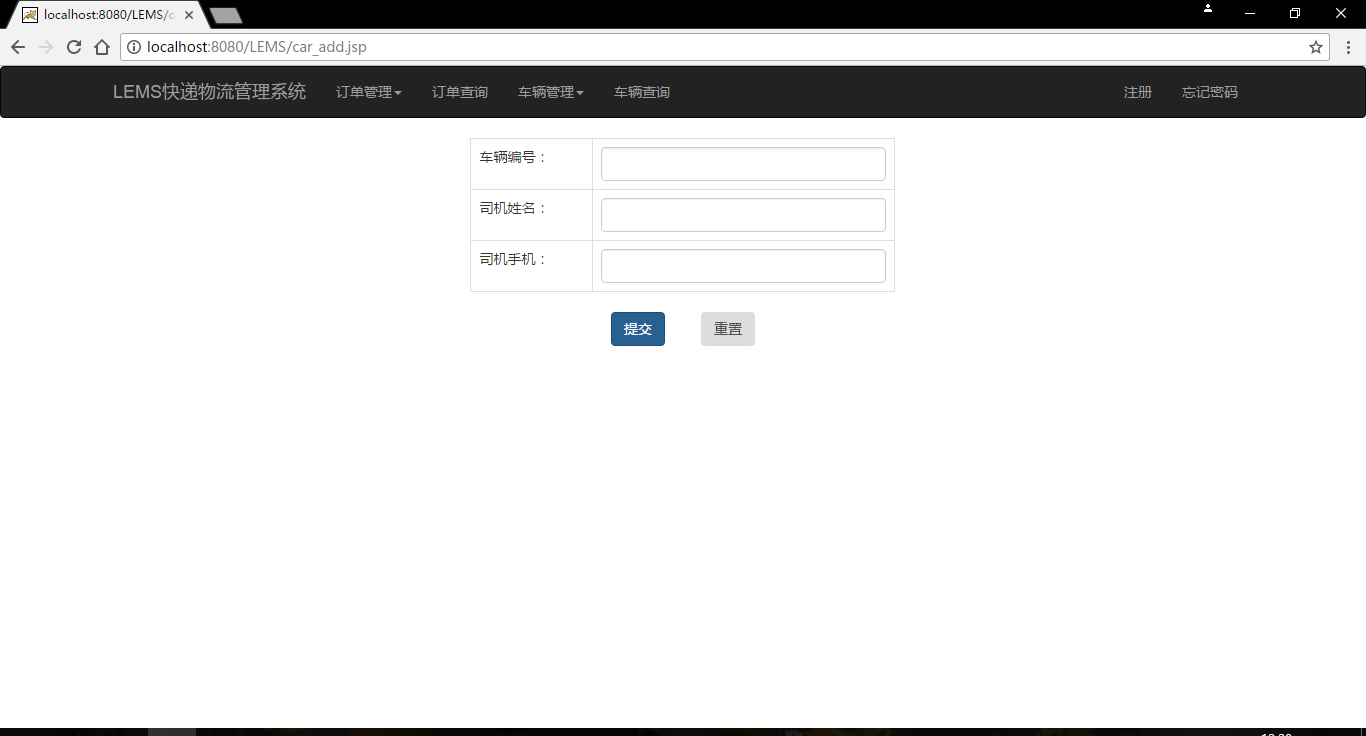


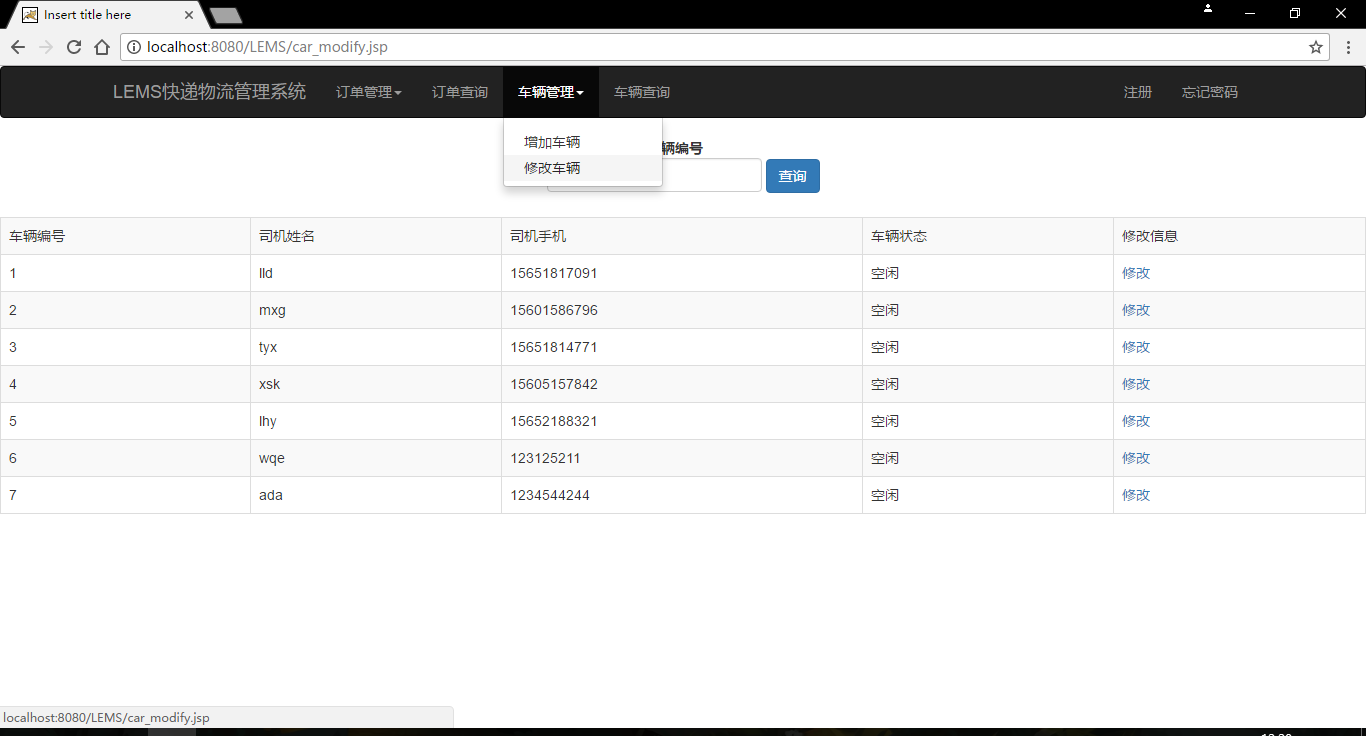
1. 订单管理界面





1. 车辆管理界面





## 硬件接口

鼠标、键盘必须是USB接口；

## 软件接口

需要Java JDK 1.8 、Tomcat 8.5和MySQL 5.1支持

## 故障处理

重启系统或重启数据库可恢复故障。

# 其他需求

## 检测或验收标准

支持用户信息管理，订单管理，车辆管理等功能，且出错率低于5%；

## 可用性、可维护性、可靠性、可转换性、可移植性要求

软件故障率低于5%；

软件要求模块设计，设备驱动可以更换；

软件可以在不同的Windows平台使用；

软件支持不同的PC兼容机。

## 安全保密性要求

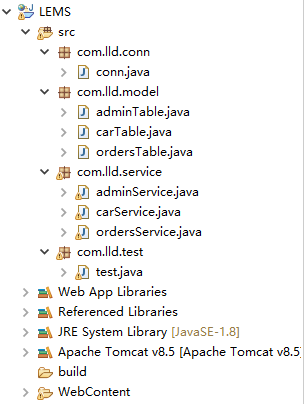
数据库中用户的账号密码，订单信息保密。

## 开发要求

要求在Eclipse JavaEE Oxygen 上开发；

需要Java JDK 1.8 、Tomcat 8.5和MySQL 5.1支持

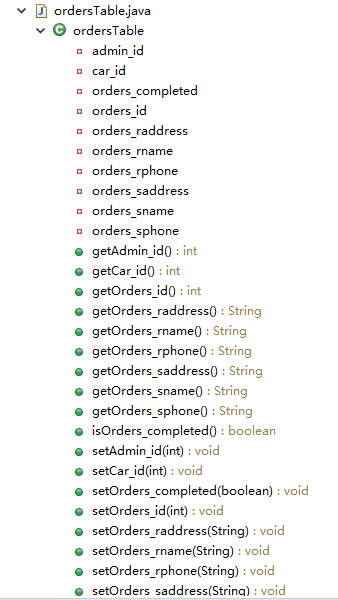
1. **接口设计**
   1. **外部接口设计**
      1. **外部数据接口**
      2. **外部系统或设备接口**
   2. **内部接口设计规约**
      1. **内部模块接口调用关系**

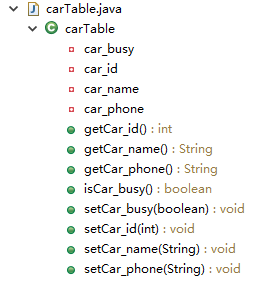


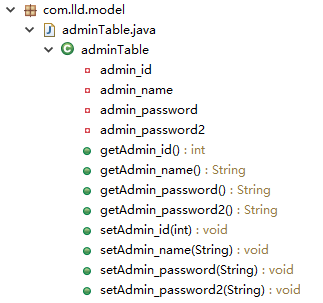
* + 1. **接口数据结构**

本程序采用MVC结构，以下为各个包的结构

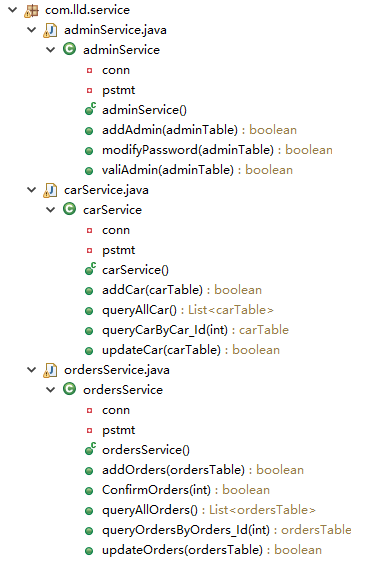
1. Model
2. Admin 管理员实体
3. Car 车辆实体
4. Orders 订单实体







二．Controler



、(1)adminService

注册管理员 addAdmin();

修改密码 modifyPassword();

验证登陆 valiAdmin();

(2)carService

添加车辆 addCar();

查询所有车辆 queryAllCar();

通过车辆编号查询车辆 queryOneCarByCar\_id();

修改车辆 updateCar();

(3)ordersService

添加订单 addOrders();

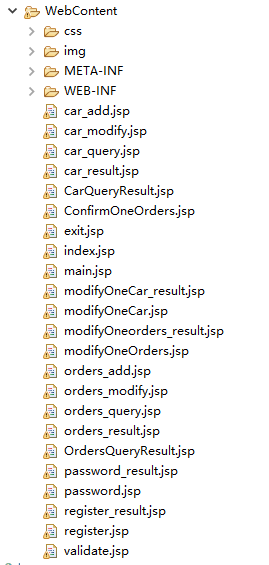
确认订单 confirmOrders();

查询所有订单 queryAllOrders();

通过订单编号查询订单 queryOneOrdersByOrders\_id();

修改订单 modifyOrders();

三．View



## 关键部分代码

### 连接数据库

**public** Connection getCon() {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost/lems?useUnicode=true&characterEncoding=utf-8";

String user = "root";

String password = "123456";

Connection conn = DriverManager.*getConnection*(url, user, password);

System.***out***.println(conn.getMetaData().getURL());

**return** conn;

} **catch** (Exception e) {

e.printStackTrace();

**return** **null**;

}

}

### 登陆验证

**public** **boolean** valiAdmin(adminTable admin) {

**try** {

pstmt = conn.prepareStatement("select \* from admin where admin\_name=? and admin\_password=?");

pstmt.setString(1, admin.getAdmin\_name());

pstmt.setString(2, admin.getAdmin\_password());

ResultSet rs = pstmt.executeQuery();

**if** (rs.next()) {

**return** **true**;

} **else** {

**return** **false**;

}

} **catch** (SQLException e) {

e.printStackTrace();

}

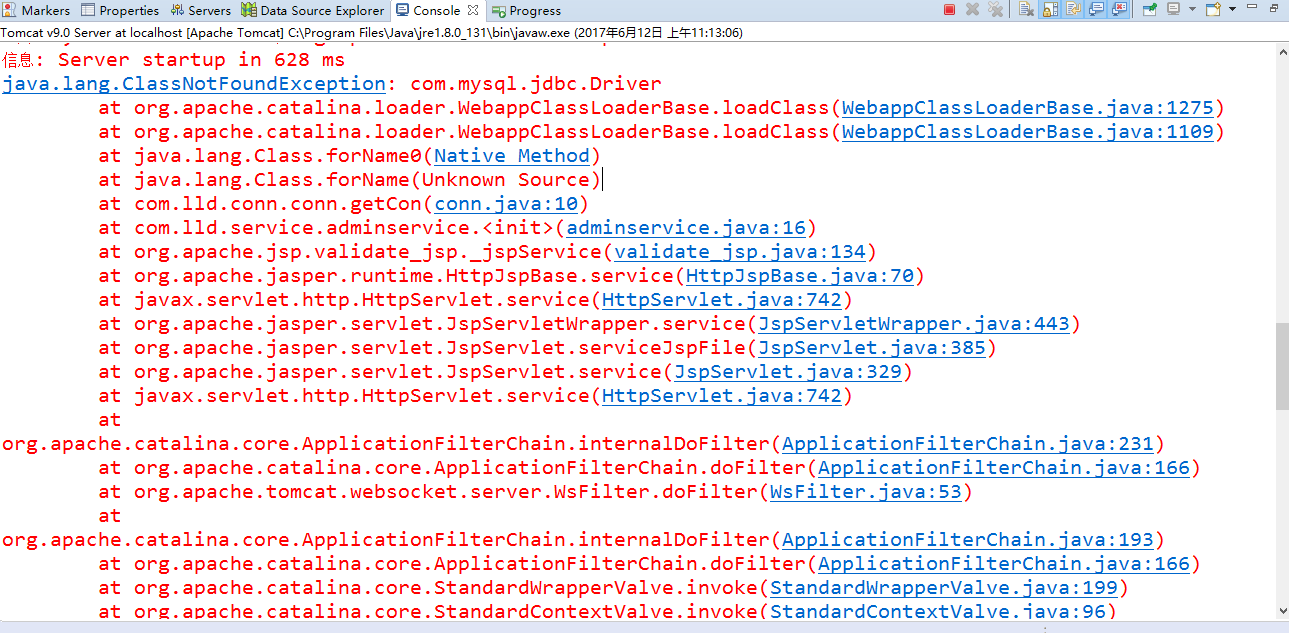
**return** **false**;

}

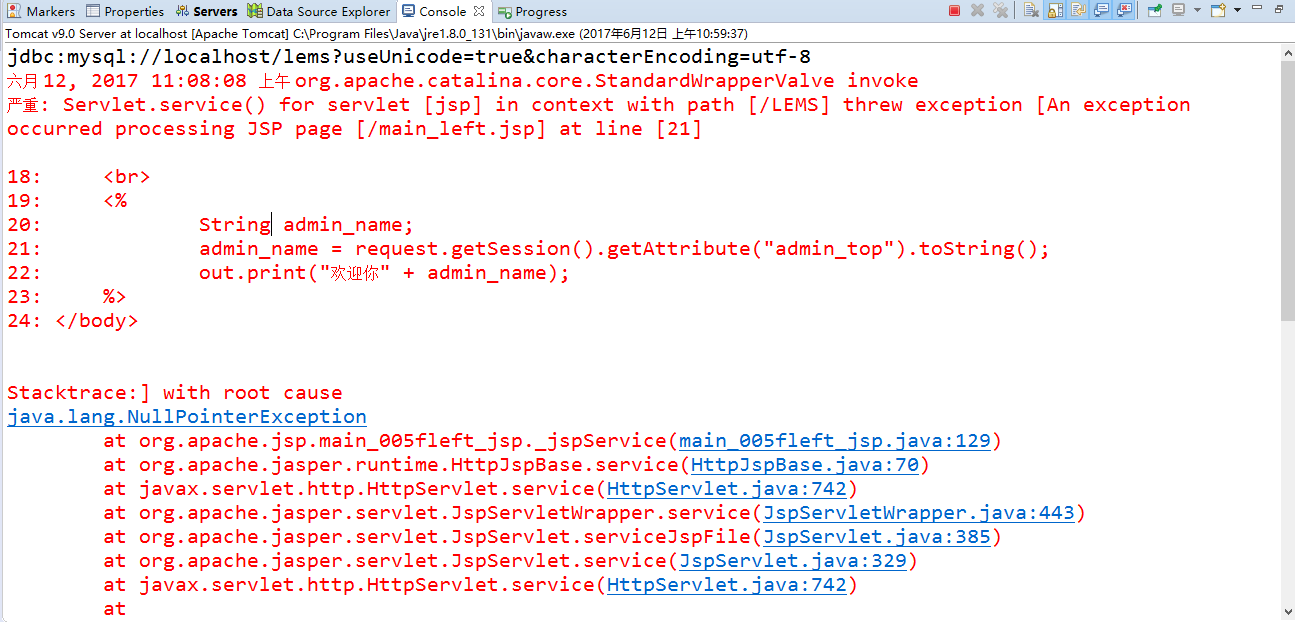
# 总结

## 错误总结

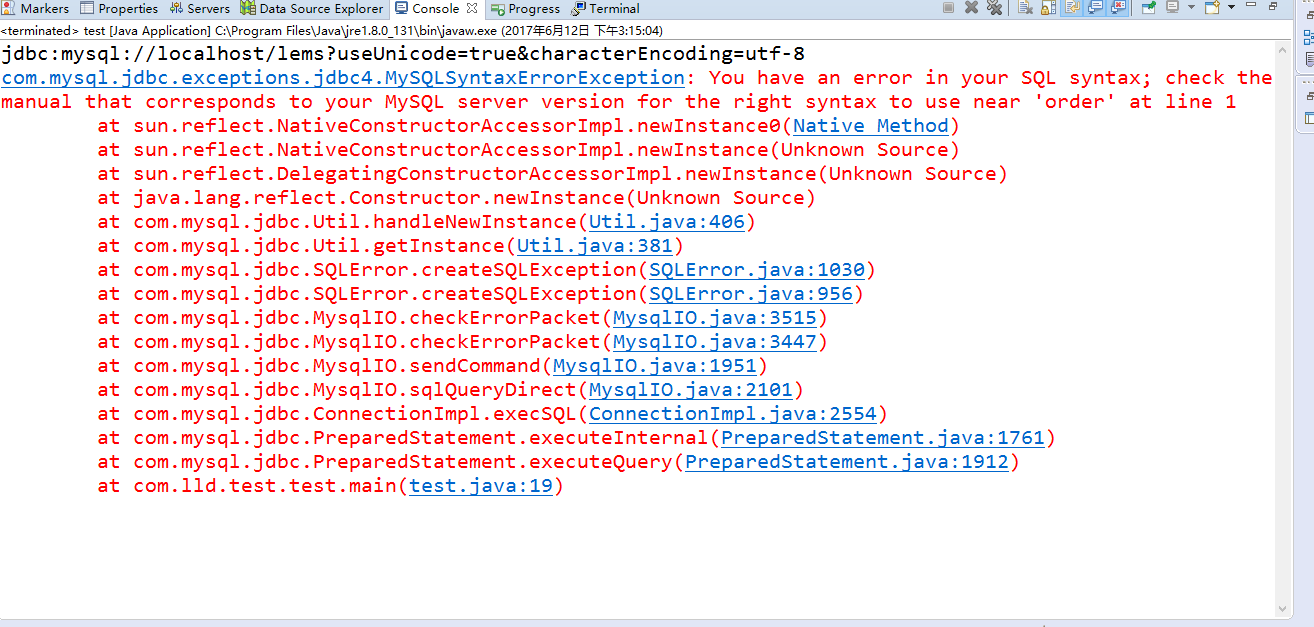
**错误一** 连接数据库的驱动包mysql-connector-java-5.1.7-bin.jar虽然添加进了项目，但是没有放入LEMS/WebContent/WEB-INF/lib/目录下仍然会报错。



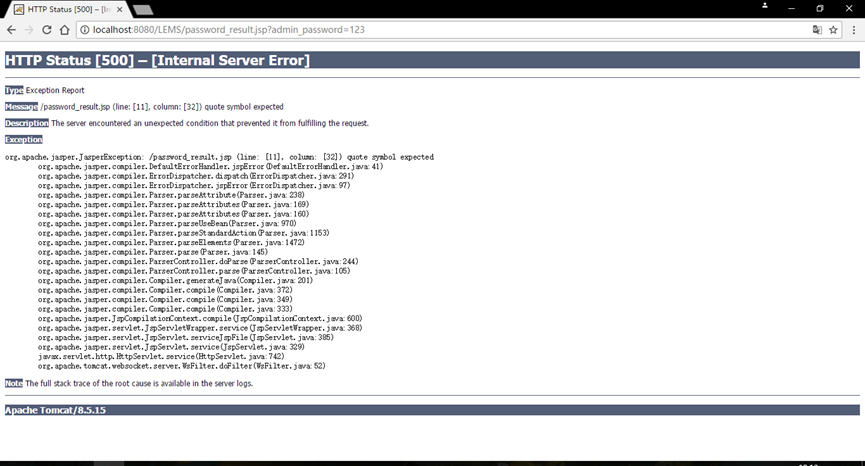
**错误二 修改了jsp页面中的javaBean之后再运行，需要重新部署项目，不然的话会继续显示之前的错误。在这个地方我浪费了很多时间，一个偶然的机会让我发现是这个原因**



错误三 数据库名order是mysql的关键字。在建立数据库之前，我并不知道order是mysql的关键字，综合百度搜索的结果，我才知道是我的数据库名和MySQL的关键字冲突了，所以，我将数据库名order该为orders。



错误四 使用Eclipse的自动提示功能时，添加javabean时，忘记在class里面加入双引号比如 class=”com.lld.service.carservice”，如果没加双引号就会报错。



## 实验总结

通过这次课程设计，我终于独立的完成了一次项目的开发，虽然是一个很简单的项目，但其中辛苦和趣味也许只有我自己一个能懂，从最开始用最简单最原始的页面的设计，到利用这几天时间去学习如何使用Bootstrap设计一个美观的页面，从连接数据库都连接不上到可以熟练地使用各种SQL语句来操控数据库，这几天，我真的学会了很多，也明白了只有自己亲自动手去操作才能真正地掌握一门技术。虽然程序还有很多不足，不过之后我还会继续完善。