

河北工业大学  
《软件设计与编程实践》  
——《物流运输平台设计与开发  
》技术报告

专业： 计算机科学与技术

班级： 计 191

组长： 王永航（195857）

成员： 刘景贤（195852）

王凡（195855）

王超（195840）

指导教师： 彭玉青

完成时间： 2021-12-4

# 目 录

项目开发计划书.....	(页码)
需求规格说明书.....	(页码)
设计规格说明书.....	(页码)
系统实现及测试报告.....	(页码)
实验工作记录和总结.....	(页码)

# 项目开发计划书

## 1. 概述

### 1.1 项目的目的与目标

随着物流行业的高速发展，物流从业人员数量急剧增加，货物和车辆信息急需通畅的平台予以发布。

项目目的：

本实验的目的是建设一个物流服务平台，本项目旨在开发一个功能实用、操作方便、简单明了的物流运输平台，实现一个可以随时发布物流消息，寻找车源消息，观看货物情况的交流平台，以提高货主和司机的信息交流效率，减少运输成本。

项目目标：

1. 项目意图实现一个物流运输平台。
2. 本系统支持不同权限用户的登录。
3. 货主可以进行注册，完善个人信息，发布货物信息，浏览发布的货物信息，查看运输状态，选择司机等功能。
4. 司机：进行注册，完善个人信息，浏览货物信息、查询货物信息，选择货物信息，修改运单状态等功能。
5. 管理员可以管理账号，管理货物，管理订单，登录系统，退出系统。
6. 项目代码具有一定的可扩充性，可以应对将来的更新需求。
7. 用户界面简洁方便，美观大方。

### 1.2 系统功能范围及过程规划

系统总体功能：物流运输平台共设有货主、司机、系统管理员三种身份，主要实现以下功能：

货主：

- (1) 进行注册，输入账号密码登录
- (2) 完善个人信息
- (3) 发布货物信息
- (4) 浏览发布的货物信息
- (5) 查看运输状态
- (6) 选择司机，确认订单
- (7) 支付订单
- (8) 退出系统

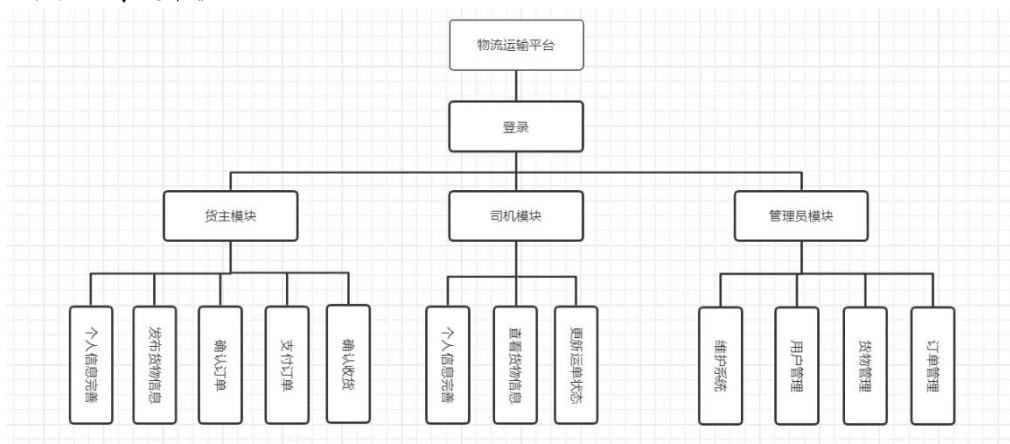
司机：

- (1) 进行注册。输入账号密码登录
- (2) 完善个人信息
- (3) 浏览货物信息、查看货物信息、筛选货物信息
- (4) 选择货物进行承运、更新运单状态
- (5) 退出系统

系统管理员：

- (1) 用户管理
- (2) 货物管理

- (3) 订单管理
- (4) 系统维护



## 2 可行性分析

### 2.1 社会可行性

本项目符合当前社会的发展需求，随着交通的便利，以及物品运输的需求大大增加，通过物流运输平台，物流行业人员可以随时发布或找到自己适合的货源信息、车源信息，便于货主随时发布货物信息，及时了解货物承运情况，提高货物运输速度，减少运输成本，有利于社会的经济发展。

本系统的设计将严格遵守国家法律，系统中包含大量用户的个人信息，本系统保证用户个人隐私仅用于本产品业务，严格保证用户信息安全。

### 2.2 经济可行性

估算物流运输平台的开发费用和今后的运行、维护费用，并将费用与效益进行比较。

平台的投资成本可以从开发成本和维护成本两方面考虑。开发成本涉及到开发人员的开发费用、服务器的租用费用以及域名申请费用；维护成本具有延续性，期望是可控的。

平台的效益可以从货主和司机两方面考虑。应重点保证货主和司机信息的正确性和安全性。广告收益、司机和货主注册收益是可观的。该产品所追求的市场是物流运输平台，它可以便于货主随时发布货物信息，及时了解货物承运情况，提高货物运输速度，减少运输成本。便于车主寻找货物信息，提高车辆利用率，具有前景。

## 2.3 技术可行性

考虑系统实施的可行性，现有的几种主流开发技术都较为成熟，可以满足不同的系统需求，基于现在先进的数据库技术以及稳定的服务器，可以保证系统功能的实现与运行。现行几大数据库如 MySQL、SQL Server 都具备数据安全保护功能，可保证系统的安全性。服务器如阿里云、腾讯云，可保证系统的稳定运行。在硬件方面，由于系统没有进行复杂的数据分析与计算，对于家用或商用电脑整体硬件的配置水平、移动设备的性能标准，系统在硬件方面是可行的。所以，无论在安全性、可用性、可靠性方面都毫无疑问，因此本系统具备技术可行性。

## 3. 方案选择

	方案一	方案二
拟用系统架构	MVC	C/S
辅助 CASE 工具	后端：springboot 框架、tomcat、Mysql，前端：vue，elementui	C#、SQL Server、
拟用技术	PowerDesigner 16.5	PowerDesigner 16.5
开发平台	Intelij IDEA 2021.2	Visual Studio 2019
功能约束	基于 MVC 架构的系统可以跨平台使用，因此在满足基本功能的前提下，可以突破更多的功能约束，满足不同场景的功能需求；此外软件更新时，客户端不用更新，只需在服务器端更新，具有较好的用户体验。	基于 C/S 架构的系统可以满足基本的功能需求，但相对于 MVC 架构的系统，灵活性没有那么高，略显笨拙，可移植性低，使用场景有限；此外软件更新时需要在客户端与服务器端同时更新，限制了用户体验。
技术约束	Web 相关开发技术发展到现在已较为成熟，搭配稳定的阿里云服务器和先进安全的 MySQL 数据库技术，能够保证系统的功能实现和正常运行。对于硬件配置要求较低，一般具备浏览器的设备就可	C#搭配 SQL Server 也可以保证系统基本功能的实现和正常运行，但是这种方案比较看重设备硬件配置，对操作系统也有一定的要求和限制。

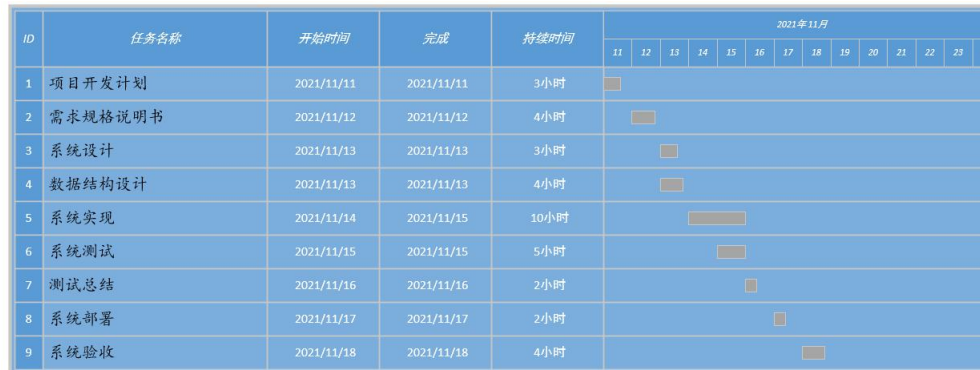
	以使用该系统。	
社会约束	用户可多场景多时段动态浏览物流的相关信息，不受空间时间限制，简单、易学	使用场景单一，可能不会被大多用户接受

方案一是典型的 Web 开发，MVC 架构模式具备很好的可移植性，可跨平台使用，在功能约束、技术约束等方面都比方案二更加适合物流管理平台，方案二的灵活性较低，不能满足用户的使用场景需求。综上所述，选择方案一作为本次系统开发的方案。

#### 4. 项目进度计划

任务关系时间表

任务编号	任务名称	所需时间 (h)	依赖
T1	项目开发计划	3	
T2	需求规格说明书	4	T1 (M1)
T3	系统设计	3	T2 (M2)
T4	数据结构设计	4	T2 (M2)
T5	系统实现	10	T3、T4 (M3)
T6	系统测试	5	T5 (M4)
T7	测试总结	2	T6 (M5)
T8	系统部署	2	T7 (M6)
T9	系统验收	4	T8 (M7)



甘特图

## 5 人员组织计划

人员组织计划表

任务编号	任务名称	所需时间 (h)	负责人	成果审查人
T1	项目开发计划	3	全体人员	王永航
T2	需求规格说明书	4	王超	王凡
T3	系统设计	3	王永航	刘景贤
T4	数据结构设计	4	王凡	王永航
T5	系统实现	10	全体人员	刘景贤
T6	系统测试	5	全体人员	王凡
T7	测试总结	2	王超	王超
T8	系统部署	2	刘景贤	王永航
T9	系统验收	4	全体人员	王超

# 需求规格说明书

## 1. 系统功能需求

### 1.1 用户需求描述

不同角色在使用系统时将会有不同的操作界面，实现不同的功能，拥有不同的执行权限。

1. 所有游客都可进入网站，查看网站的基本介绍与信息，了解网站基本功能，并能进行注册。
2. 货主与司机都可通过网站注册登录
3. 货主与司机可以修改密码
4. 货主与司机都可以完善、更改相关信息。
5. 货主方可以发布货物信息
6. 司机可以在网站浏览、查询、筛选所有货物信息
7. 司机可以选择货物与货主进行承运
8. 货主在有多个司机承运时有权选择其一进行承运
9. 司机在承运过程中，可以实时更新运单状态，承运完成后可修改为已完成
10. 货主可以通过网站查询自己的货物运单状态
11. 货主在选定司机时可以支付运费给平台
12. 货主在收到货物时确认收货，平台将运费转给司机
13. 管理员输入正确账号密码进行登录
14. 管理员可以管理货主与司机的账号信息
15. 管理员可以管理货物信息与运单信息

### 1.2 初始功能提取

表 2-1 功能需求点列表

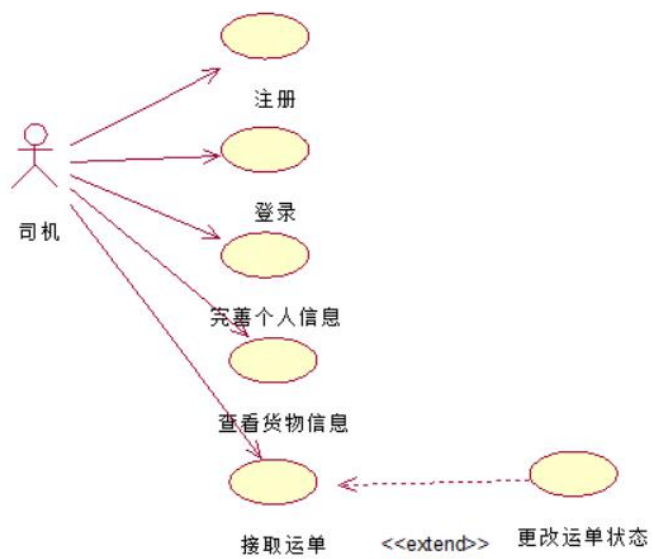
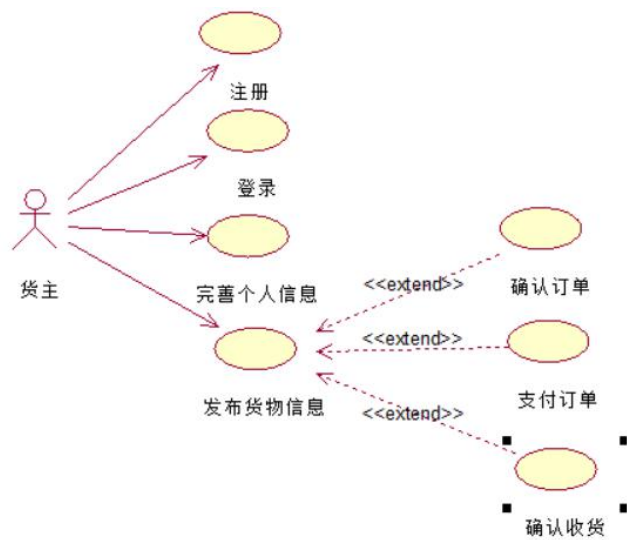
编 号	功 能 名 称	使 用 人	功 能 描 述	输 入 内 容	输 出 内 容
1	注册	货主或司机	选择身份并进行注册	用户点击“注册”按钮，填入所需信息进行注册	系统主界面
2	登录	货主、司机、系统管理员	正确输入账号密码进行登录	货主点击登录按钮，输入账号密码	系统主界面
3	完善个人信息	货主或司机	在登录后，完善个人的基本	用户点击“完善个人信息	用户信息表

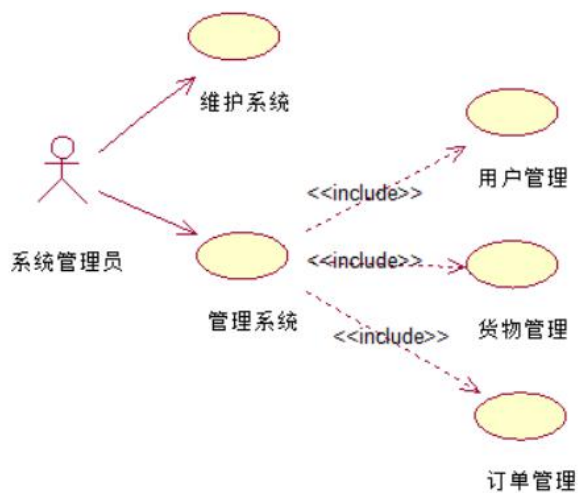


			信息	息”，填入所需完善的个人信息	
4	发布货物信息	货主	货主发布货物信息等待承运	货主点击发布货物信息，填入货物信息发布在网站上	货物信息表
5	查看货物信息	司机	司机查看网站上已发布的货物信息	司机点击“查看货物信息”	货物信息表
6	选择司机	货主	货主选择申请承运的司机	货主点击“查看司机”	司机信息表
7	修改运单状态	司机	司机实时更新货物承运状态	司机点击“修改运单状态”	订单表
8	查看运单状况	货主	货主获取实时运单状态	货主点击“查看运单状态”	订单表
9	确认收货	货主	货主确认收到货物	货主点击“确认收货”	订单表
10	管理账户	管理员	管理员管理账户	管理员点击“账户管理”	用户表
11	管理订单	管理员	管理员管理订单	管理员点击“订单管理”	订单表
12	查看消息	货主和司机	货主和司机查看消息	点击“我的消息”界面	消息表
13	查看统计数据	货主和司机	货主和司机查看统计数据	点击“历史订单”界面	订单表

### 1.3 系统需求描述

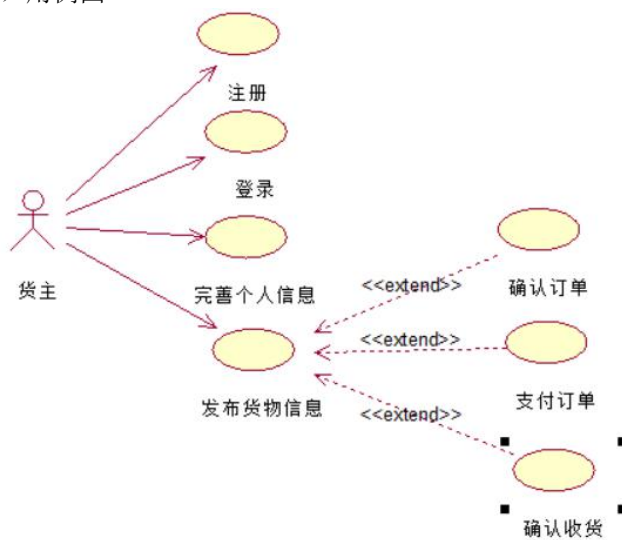
#### 1.3.1 系统用例模型构建





### 1.3.1.1 货主用例建模

#### (1) 用例图



#### (2) 用例表

用例标识和名称	注册
描述	货主进行注册，获得登录物流平台的身份

参与者	货主
优先级	基本的
风险	低
前置条件和假设	系统已进入注册界面
主事件流	<ol style="list-style-type: none"> <li>1. 货主进入注册界面</li> <li>2. 输入手机号</li> <li>3. 选择注册身份</li> <li>4. 输入密码</li> <li>5. 点击注册按钮</li> </ol>
可选事件流	在注册过程中，可随时选择退出，不注册
后置条件	注册成功，进入登录界面，账号密码保存到用户表中
用例标识和名称	登录
描述	货主可通过账号密码登录系统
参与者	货主
优先级	基本的
风险	低
前置条件和假设	货主已进入登陆界面
主事件流	<ol style="list-style-type: none"> <li>1. 货主进入登录界面</li> <li>2. 输入账号</li> <li>3. 输入密码</li> <li>4. 点击登录按钮</li> </ol>

可选事件流	若 2 步，没有账号，则显示需要注册 若 3 步，密码输入错误，则显示重新输入
后置条件	登陆成功，进入系统

用例标识和名称	完善个人信息
描述	货主完善个人信息
参与者	货主
优先级	基本的
风险	低
前置条件和假设	货主已登入系统，进入个人信息界面
主事件流	1. 货主进入个人信息界面 2. 输入姓名、联系方式、常用货物、住址 3. 点击保存按钮
可选事件流	若 2 输入错误，可重新输入
后置条件	点击保存按钮后，信息保存在货主信息表中
用例标识和名称	发布货物信息
描述	货主填写货物的各种信息
参与者	货主
优先级	基本的

风险	低
前置条件和假设	系统进入货物信息界面
主事件流	<ol style="list-style-type: none"> <li>1. 货主进入主界面</li> <li>2. 货主点击发布货物信息按钮</li> <li>3. 填写货物名称、货物类别、运输起点、运输目的地、运费要求</li> <li>4. 货主点击确定</li> </ol>
可选事件流	若填写错误，可重新填写
后置条件	完成后，保存在货物信息表中

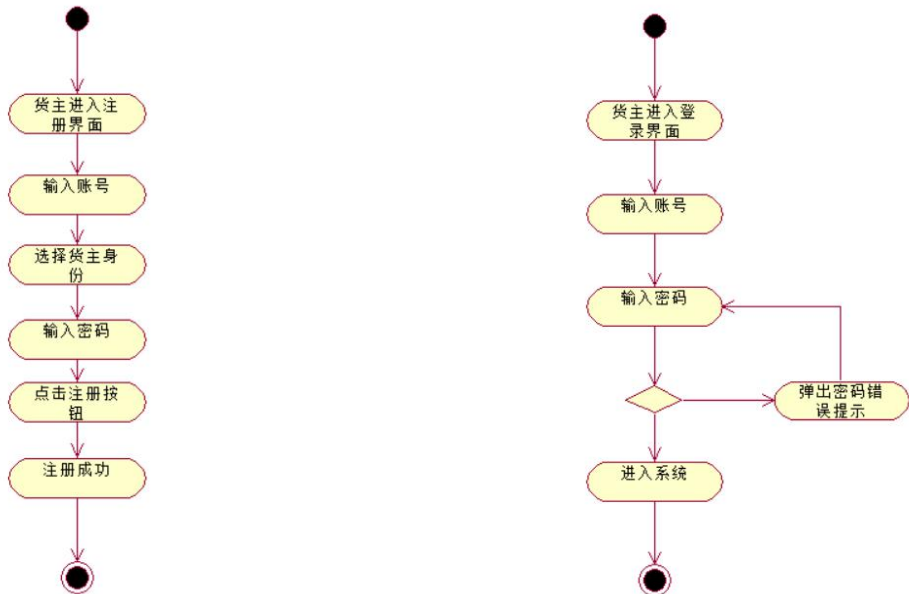
用例标识和名称	选择司机
描述	货主通过选择司机来确认订单
参与者	货主
优先级	基本的
风险	低
前置条件和假设	系统进入货物信息界面
主事件流	<ol style="list-style-type: none"> <li>1. 货主进入货物信息界面</li> <li>2. 货主点击查看司机按钮</li> <li>3. 货主选择司机</li> <li>4. 货主点击确定</li> </ol>
可选事件流	若只有一个司机，则不选择。

后置条件	生成订单保存到订单表中
------	-------------

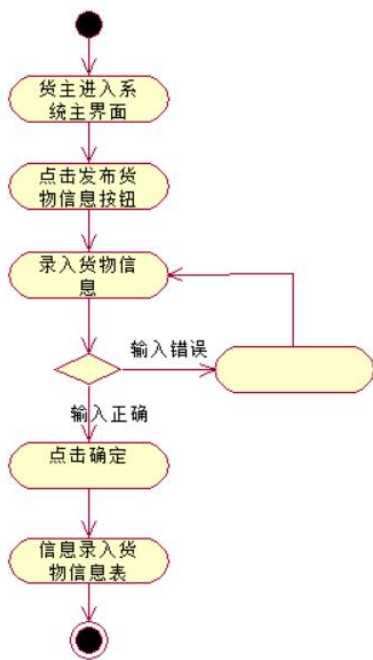
用例标识和名称	确认收货
描述	货主确认货品到达以确认收货
参与者	货主
优先级	基本的
风险	低
前置条件和假设	货主已进入订单界面
主事件流	<ol style="list-style-type: none"> <li>1. 货主进入订单界面</li> <li>2. 货主确定货品已到达且无损坏</li> <li>3. 点击确认收货按钮</li> </ol>
可选事件流	若货品没到达，可选择返回
后置条件	确认收货，显示订单已完成
用例标识和名称	支付订单
描述	货主确认订单完成后，支付订单
参与者	货主
优先级	基本的
风险	低

前置条件和假设	货主已确认收货
主事件流	<ol style="list-style-type: none"> <li>1. 进入订单支付界面</li> <li>2. 点击支付按钮</li> <li>3. 输入支付密码</li> </ol>
可选事件流	若支付密码输入错误，则重新输入
后置条件	支付成功，返回系统首界面

(3) 活动图

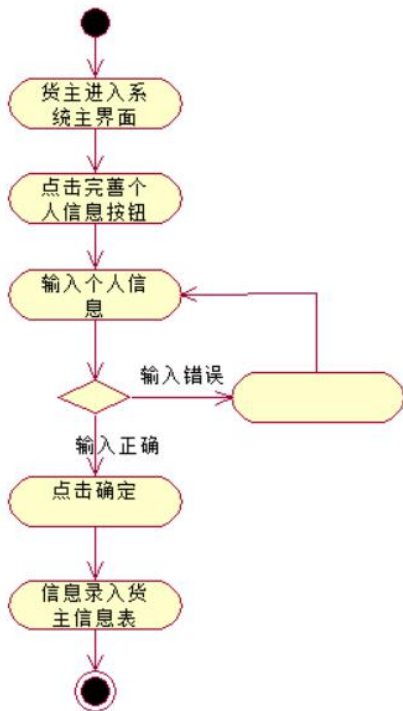






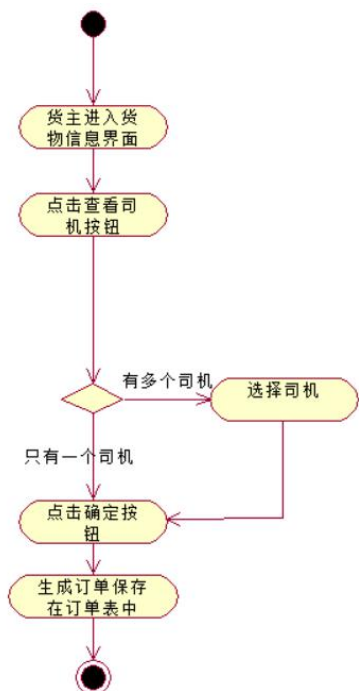
货主注册活动图

货主登录活动图

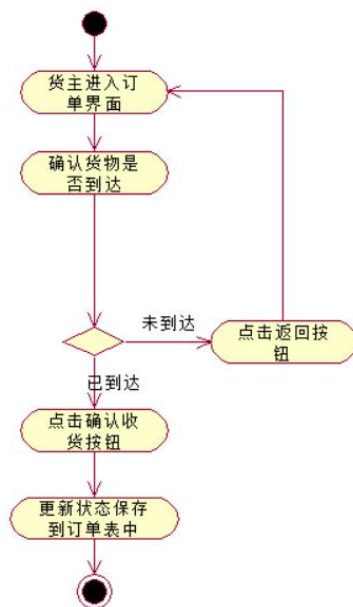


货主完善个人信息活动图

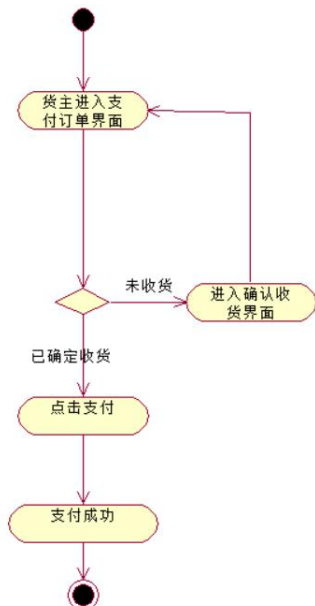
货主发布货物信息活动图



选择司机活动图



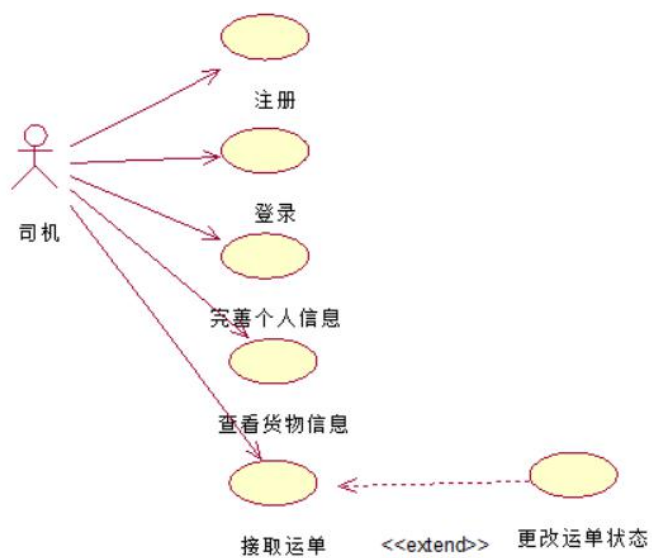
确认收货活动图



确认收货活动图

### 1.3.1.2 司机用例建模

#### (1) 用例图



(2) 用例表

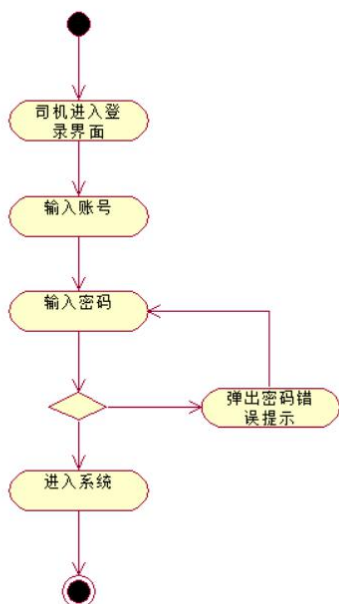
用例标识和名称	注册
描述	司机进行注册，获得登录物流平台的身份
参与者	司机
优先级	基本的
风险	低
前置条件和假设	系统已进入注册登录界面
主事件流	1. 司机进入注册界面 2. 输入手机号 3. 选择注册身份——司机 4. 输入密码 5. 点击注册按钮
可选事件流	在注册过程中，可随时选择退出，不注册

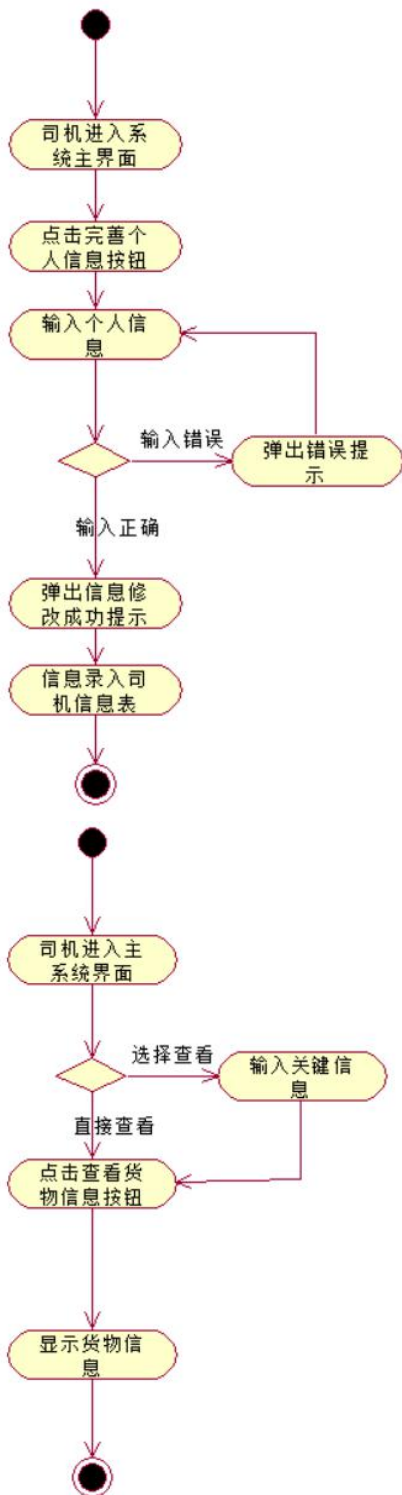
后置条件	注册成功，进入登录界面
用例标识和名称	登录
描述	司机可通过账号密码登录系统
参与者	司机
优先级	基本的
风险	低
前置条件和假设	司机已进入登陆界面
主事件流	1. 司机进入登录界面 2. 输入账号 3. 输入密码 4. 点击登录按钮
可选事件流	若 2 步，没有账号，则显示需要注册 若 3 步，密码输入错误，则显示重新输入
后置条件	登陆成功，进入系统
用例标识和名称	完善个人信息
描述	司机完善个人信息
参与者	司机
优先级	基本的
风险	低

前置条件和假设	司机已登入系统，进入个人信息界面
主事件流	<ol style="list-style-type: none"> <li>1. 司机进入个人信息界面</li> <li>2. 输入姓名、联系方式、车型、车牌号、可运输类型</li> <li>3. 点击保存按钮</li> </ol>
可选事件流	若 2 输入错误，可重新输入
后置条件	点击保存按钮后，显示已保存的信息
用例标识和名称	查看货物信息
描述	司机可以查看货物信息
参与者	司机
优先级	基本的
风险	低
前置条件和假设	已进入货物信息界面
主事件流	<ol style="list-style-type: none"> <li>1. 司机进入货物信息界面</li> <li>2. 司机点击查看按钮</li> <li>3. 显示货物信息</li> </ol>
可选事件流	第 2 步司机可选择查询查看，通过关键字查询
后置条件	查看完信息后，系统还在信息界面，可选择退出
用例标识和名称	更改运单状态

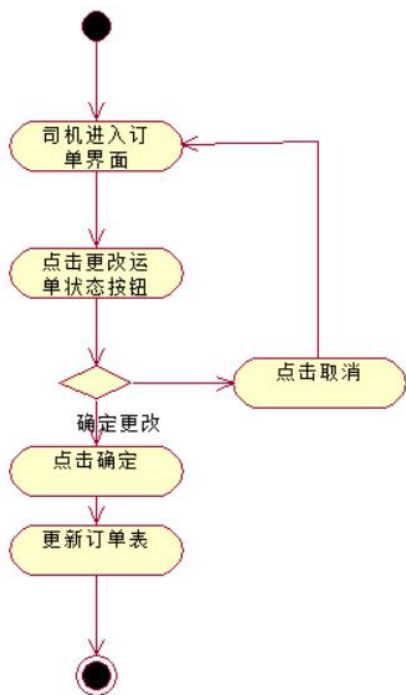
描述	司机对运单状态进行更改
参与者	司机
优先级	基本的
风险	低
前置条件和假设	司机已登入系统，进入订单界面
主事件流	1. 司机进入订单界面 2. 点击更改运单状态按钮 3. 点击确定按钮
可选事件流	第 2 步若司机反悔不想接取，也可以取消
后置条件	点击确定后，更新订单表

(3) 活动图



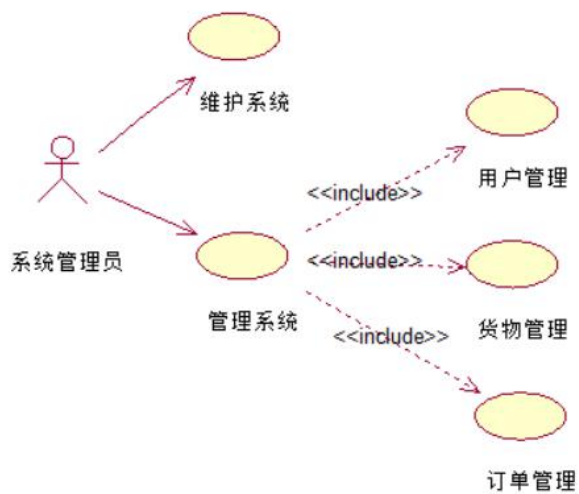






### 1.3.1.3 系统管理员用例建模

#### (1) 用例图



#### (2) 用例表

用例标识和名称	用户管理
---------	------

描述	管理员可以对用户进行管理，对用户账号进行修改添加删除等操作
参与者	管理员
优先级	基本的
风险	低
前置条件和假设	管理员已登入系统，进入用户管理界面
主事件流	<ol style="list-style-type: none"> <li>1. 管理员进入用户管理界面</li> <li>2. 2.1 管理员选择用户，点击删除按钮 2.2 显示删除成功</li> <li>3. 3.1 管理员点击添加按钮 3.2 输入账号密码 3.3 显示添加成功</li> <li>4. 4.1 管理员收到重置密码请求 4.2 管理员重置密码 4.3 显示修改成功</li> </ol>
可选事件流	可随时返回
后置条件	更新的用户信息保存到用户表中
用例标识和名称	货物管理
描述	管理员对货物进行管理
参与者	管理员
优先级	基本的
风险	低

前置条件和假设	管理员已登入系统，进入货物管理界面
主事件流	1. 管理员进入货物管理界面 2. 选择货物信息 3. 点击删除状态按钮 4. 点击确定按钮
可选事件流	第 4 步若不想删除，也可以取消
后置条件	点击确定后，把更新的货物信息保存到货物信息表中
用例标识和名称	订单管理
描述	管理员对订单进行管理
参与者	管理员
优先级	基本的
风险	低
前置条件和假设	管理员已登入系统，进入订单管理界面
主事件流	1. 管理员进入订单管理界面 2. 选择想要删除的订单信息 3. 点击删除按钮 4. 点击确定按钮
可选事件流	第 4 步若反悔不想删除，也可以取消
后置条件	点击确定后，更新订单表

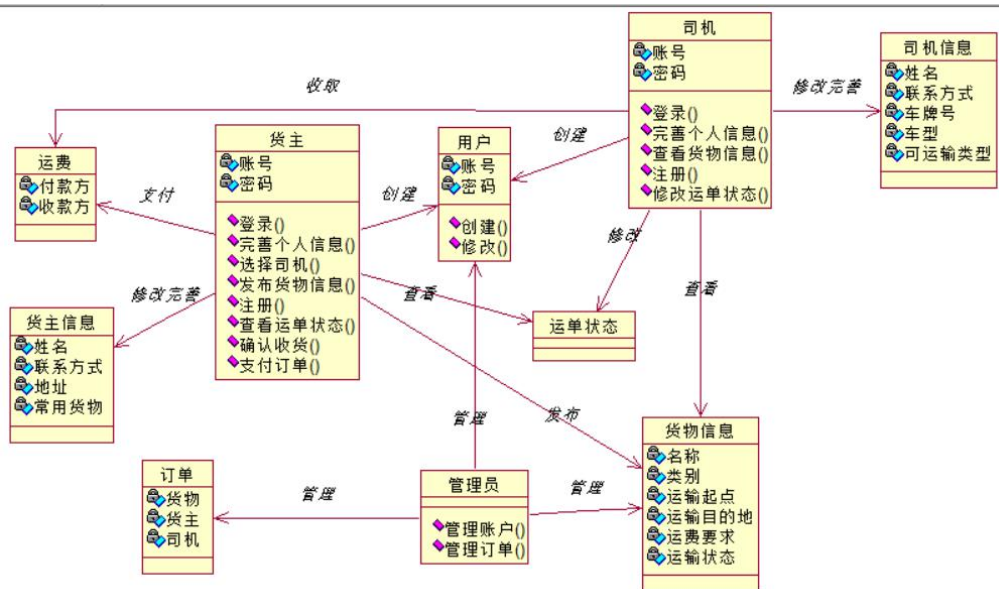
### 1. 3. 2 分析类模型

#### 1. 3. 2. 1 关键抽象表

候选关键抽象	排除原因	选定名称
货主		货主
司机		司机
个人信息		货主信息
姓名	货主和司机的属性	
联系方式	货主和司机的属性	
常用货物	货主的属性	
地址	货主的属性	
车型	司机的属性	
车牌号	司机的属性	
可运输类型	司机的属性	
个人信息		司机信息
货物信息		货物信息
货物名称	货物的属性	
货物类别	货物的属性	
运输起点	货物的属性	

运输目的地	货物的属性	
运费要求	货物的属性	
运输状态	货物的属性	
平台	概念太大	
运单状态		运单状态
货物运单		订单
运费		运费
管理员		管理员
账户		用户

### 1.3.2.2 类图模型



## 2. 系统性能需求

表 2-1 性能需求点列表

编 号	性能名称	使用部门	使用岗位	性能描述	目标	检验指标
1	账户响应	所有部门	所有岗位	账号登录系统 响应时间尽可能短	响应时间不超过 5 秒	响应时间
2	搜索响应	所有部门	所有岗位	搜索时间尽可能短	搜索时机不超过 3 秒	搜索时间
3	更新处理时间	所有部门	管理员、司机、货主	更新处理时间尽可能短	更新处理时间不超过 3 秒	处理时间
4	数据的转换和 传送时间	所有部门	系统	尽可能快的转换和传送数据	时间不超过 3 秒	传送时间
5	支持的终端数	所有部门	管理员、司机、货主	所有 pc 端、手机端都可以使用	pc 端、手机端可以完美使用	Pc、手机端成功使用网站
6	支持的并行操作的使用者数	所有部门	管理员、司机、货主	并行操作的使用人数尽可能多	使用用户要求在 2000 以上	动态用户数量
7	处理的文件和记录数	所有部门	系统	处理的文件和存储的记录数尽可能多	处理的文件和存储的记录数在 20000 条以上	存储条数
8	精度要求	所有部门	系统	尽可能准确	索引精度尽可能做到百分之百	索引精度
9	安全性	所有部门	系统	不同类型用户拥有不同的操作权限，提高数据安全性，避免系统遭受攻击、被注入等损失	用户权限与忽略条件下数据安全	
10	稳定性	所有部门	管理员、司机、货主	在整个使用过程中，所有用户均可稳定正确的使用	使用过程中的稳定性	
11	可扩展性	所有部门	管理员、司机、货主	用户能在 PC 端、移动端对系统进行操作	跨平台使用稳定	

12	灵活性	所有部门	管理员、司机、货主	网站灵活性较好	设计合理, 体验感良好	
13	故障处理能力	所有部门	管理员、司机、货主	能尽可能处理操作过程中出现的各种异常	故障处理	

### 3. 系统界面与接口需求

#### 3.1 界面需求

##### • 界面原则要求

1. **页面内容**: 主题突出明确, 栏目、菜单设置和布局合理, 传递的信息准确及时。

内

容丰富, 文字准确, 语句通顺; 专用术语规范, 行文格式统一规范。

2. **导航结构**: 页面具有明确的导航指示, 且便于理解, 方便用户使用。

链

3. **技术环境**: 页面大小适当, 能用各种浏览器以不同分辨率浏览; 无错误链接和空链接。

4. **艺术风格**: 版面形象清新悦目、布局合理, 字号大小适宜、字体选择合适, 前后一致, 美观大方; 动与静搭配恰当; 色彩和谐自然, 与主题内容相协调。

##### • 某些功能模块的特殊的界面要求

[1] 输入设备: 键盘、鼠标;

[2] 输出设备: 显示器;

[3] 显示风格: IE 界面;

[4] 显示方式: 1024\*768;

[5] 输出格式: 显示布局;

#### 3.2 接口需求

表 2-2 外部接口需求点列表

编 号	接口名称	接口规范	接口标准	入口参数	出口参数	传输频率
1	人机接口	选择和输入界面、数据输入输出	选择和输入界面、数据输入输出	管理员、司机、货主账号登录	成功登录	
2	机机接口	数据备份、数据迁移、数据导入、数据传输	数据备份、数据迁移、数据导入、数据传输	数据库实体主码一致性、属性一致性	符合一致性约束、数据导入正确、传输迅速安全	

3						
---	--	--	--	--	--	--

#### 4. 目标系统假设与约束条件

[1]此系统严格遵循国家法律法规等相关制度。

[2]最低硬件要求：

处理器：intel i3 或等效处理器；

内存：128MB；

[3]运行环境：

操作系统 Win10 及以上。

浏览器:IE6.0 或更高版本

[4]系统投入使用的最晚日期为 2021-11-30。



# 设计规格说明书

## 1. 引言

### 1.1 命名规则

#### 1) 变量对象命名规则:

变量名应以小写字母开头，多个英文单词之间采用首字母大写方式分割；类名应以大写字母开头，多个英文单词之间采用首字母大写方式分割；

xml 文件代码个变量均用小写字母和数字，多个英文单词之间用下划线分割；

#### 2) 数据库对象命名规则:

申明数据库表名、字段名、索引名、视图名等对象的命名规则。各表名命名规则同类名；字段名，索引名，视图名命名规则同变量名；

### 1.2 术语定义

术语定义表

序号	术语名称	术语定义
1	总体结构	软件系统的总体逻辑结构。按照不同的设计方法，有不同的总体逻辑结构。若采用面向功能或面向数据的设计方法，则总体逻辑结构为一树形的功能模块结构图。若采用面向对象或面向部件（构件）的设计方法，则总体逻辑结构为部件（构件）的组装图
2	外部接口	本软件系统与其他软件系统之间的接口，接口设施可以是中间件。接口描述包括：传输方式、带宽、数据结构、传输频率、传输量、传输协议
3	数据结构	数据结构包括：数据库表的结构、其他数据结构等
4	概念数据模型 CDM	关系数据库的逻辑设计模型，叫做概念数据模型。主要内容包括一张逻辑 E-R 图及其相应的数据字典
5	物理数据模型 PDM	关系数据库的物理设计模型，叫做物理数据模型。主要内容包括一张物理表关系图及其相应的数据字典
6	角色	数据库中享有某些特权操作的用户，叫做角色。角色的权利通过授权来实现
7	子系统	具有相对独立功能的小系统叫做子系统。一个大的软件系统可以划分为多个子系统，每个子系统可由多个

		模块或多个部件组成
8	模块	具有功能独立、能被调用的信息单元叫做模块。模块是结构化设计中的概念
9	内部接口	软件系统内部各子系统之间、各部件之间、各模板之间的接口，叫做内部接口。接口描述包括：调用方式、入口信息、出口信息等
10	相关文件	相关文件是指当本文件内容变更后，可能引起变更的其他文件。如需求分析报告、详细设计说明书、测试计划、用户手册
11	参考资料	参考资料是指本文件书写时用到的其他资料。如各种有关规范、模板、标准、准则
12	HTML	超文本标记语言，可包括图片链接音乐等信息。
13	CSS	层叠样式表是一种用来表现 HTML（标准通用标记语言的一个应用）或 XML 等文件样式的计算机语言
14	JavaScript	JavaScript 一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型
15	MYSQL	关系型数据库管理系统
16	云服务器	云服务器是一种简单高效、安全可靠、处理能力可弹性伸缩的计算服务
17	异步请求 Ajax	AJAX 即异步 JavaScript 和 XML，是指一种创建交互式网页应用的网页开发技术
18	jquery	一个快速、简洁的 JavaScript 框架，封装 JavaScript 常用的功能代码，提供一种简便的 JavaScript 设计模式，优化 HTML 文档操作、事件处理、动画设计和 Ajax 交互
19	Vuejs	个构建数据驱动的 web 界面的渐进式框架通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件

20	apache	一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台和安全性能被广泛使用。它快速、可靠并且可通过简单的 API 扩展，将 Perl/Python 等解释器编译到服务器中
----	--------	--

### 1.3 相关文档

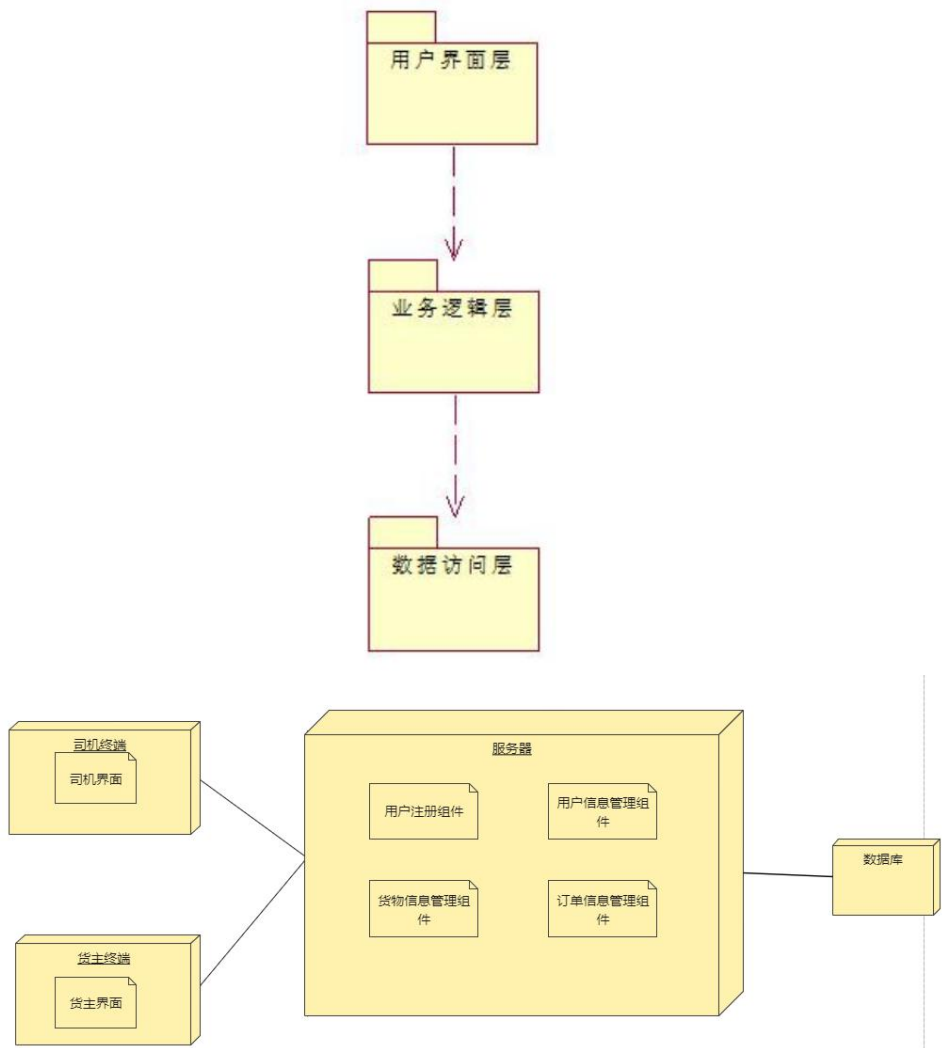
- [1] 需求分析报告
- [2] 数据库设计规范
- [3] 命名规范
- [4] 用户手册
- [5] 测试计划

## 2. 系统设计

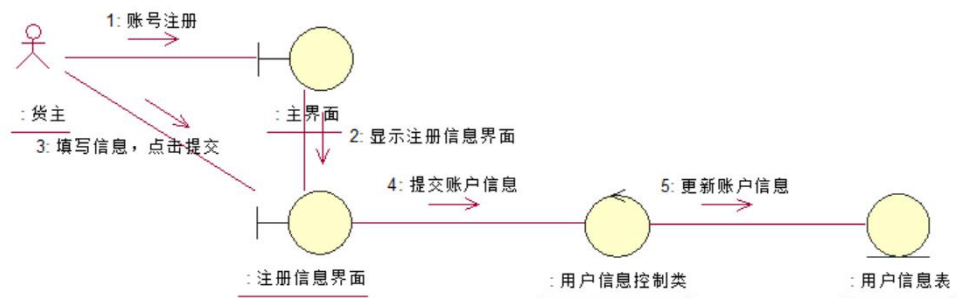
### 2.1 体系结构设计

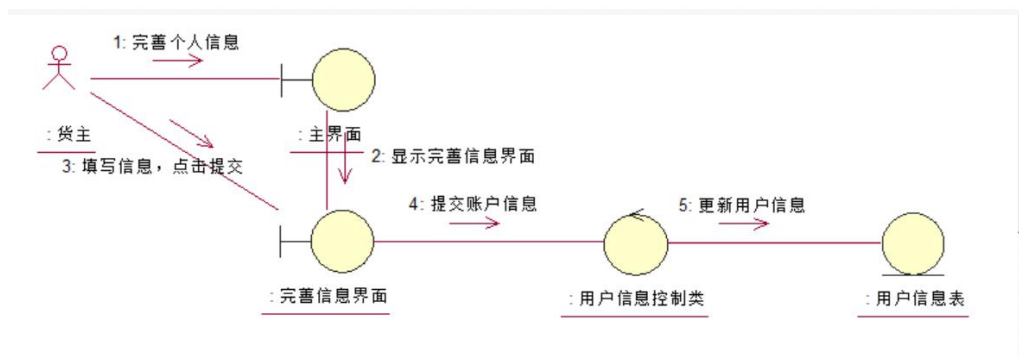
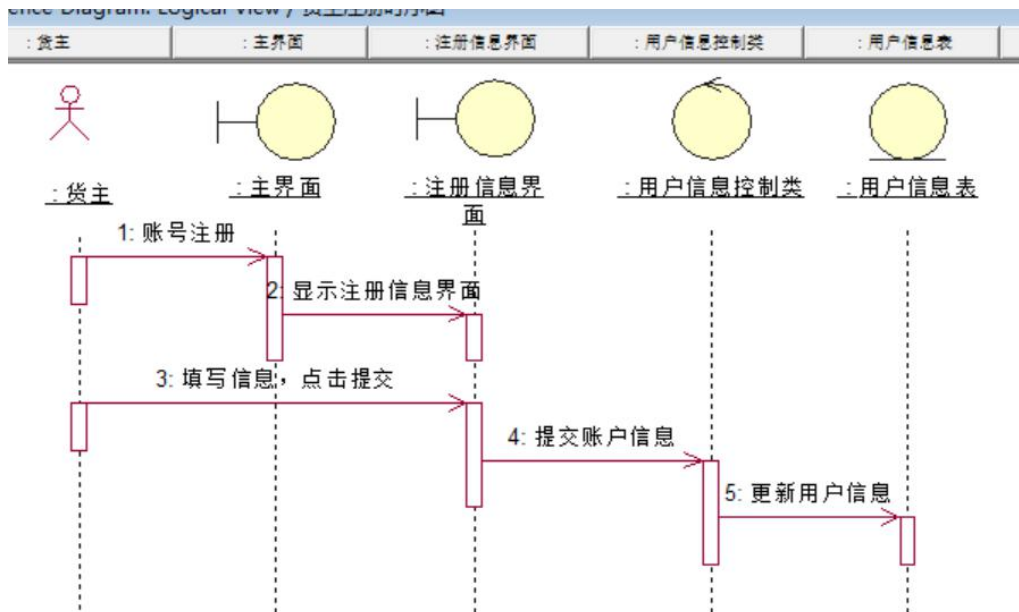
本系统结构为典型的 B/S 架构。与传统的 C/S (Client/Server) 架构的应用系统相比，B/S 架构系统在系统部署、实施以及维护方面都节约了大量成本。B/S 结构的特点是在客户端使用标准的 Web 页面浏览器(如 Internet Explorer 等)，不需要安装特殊的应用程序，减少了升级和维护的难度；所有的业务数据都保存在 Server 端，确保了数据的安全。基于 WEB 的软件系统其结构可以分为三层：

1. 用户界面层：为客户端提供对应用程序的访问。这一层由 ASP 窗体和代码隐藏文件组成。Web 窗体只是用 HTML 提供用户操作，而代码隐藏文件实现各种控件的事件处理。
2. 业务逻辑层：是为了解决特定问题的要求实现的，它包含各种业务规则和逻辑的实现。业务规则完成如客户申请信息的验证这样的任务。
3. 数据访问层：为业务规则层提供数据服务

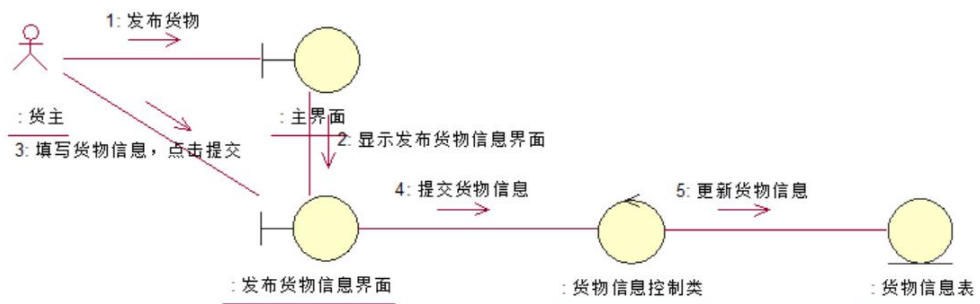
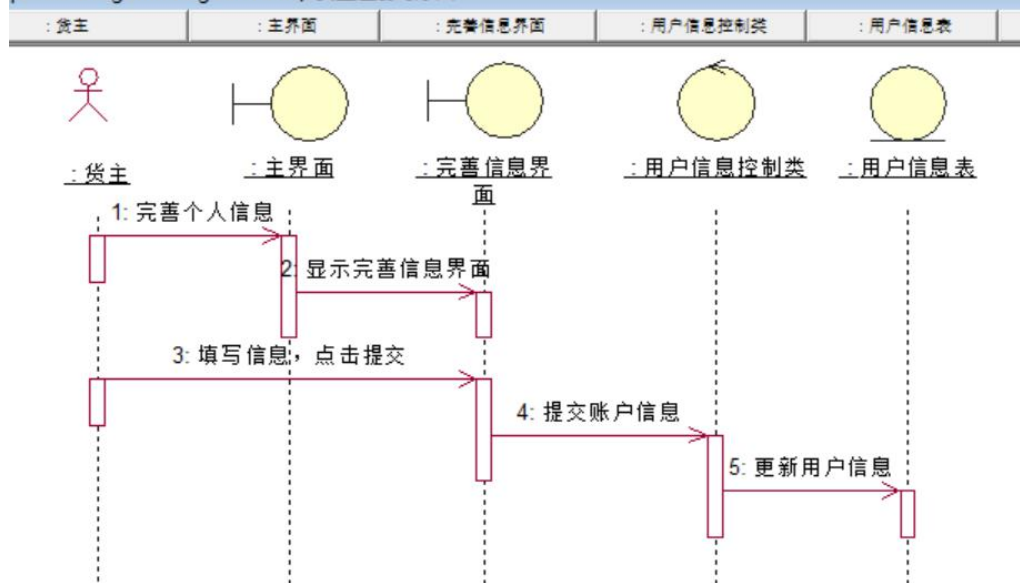


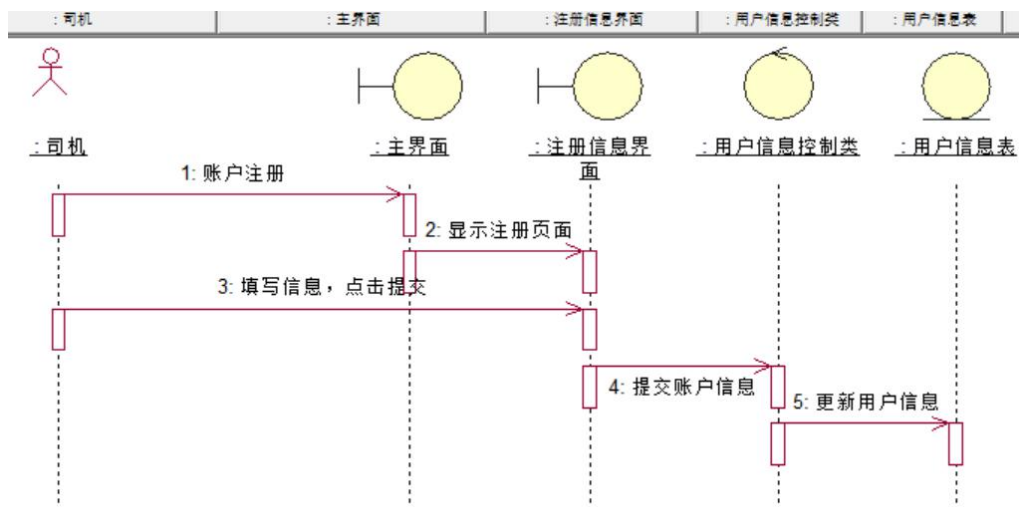
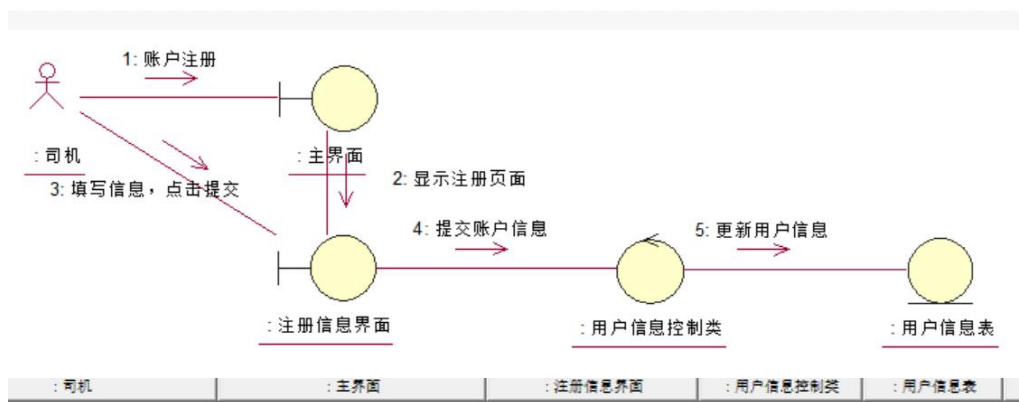
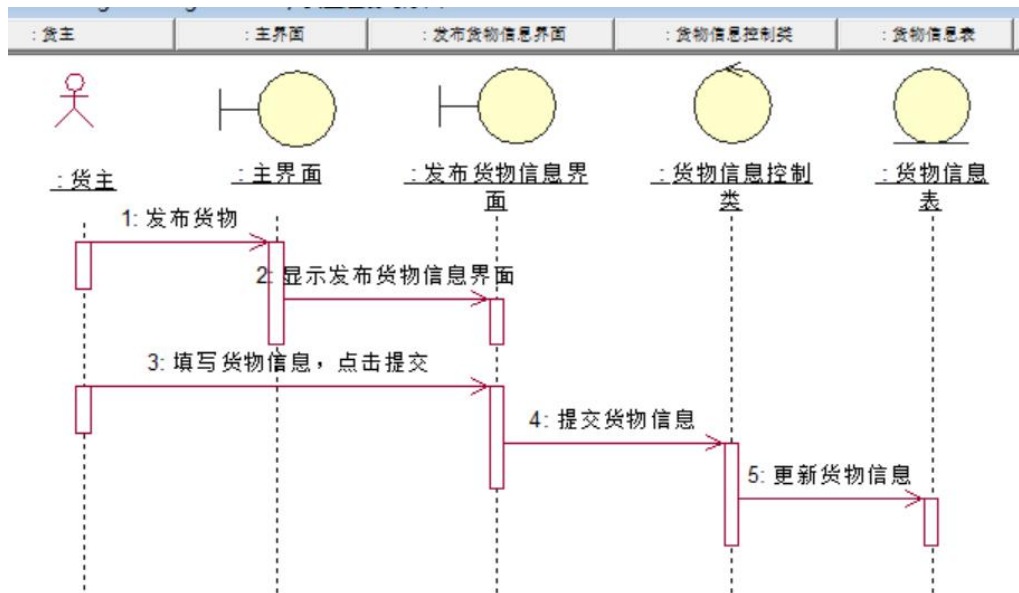
## 2.2 类设计

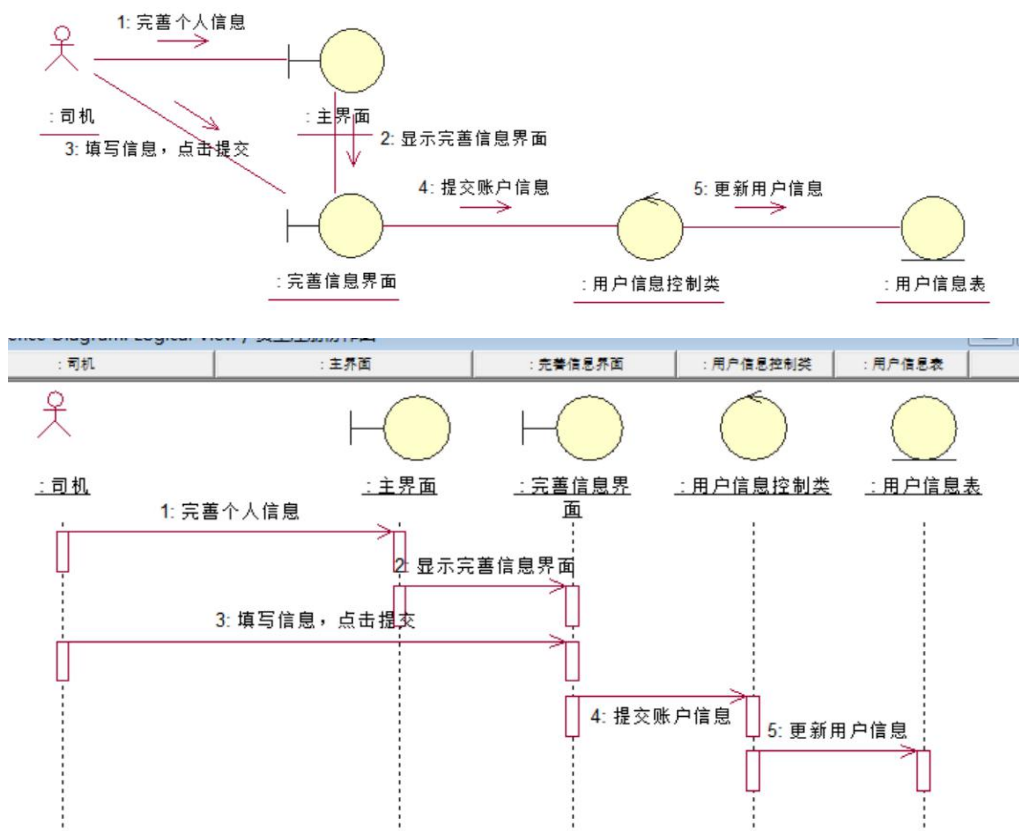




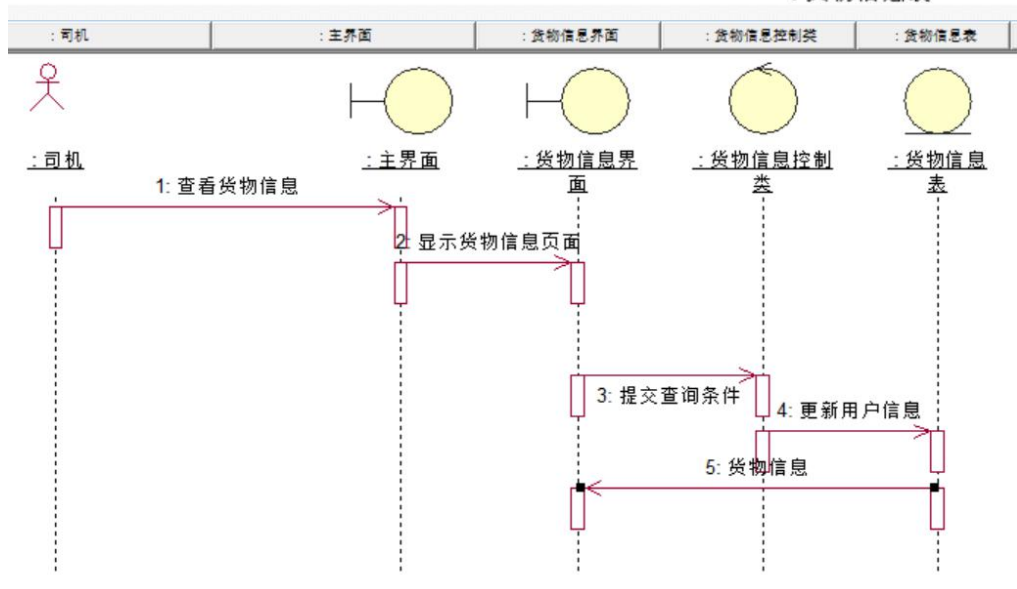
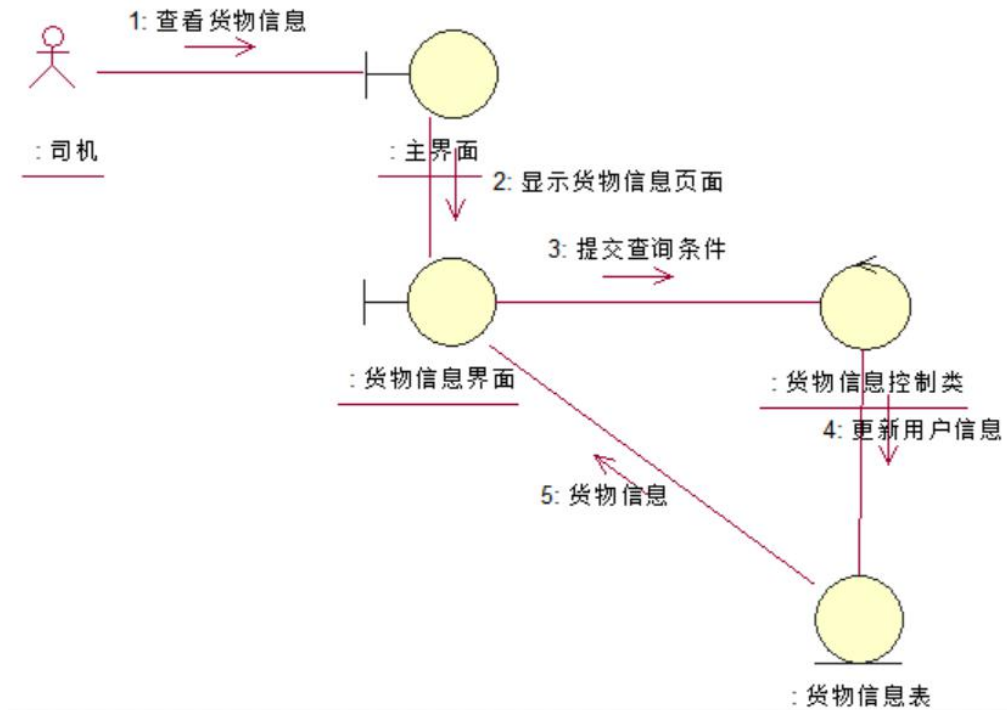
Sequence Diagram: Logical View / 货主注册时序图

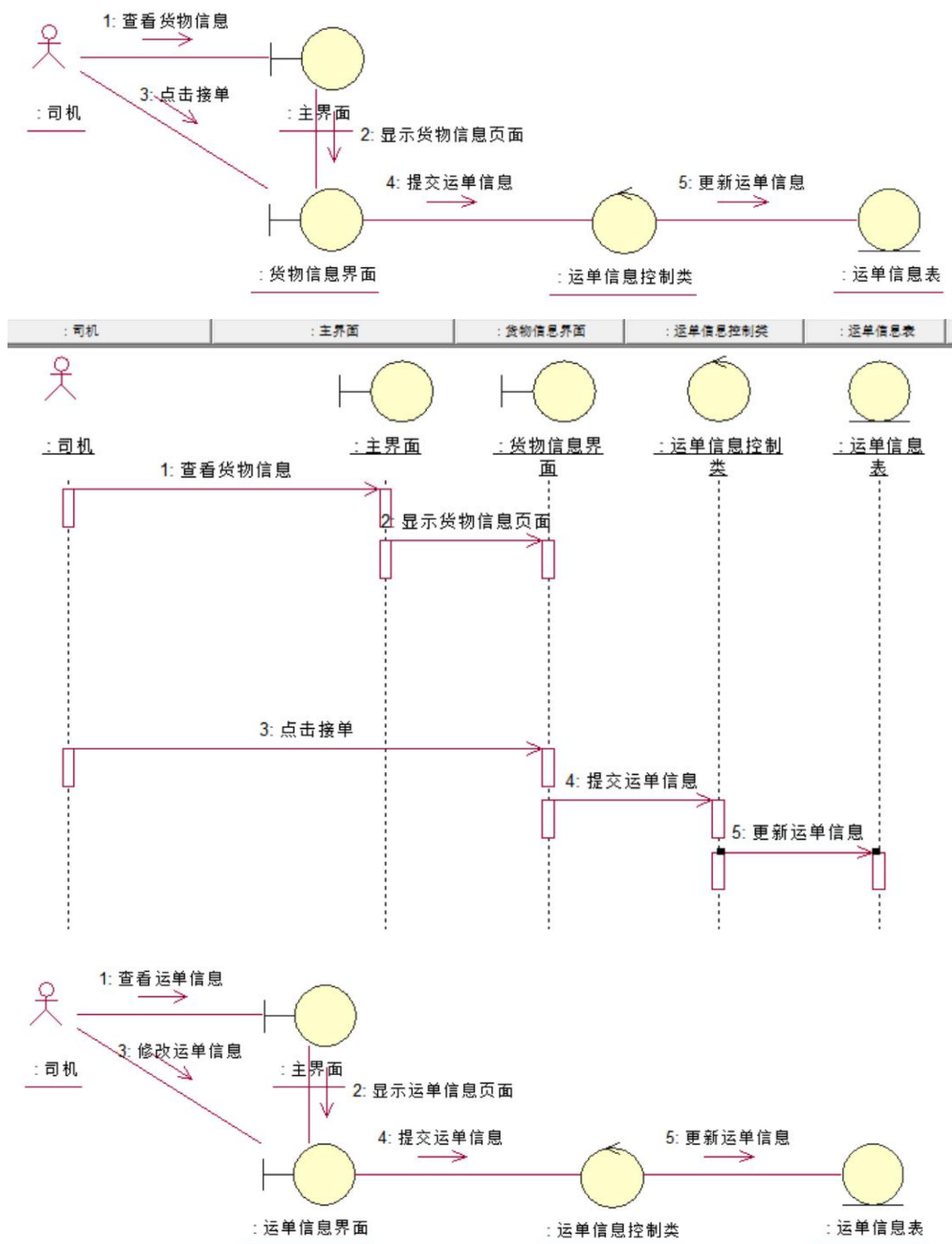


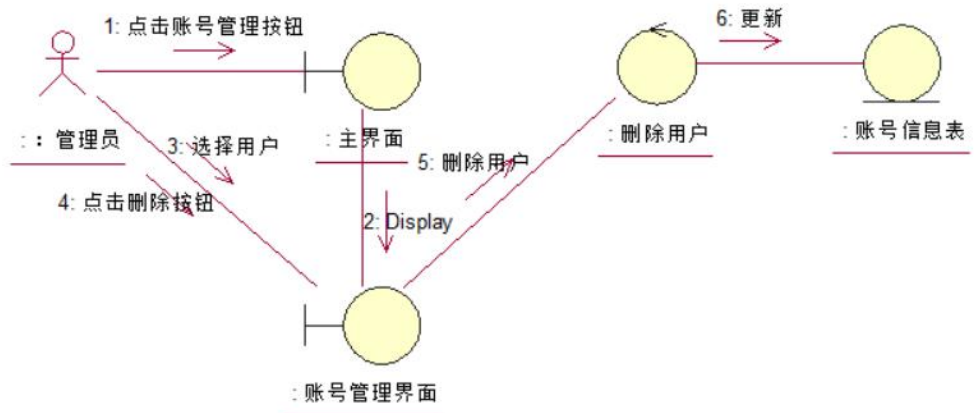
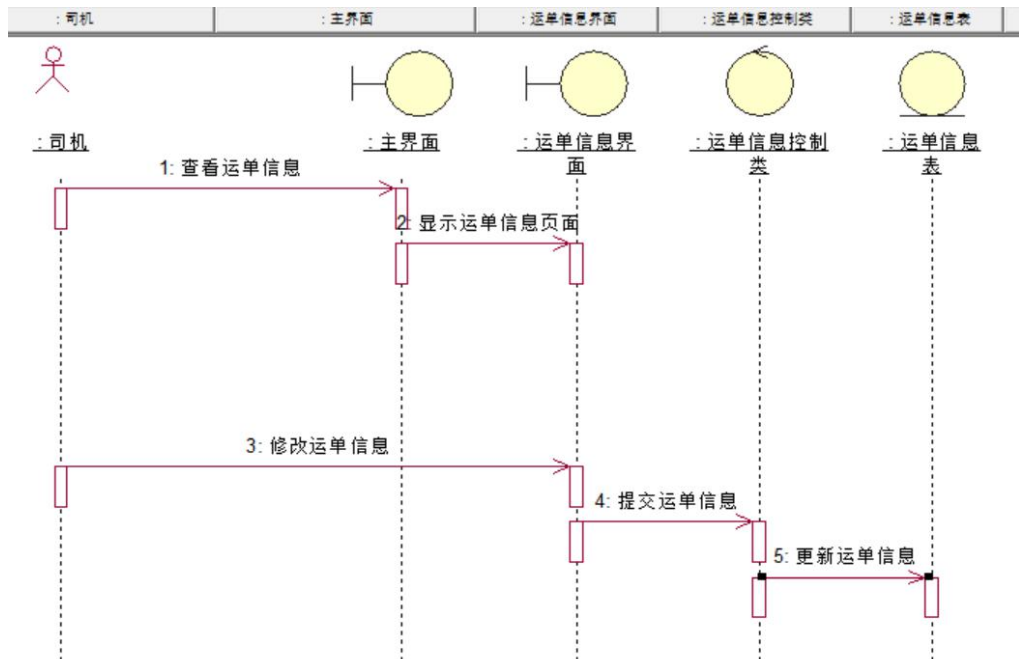


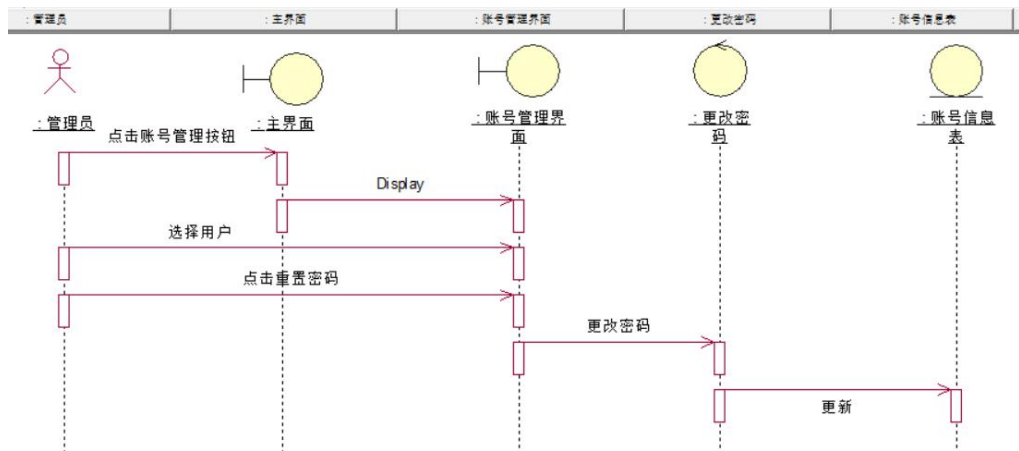
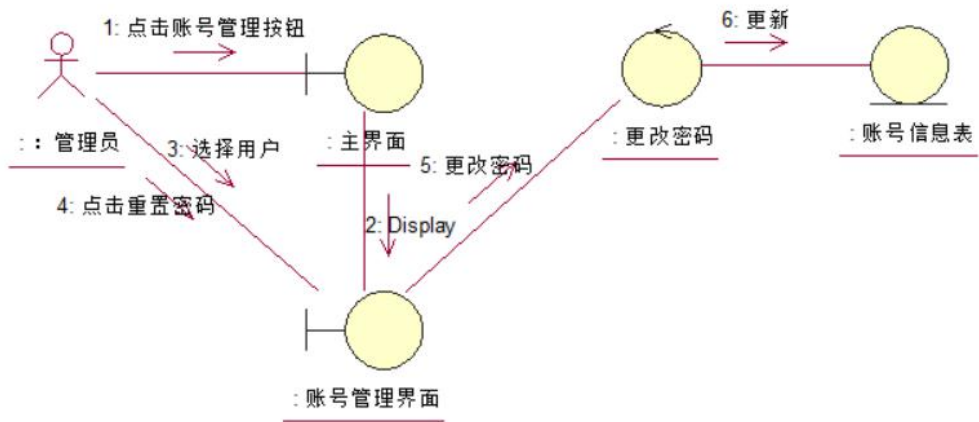
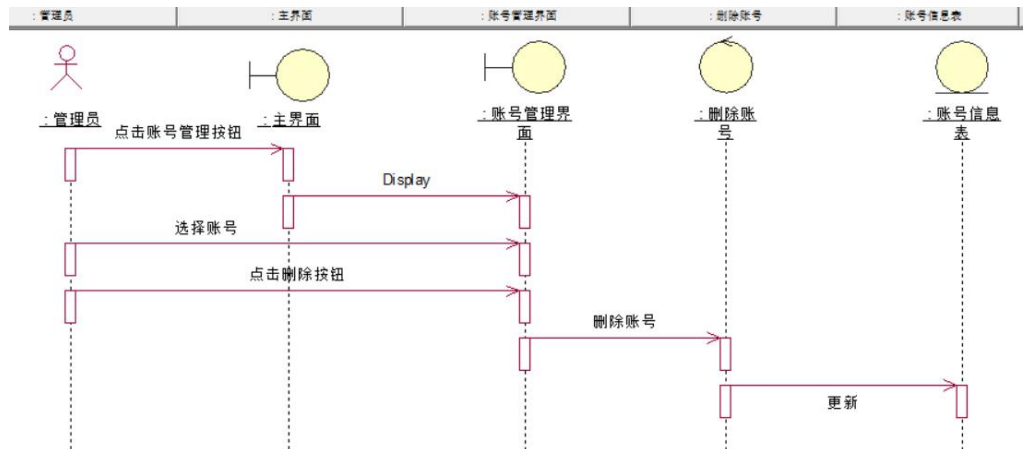


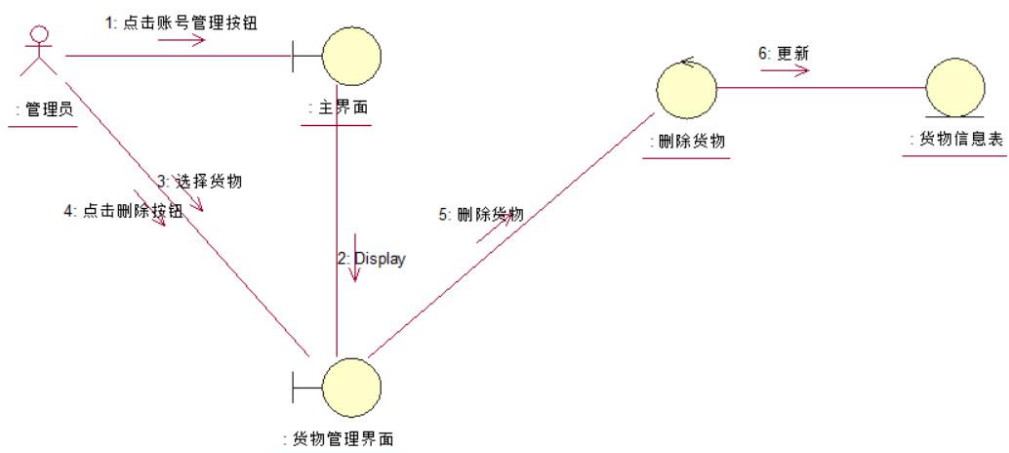
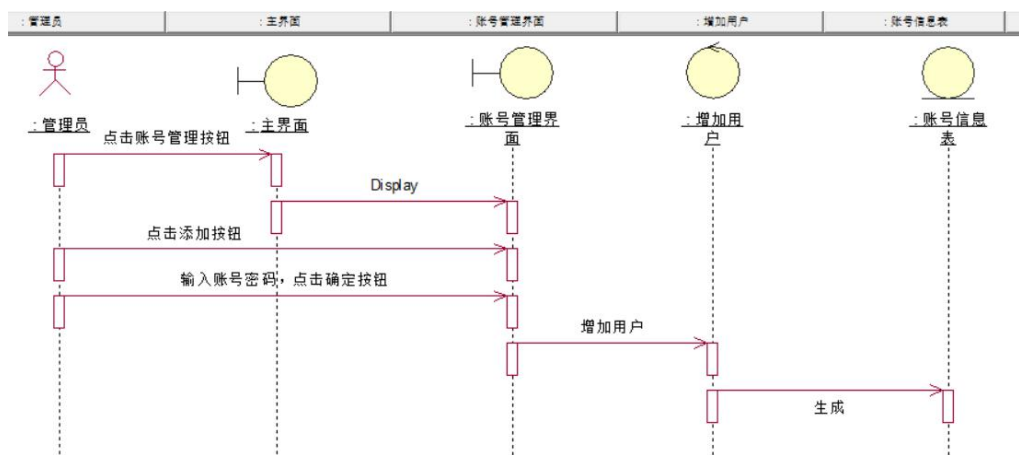
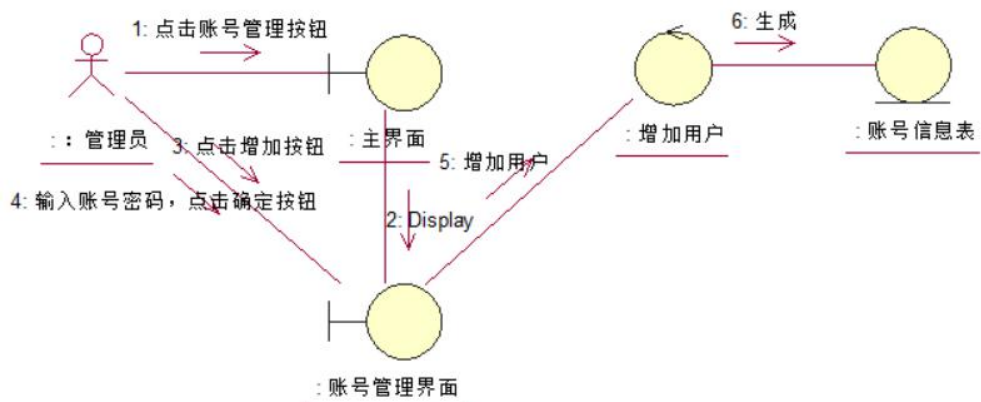


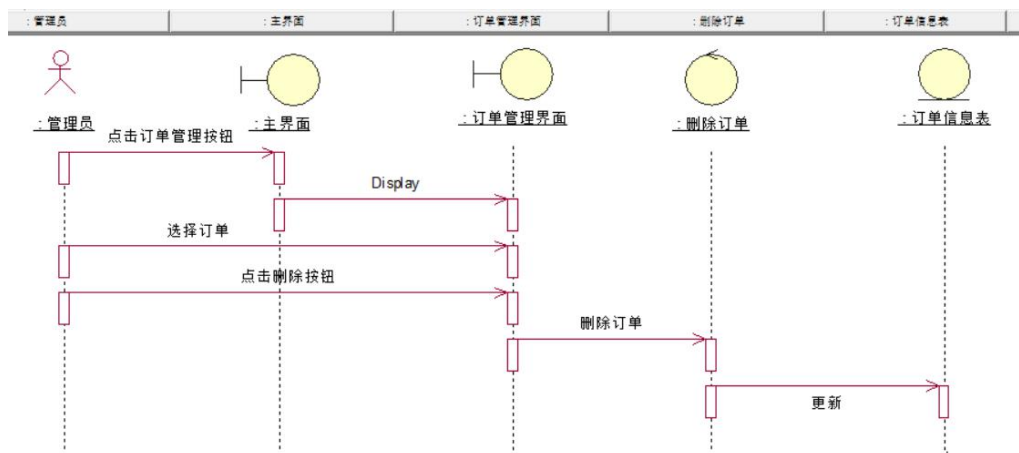
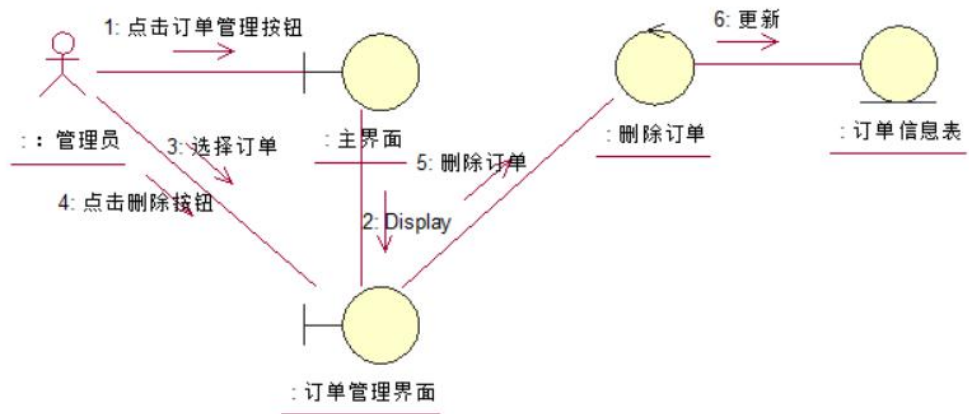
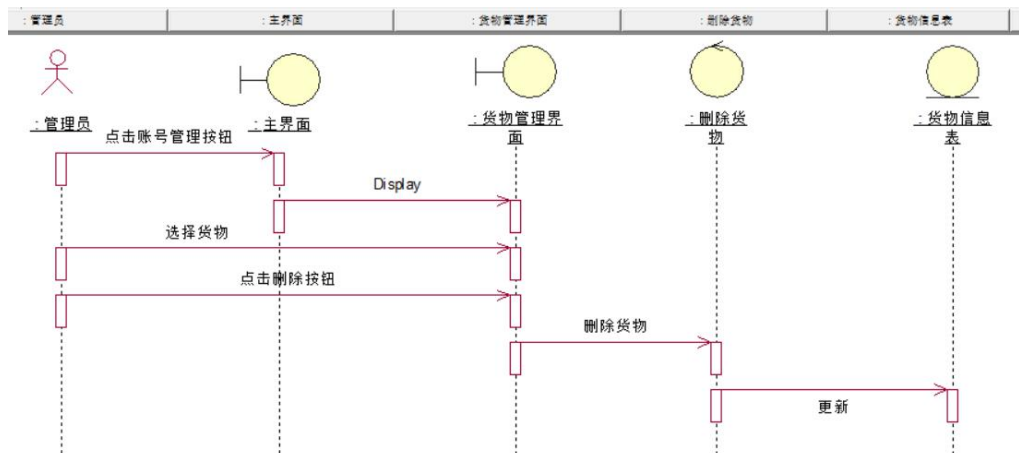






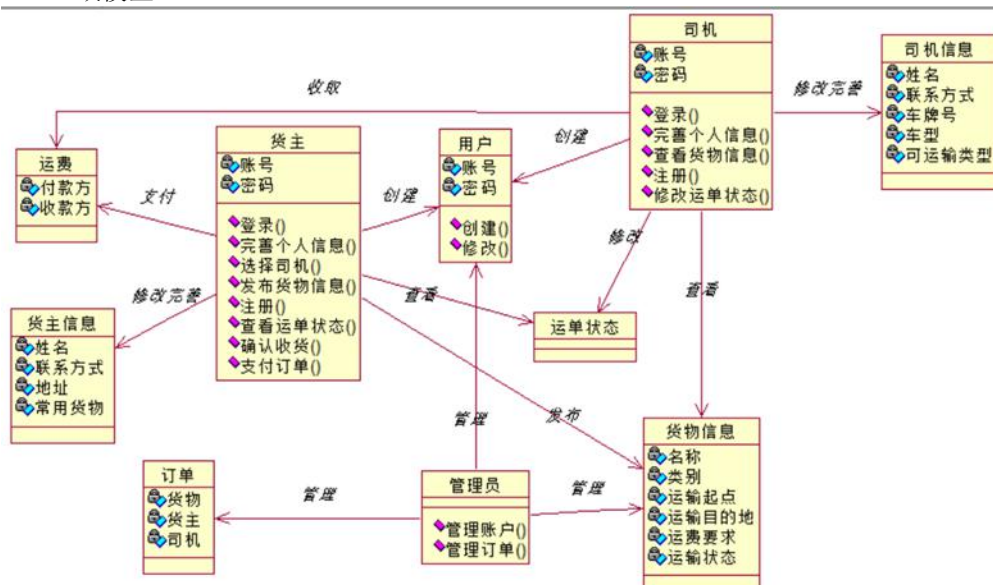






## 2.2 类说明

### 2.3.1 域模型



### 2.3.2 实体类清单

实体类清单

编号	类名（英文名）	功能简述	接口简述
E-1	用户信息表 (user information)	保存注册的账号密码，身份信息——货主	将注册信息保存到用户信息表中
E-2	用户信息表 (user information)	保存填入的个人信息	将货主的个人信息保存在货主信息表中
E-3	货物信息表 (cargo information)	保存填入的货物信息	将货主发布的货物信息保存在货物信息表中
E-4	用户信息表 (user information)	保存注册的账号密码，身份信息——司机	将注册信息保存到用户信息表中
E-5	用户信息表 (user information)	保存填入的个人信息	将司机的个人信息保存在货主信息表中
E-6	货物信息表 (cargo information)	司机查看货物信息	输入查询条件，显示货物信息表中的货物信息

E-7	运单信息表 (waybill information)	司机接取运单	司机接单后, 更新运单信息到运单表中
E-8	运单信息表 (waybill information)	司机更改运单状态	司机改变运单状态保存到运单表中
E-9	用户信息表 (user information)	管理员删除用户	管理员删除用户, 更新到用户信息表中
E-10	用户信息表 (user information)	管理员修改密码	管理员输入密码, 更新到用户信息表中
E-11	用户信息表 (user information)	管理员增加用户	管理员输入账号密码身份, 更新到用户信息表中
E-12	货物信息表 (cargo information)	管理员删除货物信息	选择货物删除, 更新货物信息表
E-13	订单信息表 (order information)	管理员删除订单信息	选择订单删除, 更新订单信息表

### 2.3.3 控制类清单

控制类清单

编号	类名 (英文名)	功能简述	接口简述
C-1	用户信息控制类	服务于货主注册账号	用户注册的接口
C-2	用户信息控制类	服务于货主修改个人信息	完善货主信息的接口
C-3	货物信息控制类	服务于货主发布货物信息	发布货物信息的接口
C-4	用户信息控制类	服务于司机注册账号	用户注册的接口



C-5	用户信息控制类	服务于司机修改个人信息	完善司机信息的接口
C-6	货物信息控制类	控制页面与内部货物信息数据的联系。	将数据库的货物信息实体数据传到发出请求的页面。
C-7	运单信息控制类	服务于司机接单	更新运单的接口。
C-8	运单信息控制类	服务于司机修改运单状态	修改运单状态的接口
C-9	删除用户	服务于管理员删除用户	删除用户的接口
C-10	更改密码	服务于管理员修改密码	修改密码的接口
C-11	增加用户	服务于管理员增加用户	增加用户的接口
C-12	删除货物	服务于管理员删除货物	删除货物的接口
C-13	删除订单	服务于管理员删除订单	删除订单的接口

2.3.4 边界类清单

边界类清单

编 号	类名（英文名）	功能简述	接口简述
B-1	主界面（MainUI:）	显示所有功能按钮与提醒界面。	可以进入各个功能页面，也可以从各个界面跳回主界面
B-2	注册信息界面	用户填写账号密码。	验证注册信息，成功失败信息返回到此界面。
B-3	完善信息界面	货主填写个人信息	保存个人信息，成功失败信息返回到此界面。

B-4	发布货物信息界面	货主填写货物信息	保存货物信息，成功失败信息返回到此界面。
B-5	完善信息界面	司机填写个人信息	保存个人信息，成功失败信息返回到此界面。
B-6	货物信息界面	显示货物信息	验证查询条件，成功失败信息返回到此界面。
B-7	货物信息界面	显示货物信息，接单	更新运单，成功失败信息返回到此界面。
B-8	运单信息界面	显示货物信息，修改运单状态	更新运单状态，成功失败信息返回到此界面。
B-9	账号管理界面	删除用户	删除用户，成功失败信息返回到此界面。
B-10	账号管理界面	修改用户密码	修改用户密码，成功失败信息返回到此界面。
B-11	账号管理界面	增加用户	增加用户，成功失败信息返回到此界面。
B-12	货物管理界面	删除货物	删除货物，成功失败信息返回到此界面。
B-13	订单管理界面	删除订单	删除订单，成功失败信息返回到此界面。

### 3. 数据结构设计

#### 3.1 持久类分析

1. 账号信息表
2. 司机信息表
3. 货主信息表
4. 货物信息表
5. 司机意向表
6. 订单表
7. 运单表
8. 账户信息表

- 9. 交易记录表
- 10. 货物类型表
- 11. 车辆类型
- 12. 消息信息表

### 3.2 数据库表名清单

3.2.1 数据库表名清单，如表 3-1 所示。

表 3-1

序号	中文表名	英文表名	表功能说明
1	账号信息表	login_user	存放账号信息
2	司机信息表	Driver	存放司机个人信息
3	货主信息表	Consumer	存放货主信息
4.	货物信息表	Goods	存放货物运输信息
5.	司机意向表	Driver_goods	存放司机有意向的运单
6.	订单表	orders	存放订单信息
8	账户信息表	Account	存放用户的账户信息
9	交易记录表	trading_recordrecord	存放用户的交易记录
10	货物类型表	Goods_type	存放货物类型
11	车辆类型	Car_type	存放车辆类型
12	消息记录表	message	存放消息记录

3.2.2ER 图



- login\_id->gender  
login\_id->telephone  
login\_id->used\_type  
login\_id->address
- (3) 货物信息表 (货物 id, 货物名称, 货物类别, 运输起点, 运输目的地, 运费要求, 发布时间)  
goods(goods\_id, name, type, start, destination, fees)  
goods\_id->name  
goods\_id->type  
goods\_id->start  
goods\_id->destination  
goods\_id->fees
- (4) 司机意向表 (订单意向 id, 货物 id, 司机 id, 运费报价, 生成时间)  
driver\_intention(intention\_id, goods\_id, login\_id, fees, create\_time)  
intention\_id->goods\_id  
intention\_id->login\_id  
intention\_id->fees  
intention\_id->create\_time
- (5) 订单信息表 (订单 id, 订单意向 id, 订单生成时间)  
Orders(order\_id, intention\_id, create\_time)  
Order\_id->intention\_id  
Order\_id->create\_time
- (6) 账户信息表 (账号, 账户余额)  
Account(login\_id, money)  
login\_id->money
- (7) 付款记录表 (订单号, 付款方, 收款方)  
Trading\_record(tradeId, order\_id, )
- (8) 货物类型表 (类型 id, 类型名称)  
Goods\_type(type\_id, name)  
Type\_id->name
- (9) 车辆类型表 (类型 id, 类型名称)  
car\_type(type\_id, name)  
Type\_id->name
- (11) 消息记录表 (消息 id, 货主 id, 货物 id, 来源 id, 类型)  
message(message\_id, user\_id, goods\_id, source, status)  
message\_id->user\_id  
message\_id->source  
message\_id->status  
message\_id->goods\_id

### 3.3 数据库表的详细清单

对象 account @logisticsPlatform ...									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名	类型	长度	小数点	不是 null	虚拟	键	注释		
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键		
login_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户主键		
money	float	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		账户余额		
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除		
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记		
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间		
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间		

表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型									
对象 car_type @logisticsPlatform...									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名	类型	长度	小数点	不是 null	虚拟	键	注释		
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键		
car_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		类型编号		
name	varchar	40	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		类型名称		
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除		
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记		
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间		
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间		

建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型									
对象 customer @logisticsPlatform...									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名	类型	长度	小数点	不是 null	虚拟	键	注释		
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键		
login_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户主键		
customer_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货主id		
name	varchar	40	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		姓名		
gender	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		性别		
telephone	varchar	13	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		电话		
user_goods	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		常运货物		
address	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		住址		
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除		
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记		
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间		
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间		

新建查询

表

视图

函数

事件

用户

查询

报表

备份

自动运行

模型

对象

driver @logisticsPlatform (8...

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键
login_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户主键
driver_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		司机id
name	varchar	40	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		姓名
gender	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		性别
telephone	varchar	13	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		电话
car_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		车辆类型编号
car_number	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		车牌号
trans_type	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		运输类型
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

查询	表	视图	函数	事件	用户	查询	报表	备份	自动运行	模型
10	对象 driver_intention @logisticsP...									
manage	保存 添加字段 插入字段 删除字段 主键 上移 下移									
tion_schema	字段 索引 外键 触发器 选项 注释 SQL 预览									
sPlatform										
	名	类型	长度	小数点	不是 null	虚拟	键	注释		
	id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键		
	intention_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		意向货物编号		
	good_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货物编号		
	driver_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		司机id		
	customer_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货主id		
	fees	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		运费报价		
	remark	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		备注		
ance_schema	is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除		
	note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记		
anFu	create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间		
o	update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间		

name  
n\_schema  
iform

对象 goods @logisticsPlatform (...)

保存 添加字段 插入字段 删除字段 主键 上移 下移

字段 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键
customer_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货主id
good_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货物编号
name	varchar	40	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货物名称
type_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货物类型编号
start	varchar	50	0	<input type="checkbox"/>	<input type="checkbox"/>		起点
destination	varchar	50	0	<input type="checkbox"/>	<input type="checkbox"/>		终点
fees_min	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		运费要求-最低
fees_max	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		运费要求-最高
remark	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		备注
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

ce\_schema

u

I0

</



表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型										
对象 trading_record @logisticsPl...										
保存 添加字段 插入字段 删除字段 主键 上移 下移										
字段	索引	外键	触发器	选项	注释	SQL 预览				
名	类型	长度	小数点	不是 null	虚拟	键	注释			
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键			
trade_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		交易编号			
order_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		订单编号			
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除			
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记			
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间			
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间			

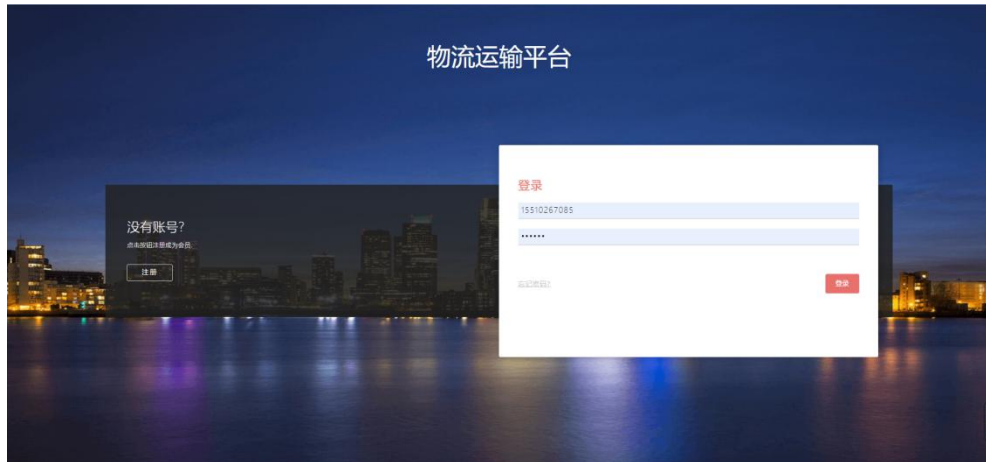
  

对象 message @logisticsPlatform...										
保存 添加字段 插入字段 删除字段 主键 上移 下移										
字段	索引	外键	触发器	选项	注释	SQL 预览				
名	类型	长度	小数点	不是 null	虚拟	键	注释			
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	自增主键			
message_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		消息编号			
login_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户主键			
good_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		货物主键			
from_user_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		来源用户id			
status	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		消息类型 0未知 1.有司机提出报价, 2.货物已到达 3.报			
is_delete	tinyint	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否删除			
note	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		笔记			
create_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间			
update_time	datetime	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间			

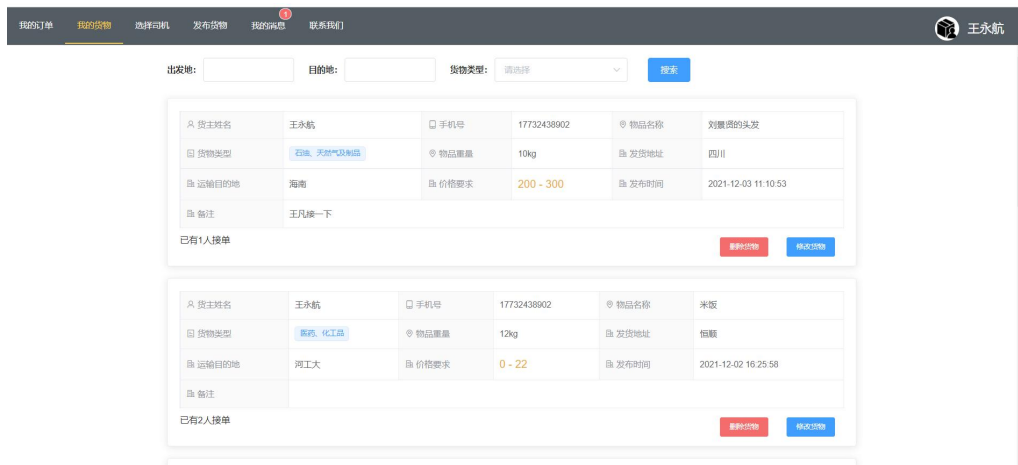
# 系统实现及测试报告

## 1 系统界面

(系统主要界面及界面风格说明。)



## 用户登录界面



## 货主货物界面

货物名称

货物类别

请选择货物类别

货物重量

1

kg

运输起点

运输终点

价格区间

0

~

¥

备注

立即发布

## 货主发布货物

基础设置

修改密码

个人账户

系统设置

更改头像

账号:

15510267085

姓名:

王永航

性别:

男

电话:

17732438902

家庭住址:

河工大

常运货物

类型:

电子产品

石油、天然气及制品

农副产品、食品及饮品

矿石、矿建材料

全部订单

运输中订单

等待签收

已完成订单

物品名称	吃的	货物类型	运输设备及零配件	发货地址	定州
运输目的地	河工大	发布时间	2021-12-03 22:06:38	司机姓名	芮振瑞
司机车辆类型	轻型普通货车	司机车牌号	冀HJ1368	司机备注	
司机报价	¥88	订单状态	已到达目的地		

确认收货

物品名称	蔬菜	货物类型	农副产品、食品及饮品	发货地址	天津
运输目的地	北京	发布时间	2021-12-03 11:57:21	司机姓名	芮振瑞
司机车辆类型	轻型普通货车	司机车牌号	冀HJ1368	司机备注	
司机报价	¥1	订单状态	运输中		

物品名称	米饭	货物类型	医药、化工品	发货地址	恒顺
------	----	------	--------	------	----

## 货主查看订单

## 货主个人信息

基础设置

修改密码

个人账户

系统设置

¥12

充值

---交易记录---

日期	来源	交易类型	订单编号	金额
2021-12-03 22:06:55	芮振瑞	订单付款	order-e144c5c	-88
2021-12-03 11:57:34	芮振瑞	订单付款	order-41fd19c	-1
2021-12-03 10:11:37	王永航	订单付款	order-c66aebc	-100
2021-12-03 09:01:35	王永航	订单付款	order-19755b	-99
2021-12-02 20:11:07	王凡	订单付款	order-ee16e0	-111

共 10 条 5条/页 < 1 2 > 前往 1 页

## 货主账户及交易记录

## ---我的消息---

一键已读

您关于货物 一些好东西 的运费已到账。

2021-12-04 13:24:39

点击查看

您对货物 一些好东西 的报价已被货主 王超 接受

2021-12-04 13:24:09

点击查看

您对货物 成鑫 的报价已被货主 王永航 接受

2021-12-03 22:00:55

点击查看

您关于货物 盖茨 的运费已到账。

2021-12-03 11:59:48

点击查看

您对货物 盖茨 的报价已被货主 王永航 接受

2021-12-03 11:57:34

点击查看

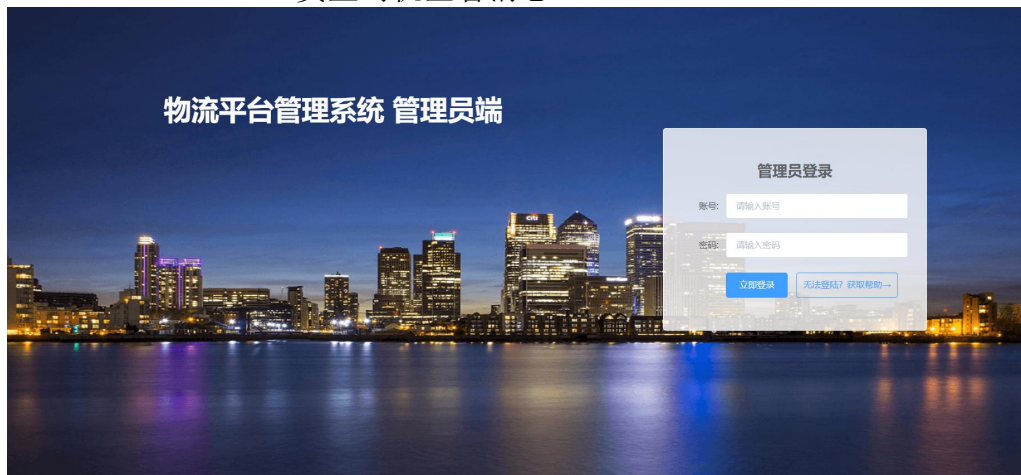
共 5 条

10条/页

< 1 >

前往 1 页

## 货主司机查看消息



## 管理员登录界面

管理员04512

账户管理

货物管理

订单管理

交易记录管理

系统设置

账户管理 / 账户列表

姓名:  查询

账号	姓名	用户类型	账户余额	上次登录时间	操作
175200277128	王永航	司机	2	2021-12-03T11:15:19	删除账号
12345678911	王凡	司机	1554	2021-12-02T22:02:28	删除账号
13146897978	芮振强	司机	56140	2021-12-03T23:02:06	删除账号
13146895958	王凡	司机	0	2021-12-04T13:45:34	删除账号
13333333333	王凡l	司机	50	2021-12-04T13:46:01	删除账号
15510267085	王永航	货主	12	2021-12-04T14:48:29	删除账号
17320027713	刘景	货主	12	2021-11-27T14:22:16	删除账号
18888888888	李嘉熙	货主	7777	2021-12-03T11:15:34	删除账号
18398758845	王超	货主	485	2021-12-04T13:19:22	删除账号

## 管理员管理账户等信息

### 2 系统实现模块说明

物流运输平台共设有货主、司机、系统管理员三种角色，所有游客都可进入网站，查看网站的基本介绍与信息，了解网站基本功能，并能进行注册。

对于货主来说，主要是注册、登录、修改个人信息、发布货物信息、选择司机、查看订单状态、确认收货、支付订单等模块。

对于司机来说，主要是注册、登录、修改个人信息、查看、筛选货物信息、选择运单、修改运单状态等模块。

对于管理员来说主要是用户管理、货物管理和订单管理等模块。

编号	功能名称	功能描述
1	注册模块	获得系统认证身份
2	登录模块	已有系统认证身份的用户登陆系统
3	更改个人信息模块	更改用户基本信息
4	发布货物信息模块	发布货物信息
5	选择司机模块	选择司机
6	支付订单模块	支付订单运费
7	查看货物信息模块	查看货物信息列表
8	选择运单模块	选择货物进行承运
9	修改运单状态模块	修改运单状态
10	查看订单状态	查看运单状态

11	确认收货	点击确认收货
12	用户管理	增加修改删除用户
13	查看消息	货主和司机查看消息
14	统计数据	货主和司机统计一段时间内完成的订单
15	查看交易记录	货主和司机查看自己的交易记录
16	货物管理	删除货物
17	订单管理	删除订单

## 2.1 模块 1（模块名称）

### 登录注册模块

#### 2.1.1 描述

主要功能：用户通过该模块可以实现账户的注册登录功能，注册后，可以进行登录，登录后，用户可以进入相应的系统界面，进行相关功能操作。

处理的数据：注册时：用户输入姓名，电话，密码以及注册类型

产生的结果：若数据合理且没有重复手机号的情况下注册成功

登录时：用户输入用户名、登录密码

产生的结果：经过系统审核等操作，提示用户登录成功信息。

#### 2.1.2 主要代码

前端：<!DOCTYPE html>

<html lang="en" >

<head>

```

<meta charset="UTF-8">
<title>物流运输平台</title>
  <link rel="stylesheet" href="../css/reset.min.css">
  <link rel="stylesheet" href="../css/login.css">
  <link rel="stylesheet"
href="../css/element-ui@2.13.0-theme-chalk-index.css">
</head>
<body>
<div id="app">
  <h1 style="text-align: center;color: white;font-size: 50px;padding-top:
100px">物流运输平台</h1>
  <section class="user">
    <div class="user_options-container">
      <div class="user_options-text">
        <div class="user_options-unregistered">
          <h2 class="user_unregistered-title">没有账号? </h2>
          <p class="user_unregistered-text">点击按钮注册成为会员.</p>
          <button class="user_unregistered-signup" id="signup-button"
@click="turnLeft()">注册</button>
        </div>

        <div class="user_options-registered">
          <h2 class="user_registered-title">已有账号?</h2>
          <p class="user_registered-text">点击按钮会员登录.</p>
          <button class="user_registered-login" id="login-button"
@click="turnRight()">登录</button>
        </div>
      </div>

      <div class="user_options-forms" id="user_options-forms">
        <div class="user_forms-login">
          <h2 class="forms_title">登录</h2>

```



```

    <form class="forms_form">
      <fieldset class="forms_fieldset">
        <div class="forms_field">
          <input v-model="loginForm.telephone" placeholder="手机
号" class="forms_field-input" />
        </div>
        <div class="forms_field">
          <input v-model="loginForm.password" type="password"
placeholder="密码" class="forms_field-input"/>
        </div>
      </fieldset>
      <div class="forms_buttons" style="margin-top: 60px">
        <button @click="getHelp()" type="button"
class="forms_buttons-forgot">忘记密码?</button>
        <el-button type="primary" @click="login()"
style="background-color: #E8716D;border: 1px solid #E8716D">登录
</el-button>
      </div>
    </form>
  </div>
  <div class="user_forms-signup">
    <h2 class="forms_title">注册</h2>
    <form class="forms_form">
      <fieldset class="forms_fieldset">
        <div class="forms_field">
          <label>司机: </label>
          <input v-model="registerForm.type" type="radio"
name="optionsRadios" value="1" >
          <label>货主: </label>
          <input v-model="registerForm.type" type="radio"
name="optionsRadios" value="0" >
        </div>

```

```

        <div class="forms_field">
            <input v-model="registerForm.name" type="text"
placeholder="姓名" class="forms_field-input"/>
        </div>
        <div class="forms_field">
            <input v-model="registerForm.telephone" placeholder="
手机号" class="forms_field-input"/>
        </div>
        <div class="forms_field">
            <input v-model="registerForm.password"
type="password" placeholder="密码" class="forms_field-input"/>
        </div>
        <div class="forms_field">
            <input v-model="registerForm.confirm" type="password"
placeholder="确认密码" class="forms_field-input"/>
        </div>
    </fieldset>
    <div class="forms_buttons" >
        <el-button type="primary" @click="register()"
style="background-color: #E8716D;border: 1px solid #E8716D">注册
    </el-button>
    </div>
</form>
</div>
</div>
</div>
</section>
</div>
<script src="../js/jquery-3.5.1.min.js"></script>
<script src="../js/vue-v2.6.11.js"></script>
<script src="../js/element-ui@2.13.0-index.js"></script>
<script src="../js/common.js"></script>

```

```

<script src="../../js/apiurls.js"></script>
<script>
let apps = new Vue({
  el: '#app',
  data: {
    loginForm: {
      telephone: "",
      password: "",
    },
    registerForm: {
      type: 0,
      name: "",
      telephone: "",
      password: "",
      confirm: ""
    },
  },
  methods: {
    getHelp() {
      this.$message({
        message: '请联系 234818290 获取帮助',
        type: 'info'
      });
    },
    //登录验证
    login(obj) {
      let that = this;//fetch 中 this 指针改变，用 that 保留 vue 对象
      //判空
      if (!this.loginForm.telephone || !this.loginForm.password) {
        this.$message({
          message: '请输入手机号或密码',
          type: 'warning'
        });
      }
    }
  }
});

```

```

    });
    return;
  }else {
    let data = {
      loginId: that.loginForm.telephone,
      password: that.loginForm.password
    }
    getPostData(apiUrls.login_user.login,data,'post',function (data) {
      if(data.msg === 'success') {
        that.$message({
          message: "登录成功",
          type: "success"
        })
        localStorage.setItem('token',data.data.token);
        if (data.data.userType === 0) {
          setTimeout(function (){
            window.location.href = "homePage-customer.html"
          },800)
        } else {
          setTimeout(function (){
            window.location.href = "homePage.html"
          },300)
        }
      } else {
        that.$message({
          message: data.msg,
          type: "error"
        })
      }
    })
  }
},

```

```

register() {
  let that = this;//fetch 中 this 指针改变, 用 that 保留 vue 对象
  //条件限制
  if (!this.registerForm.name || !this.registerForm.telephone
|| !this.registerForm.password || !this.registerForm.confirm) {
    this.$message({
      message: '请输入完整信息',
      type: 'warning'
    });
    return;
  }else if(this.registerForm.telephone.length != 11){
    this.$message({
      message: '请输入正确的手机号',
      type: 'warning'
    });
  }else if(this.registerForm.password.length < 6){
    this.$message({
      message: '密码至少为六位',
      type: 'warning'
    });
  }else if(this.registerForm.password != this.registerForm.confirm){
    this.$message({
      message: '两次密码输入不一致',
      type: 'warning'
    });
  }else {
    let data = {
      type: that.registerForm.type,
      name: that.registerForm.name,
      loginId: that.registerForm.telephone,
      password: that.registerForm.password,
      password1: that.registerForm.confirm
    }
  }
}

```

```

    }
    postData(apiUrls.login_user.register,data,'post',function (data)
{
    if(data.msg === 'success') {
        that.$message({
            message: "注册成功",
            type: "success"
        })
        setTimeout(function (){
            window.location.reload()
        },300)
    } else {
        that.$message({
            message: data.msg,
            type: "error"
        })
    }
})
},
turnLeft() {
    const userForms =
document.getElementById('user_options-forms')
    userForms.classList.remove('bounceRight')
    userForms.classList.add('bounceLeft')
},
turnRight() {
    const userForms =
document.getElementById('user_options-forms')
    userForms.classList.remove('bounceLeft')
    userForms.classList.add('bounceRight')
}
}

```

```

    },
  })
</script>
</body>
</html>

```

后端:

```
package com.example.demo.management.controller;
```

```

import com.baomidou.mybatisplus.core.metadata.IPage;
import com.example.demo.common.ApiResponse;
import com.example.demo.common.annotation.AccessToken;
import com.example.demo.common.exception.CustomException;
import com.example.demo.management.entity.LoginUser;
import
com.example.demo.management.entity.request.ChangePasswordRequest;
import com.example.demo.management.entity.request.LoginRequest;
import com.example.demo.management.entity.request.RegisterRequest;
import
com.example.demo.management.entity.response.AllAccountResponse;
import
com.example.demo.management.entity.response.LoginInformationResponse;
import com.example.demo.management.service.LoginUserService;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import io.swagger.annotations.ApiParam;
import io.swagger.models.auth.In;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.*;
import springfox.documentation.annotations.ApiIgnore;

```

```

import java.util.Map;

/**
 * <p>
 * 用户登录表 前端控制器
 * </p>
 *
 * @author LiuJingxian
 * @since 2021-11-15
 */
@Api(tags = "登录注册相关")
@CrossOrigin
@RestController
@RequestMapping("/management/login-user")
public class LoginUserController {

    @Autowired
    LoginUserService loginUserService;

    @PostMapping("/register")
    @ApiOperation("注册账号")
    public ApiResponse register(
        @RequestBody RegisterRequest registerRequest) {
        loginUserService.Register(registerRequest);
        return ApiResponse.success();
    }

    @PostMapping("/delUser")
    @ApiOperation(value = "删除账户")
    public ApiResponse<Object> delCount(

```



```

        @ApiParam(name = "loginId") @RequestBody Map<String,
Object> mp
    ) {
        String loingId = mp.get("loginId").toString();
        try {
            loginUserService.delUser(loingId);
        } catch (CustomException e) {
            return ApiResponse.error(e.getMsg());
        }
        return ApiResponse.success();
    }

```

```

    @PostMapping("/login")
    @ApiOperation(value = "用户登录")
    public ApiResponse<LoginInformationResponse> login(
        @Validated @RequestBody LoginRequest loginRequest
    ) {
        LoginInformationResponse loginInformationResponse =
loginUserService.login(loginRequest);
        return ApiResponse.success(loginInformationResponse);
    }

```

```

public void Register(RegisterRequest registerRequest) {
    int count = loginUserMapper.selectCount(
        Wrappers.lambdaQuery(LoginUser.class)
            .eq(LoginUser::getLoginId,registerRequest.getLoginId())
    )
    );
    if (count > 0) {
        throw new CustomException(1, "该账号已存在");
    }
}

```

```

    }
    if
(!registerRequest.getPassword().equals(registerRequest.getPassword1())) {
        throw new CustomException(1, "两次输入的密码不一致");
    }
    if (registerRequest.getPassword().length() < Common.MAXLENGTH) {
        throw new CustomException(1, "密码长度不得少于 6 位");
    }

    LoginUser loginUser = new LoginUser();
    Driver driver = new Driver();
    Customer customer = new Customer();
    Account account = new Account();

    //赋值
    loginUser.setLoginId(registerRequest.getLoginId());
    loginUser.setPassword(registerRequest.getPassword());
    loginUser.setUserType(registerRequest.getType());

    driver.setLoginId(registerRequest.getLoginId());
    driver.setDriverId(registerRequest.getLoginId());
    driver.setTelephone(registerRequest.getLoginId());
    driver.setName(registerRequest.getName());

    customer.setLoginId(registerRequest.getLoginId());
    customer.setCustomerId(registerRequest.getLoginId());
    customer.setTelephone(registerRequest.getLoginId());
    customer.setName(registerRequest.getName());

    account.setLoginId(registerRequest.getLoginId());
    /**
     * getType == 0: 货主

```

```

        * getType == 1: 司机
        */
    if (registerRequest.getType() == 0) {
        loginUserMapper.insert(loginUser);
        customerMapper.insert(customer);
        accountMapper.insert(account);
    } else {
        loginUserMapper.insert(loginUser);
        driverMapper.insert(driver);
        accountMapper.insert(account);
    }
}

```

@Override

```

public LoginInformationResponse login(LoginRequest loginRequest) {
    LoginUser loginUser =
loginUserMapper.selectOne(Wrappers.lambdaQuery(LoginUser.class)
        .eq(LoginUser::getLoginId,loginRequest.getLoginId()));
    if (loginUser == null) {
        throw new CustomException("不存在此用户");
    }
    if (!loginUser.getPassword().equals(loginRequest.getPassword())) {
        throw new CustomException("密码错误");
    }
}

```

```

        loginUser.setToken(Prefix.TOKEN_PREFIX + UUID.randomUUID());
        loginUserMapper.updateById (loginUser);
        LoginInformationResponse loginInformationResponse = new
LoginInformationResponse();
        BeanUtils.copyProperties(loginUser, loginInformationResponse);
        return loginInformationResponse;
    }
}

```

```
}
```

## 2.2 模块 2

### 货主查看订单

#### 2.2.1 描述

主要功能：货主通过该模块可以查看订单状态，从而实现确认收货，查看历史订单的操作

处理的数据：该用户的订单数据

产生的结果：用户确认收货，查看历史订单

#### 2.2.2 主要代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>货主订单</title>
  <link rel="stylesheet"
href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">
  <link rel="stylesheet" href="../css/style.css">
  <link rel="stylesheet" href="../css/bootstrap/bootstrap.min.css">
</head>
<body>
<div id="app">
  <div class="container">
    <div class="container">
      <template>
        <el-tabs v-model="activeName" type="card"
@tab-click="handleClick" class="tabs">
          <el-tab-pane label="全部订单" name="1"
width="200px"></el-tab-pane>
          <el-tab-pane label="运输中订单"
```

```

name="2"></el-tab-pane>
      <el-tab-pane label="等待签收"
name="3"></el-tab-pane>
      <el-tab-pane label="已完成订单"
name="4"></el-tab-pane>
    </el-tabs>
  </template>
  <div class="list" v-for="item in goodsInf">
    <el-card class="box-card">
      <el-descriptions class="margin-top" :column="3"
border>
        <el-descriptions-item>
          <template slot="label">
            <i class="el-icon-location-outline"></i>
            物品名称
          </template>
          {{item.goodName}}
        </el-descriptions-item>
        <el-descriptions-item>
          <template slot="label">
            <i class="el-icon-tickets"></i>
            货物类型
          </template>
          <el-tag
size="small">{{item.typeName}}</el-tag>
        </el-descriptions-item>
        <el-descriptions-item>
          <template slot="label">
            <i class="el-icon-office-building"></i>
            发货地址
          </template>
          {{item.start}}

```

```

</el-descriptions-item>
<el-descriptions-item>
  <template slot="label">
    <i class="el-icon-office-building"> </i>
    运输目的地
  </template>
  {{item.destination}}
</el-descriptions-item>
<el-descriptions-item>
  <template slot="label">
    <i class="el-icon-office-building"> </i>
    发布时间
  </template>
  {{item.driverCreateTime1}}
</el-descriptions-item>
<el-descriptions-item>
  <template slot="label">
    <i class="el-icon-office-building"> </i>
    司机姓名
  </template>
  {{item.driverName}}
</el-descriptions-item>
<el-descriptions-item>
  <template slot="label">
    <i class="el-icon-office-building"> </i>
    司机车辆类型
  </template>
  {{item.carName}}
</el-descriptions-item>
<el-descriptions-item>
  <template slot="label">
    <i class="el-icon-office-building"> </i>

```

```

        司机车牌号
    </template>
    {{item.carNumber}}
</el-descriptions-item>
<el-descriptions-item>
    <template slot="label">
        <i class="el-icon-office-building"> </i>
        司机备注
    </template>
    {{item.driverRemark}}
</el-descriptions-item>
<el-descriptions-item>
    <template slot="label">
        <i class="el-icon-office-building"> </i>
        司机报价
    </template>
    <span style="color: #EAAE3E;font-size:
18px"> ¥ {{item.fees}}</span>
</el-descriptions-item>
<el-descriptions-item>
    <template slot="label">
        <i class="el-icon-office-building"> </i>
        订单状态
    </template>
    <el-tag effect="plain"
size="medium" :type="item.tag_type">{{item.tag_info}}</el-tag>
</el-descriptions-item>
</el-descriptions>
    <el-button type="primary" class="button_click
pull-right" @click="setConfirm(item.orderId)"
v-if="item.isButtonDisabled">确认收货</el-button>
</el-card>

```

```

</div>
<el-pagination
  @size-change="handleSizeChange"
  @current-change="handleCurrentChange"
  background
  :current-page= "page.current"
  :page-sizes="[5, 10, 20, 40]"
  :page-size="page.size"
  layout = "total, sizes, prev, pager, next, jumper"
  :total="page.total"
  style="margin-bottom: 30px">
</el-pagination>
</div>
</div>

```

```

</div>
<script src="../js/jquery-3.5.1.min.js"></script>
<script src="../js/vue-v2.6.11.js"></script>
<script src="https://unpkg.com/element-ui/lib/index.js"></script>
<script src="../js/common.js"></script>
<script src="../js/apiurls.js"></script>
<script>
  const vm = new Vue({
    el:'#app',
    data:{
      activeName: '1',
      tag_type:"",
      tag_info:"",
      goodsInf: [],
      tab: "",
      page: {
        //当前页

```



```

        current: 1,
        //最大页数
        total: "",
        size: 10
    }
},
methods: {
    handleClick(tab, event) {
        this.tab = tab.name
        this.getOrders(tab.name)
    },
    getOrders(status) {
        let that = this
        if (status === null) {
            status = 1
        }
        let data = {
            "pageRequest": {
                "current": 1,
                "size": 10
            },
            status: status
        }
    }
}

```

```

getPostData(apiUrls.orders.selectCustomerOrders,data,'post',function (data)
{
    if (data.msg === 'success') {
        that.goodsInf = data.data.records
        that.page.total = data.data.total
        //遍历设置标签状态和按钮
        that.goodsInf.forEach((good,index) => {
            console.log(good)
        })
    }
})

```

```

switch (good.status) {
    case 0: {
        that.goodsInf[index].tag_type =
'primary'

        that.goodsInf[index].tag_info = '正在
运输'

        that.goodsInf[index].isButtonDisabled
= false;

        break
    }
    case 1: {
        that.goodsInf[index].tag_type =
'warning'

        that.goodsInf[index].tag_info = '已到
达目的地'

        that.goodsInf[index].isButtonDisabled
= true;

        break
    }
    case 2: {
        that.goodsInf[index].tag_type =
'success'

        that.goodsInf[index].tag_info = '运输
完成'

        that.goodsInf[index].isButtonDisabled
= false;

        break
    }
}
})
} else {
    this.$message({

```

```

        message: data.msg(),
        type: 'error'
    });
    }
    })
},
setConfirm(orderId) {
    let that = this
    that.$confirm('您确定确认到达吗?', '提示', {
        confirmButtonText: '确定',
        cancelButtonText: '取消',
        type: 'warning'
    }).then(() => {
        let data = {
            "orderId": orderId,
            status: 2
        }
        postData(apiUrls.orders.modifyOrders, data, 'post',
function (result) {
        if (result.msg === 'success') {
            that.$message({
                message: "修改成功",
                type: 'success'
            });
            that.getOrders(that.tab)
        } else {
            that.$message({
                message: data.msg,
                type: 'error'
            });
        }
    })
}

```

```

        })
    },

    // 初始页 currentPage、初始每页数据数 pagesize 和数据 data
    handleSizeChange: function (size) {
        this.page.size = size;
        console.log(this.page.size) //每页下拉显示数据
        this.getGoods()
    },
    handleCurrentChange: function(currentPage){
        this.page.current = currentPage;
        console.log(this.page.current) //点击第几页
        this.getGoods()
    },
},
mounted() {
    this.getOrders()
}
})
</script>
</body>
</html>

```

后端：

```
package com.example.demo.management.controller;
```

```

import com.baomidou.mybatisplus.core.metadata.IPage;
import com.example.demo.common.ApiResponse;
import com.example.demo.common.annotation.AccessToken;
import com.example.demo.management.entity.LoginUser;
import com.example.demo.management.entity.request.*;
import

```

```

com.example.demo.management.entity.response.SelectIttentionListResponse;
import
com.example.demo.management.entity.response.SelectOrdersResponse;
import com.example.demo.management.service.DriverIntentionService;
import com.example.demo.management.service.OrdersService;
import com.example.demo.management.service.impl.OrdersServiceImpl;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import io.swagger.annotations.ApiParam;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.*;

import springfox.documentation.annotations.ApiIgnore;

import javax.annotation.Resource;

/**
 * <p>
 * 订单信息表 前端控制器
 * </p>
 *
 * @author LiuJingxian
 * @since 2021-11-15
 */
@AccessToken
@Api(tags = "订单相关")
@CrossOrigin
@RestController
@RequestMapping("/management/orders")
public class OrdersController {

```

```

@Resource
OrdersService ordersService;

@ApiOperation("添加订单")
@PostMapping("/addOrders")
public ApiResponse<Object> addOrders(
    @ApiIgnore @RequestAttribute(name = "loginUser")
LoginUser loginUser,
    @Validated @RequestBody AddOrderRequest
addOrderRequest
) {
    ordersService.addOrders(addOrderRequest,loginUser.getLoginId());
    return ApiResponse.success();
}

@PostMapping("/delOrders")
@ApiOperation("删除订单")
public ApiResponse<Object> delOrders(
    @RequestBody String orderId
) {
    ordersService.deleteOrders(orderId);
    return ApiResponse.success();
}

@PostMapping("/modifyOrders")
@ApiOperation("修改订单")
public ApiResponse<Object> modifyOrders(
    @ApiIgnore @RequestAttribute(name = "loginUser")
LoginUser loginUser,
    @Validated @RequestBody AddOrderRequest
addOrderRequest
) {

```

```
ordersService.modifyOrders(addOrderRequest,loginUser.getLoginId());
    return ApiResponse.success();
}
```

```
@PostMapping("/checkOrders")
@ApiOperation("查看货物是否已经生成订单")
public ApiResponse<Object> checkOrders(
    @RequestBody String goodId
){
    ordersService.checkOrders(goodId);
    return ApiResponse.success();
}
```

```
@PostMapping("/selectDriverOrders")
@ApiOperation("司机查看自己的订单")
public ApiResponse<IPage<SelectOrdersResponse>>
selectDriverOrders(
    @ApiIgnore @RequestAttribute(name = "loginUser")
    LoginUser loginUser,
    @RequestBody SelectOrdersRequest selectOrdersRequest
){
    IPage<SelectOrdersResponse> page =
ordersService.selectDriverOrders(selectOrdersRequest,
loginUser.getLoginId());
    return ApiResponse.success(page);
}
```

```
@PostMapping("/selectCustomerOrders")
@ApiOperation("货主查看自己的订单")
public ApiResponse<IPage<SelectOrdersResponse>>
selectCustomerOrders(
```

```

        @ApiIgnore @RequestAttribute(name = "loginUser")
LoginUser loginUser,
        @RequestBody SelectOrdersRequest selectOrdersRequest
    ){
        IPage<SelectOrdersResponse> page =
ordersService.selectCustomerOrders(selectOrdersRequest,
loginUser.getLoginId());
        return ApiResponse.success(page);
    }

    @PostMapping("/selectAllOrders")
    @ApiOperation("查看全部订单")
    public ApiResponse<IPage<SelectOrdersResponse>> selectAllOrders(
        @ApiIgnore @RequestAttribute(name = "loginUser")
LoginUser loginUser,
        @RequestBody SelectOrdersRequest selectOrdersRequest
    ){
        IPage<SelectOrdersResponse> page =
ordersService.selectAllOrders(selectOrdersRequest);
        return ApiResponse.success(page);
    }
}

```

### 3. 系统测试

首先进行模块测试，通过白盒方式和黑盒方式进行测试。白盒测试在 3.1 部分，黑盒测试在 3.3 部分。

#### 3.1 模块测试

##### 3.1.1 登录模块：



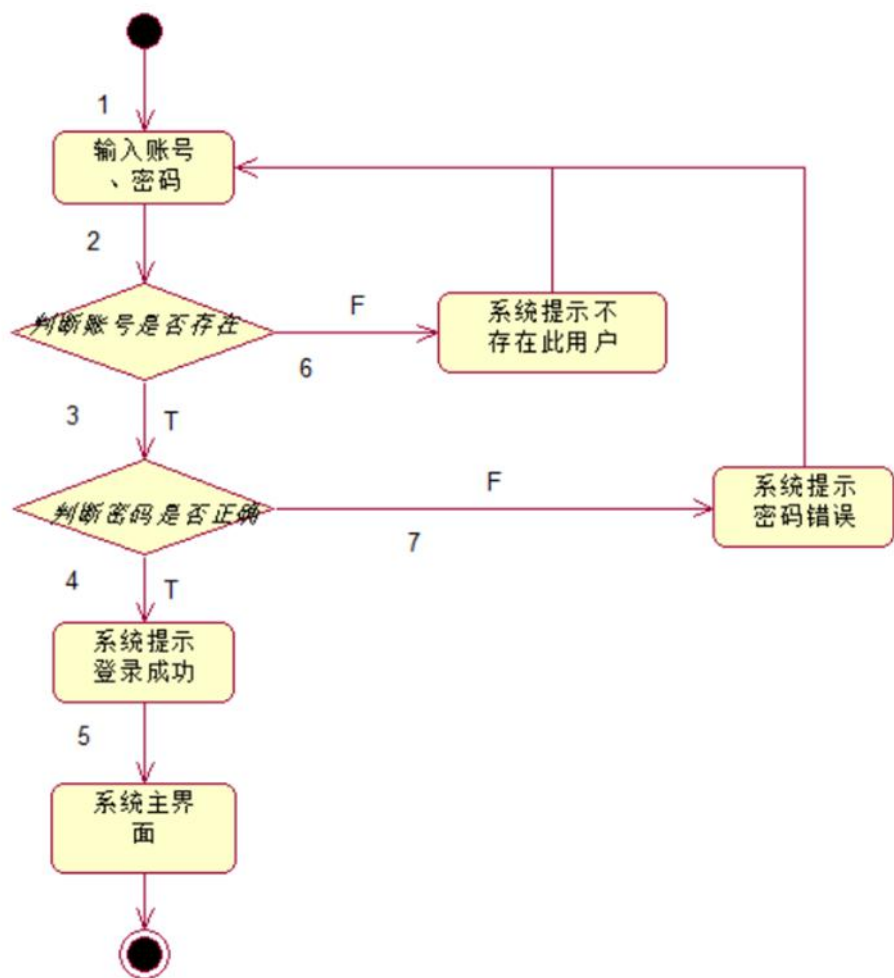


表 3-2 登录模块测试记录

编号	路径标识	输入	期望输出	输出内容	发现问题	测试结果	测试时间	测试人
1	1、2、3、4、5	15510267085, 111111	系统提示登陆成功, 进入主系统	系统提示登陆成功, 进入主系统	无	√	12.4	王永航
2	1、2、		第一次	第一次				王永

	6、2、3、 4、5	17320027713, 111111  17320027713, 123456	系统提示密 码错误, 第二次 系统提示登 录成功, 进 入主系统	系统提示密 码错误, 第二次 系统提示登 录成功, 进 入主系统	无	√	12.4	航
3	1、2、 3、7、2、 3、4、5	17320027711, 123456  17320027713, 123456	第一次 系统提示不 存在此用 户, 第二次 系统提示登 录成功, 进 入主系统	第一次 系统提示不 存在此用 户, 第二次 系统提示登 录成功, 进 入主系统	无	√	12.4	王永航

### 3.1.2 发布货物模块

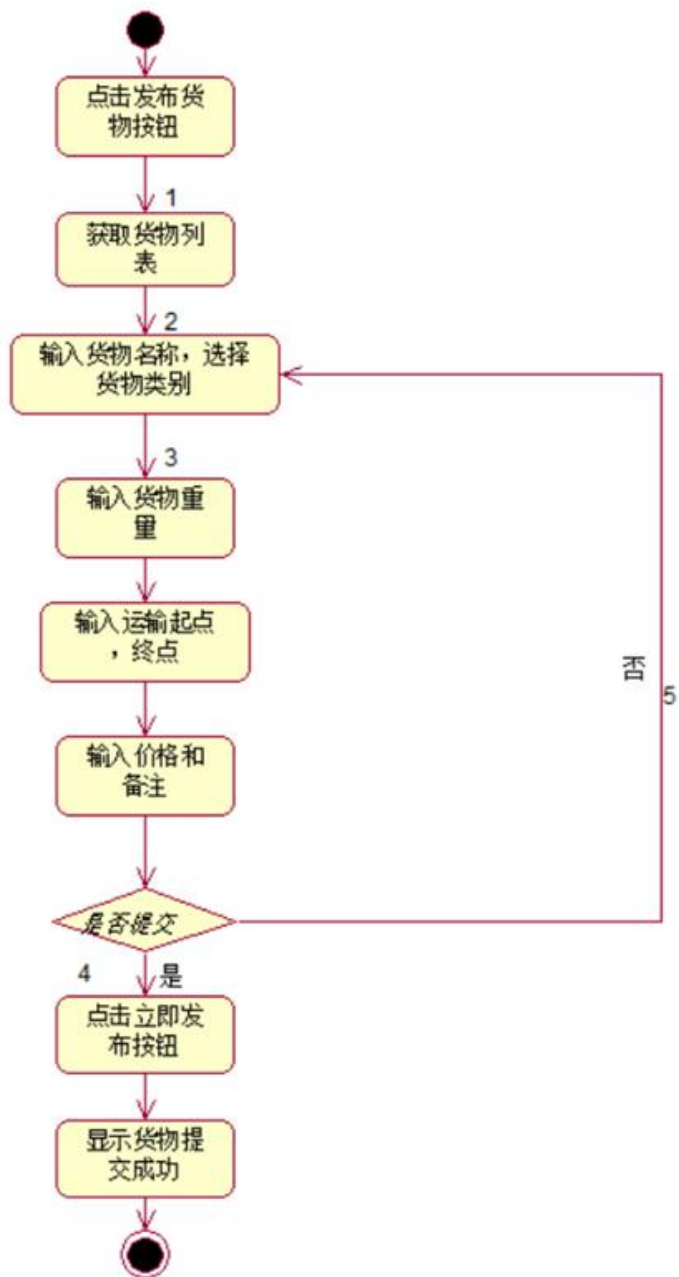


表 3-2 发布货物模块测试记录

编号	路径标识	输入	期望输出	输出内容	发现问题	测试结果	测试时间	测试人

1	1、2、 3、4、	零食，农副产品、食品及饮品，112，河工大，北京，0，123，	系统提示货物发布成功	系统提示货物发布成功	无	√	12.4	王凡
2	1、2、 3、5、3、 4	吃的，电子通讯产品，23/123，天津，北京，123	系统提示货物发布成功	系统提示货物发布成功	无	√	12.4	王凡

### 3.1.3 账号管理模块

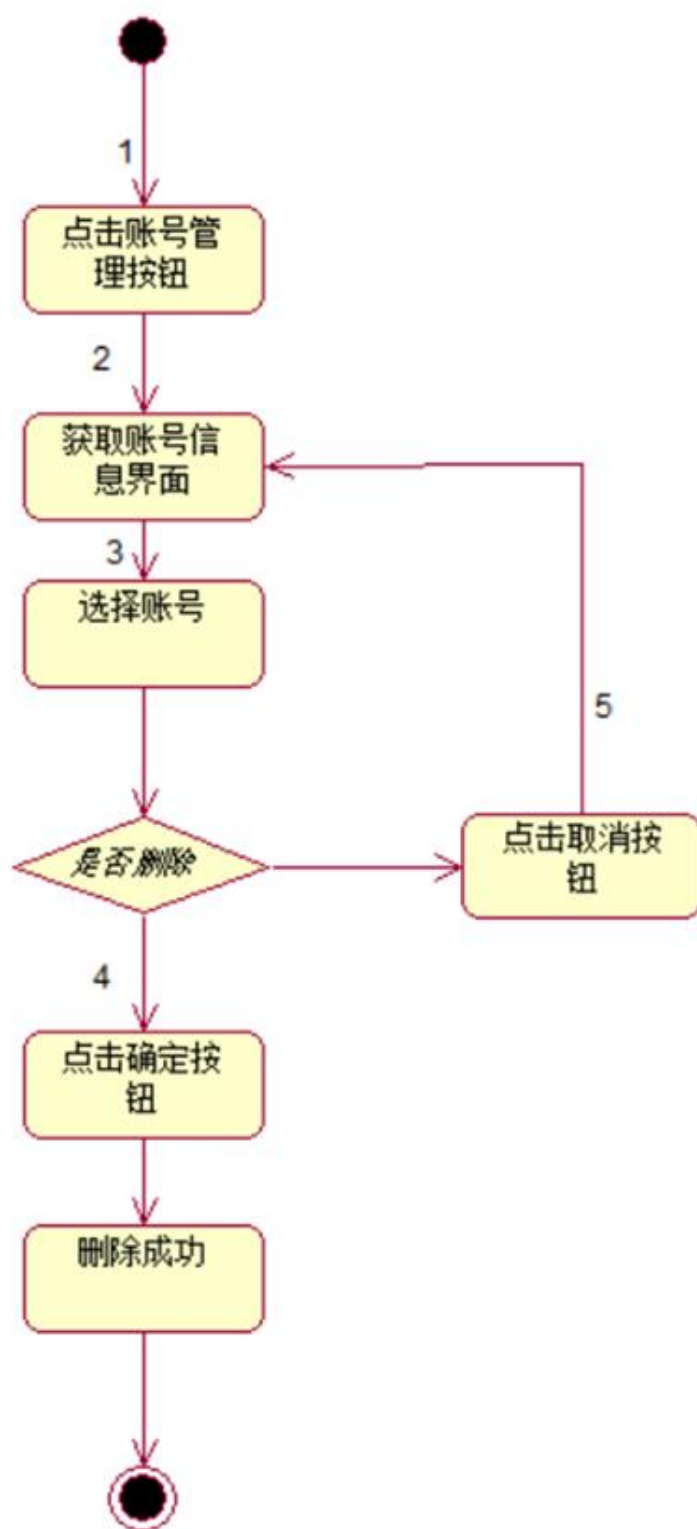


表 3-3 账号管理模块测试记录

编号	路径标识	输入	期望输出	输出内容	发现问题	测试结果	测试时间	测试人
1	1、2、3、4、	无	系统提示删除成功	系统提示删除成功	无	√	12.4	王凡
2	1、2、3、5、3、4	无	系统提示删除成功	系统提示删除成功	无	√	12.4	王凡

#### 3.1.4 运单浏览模块

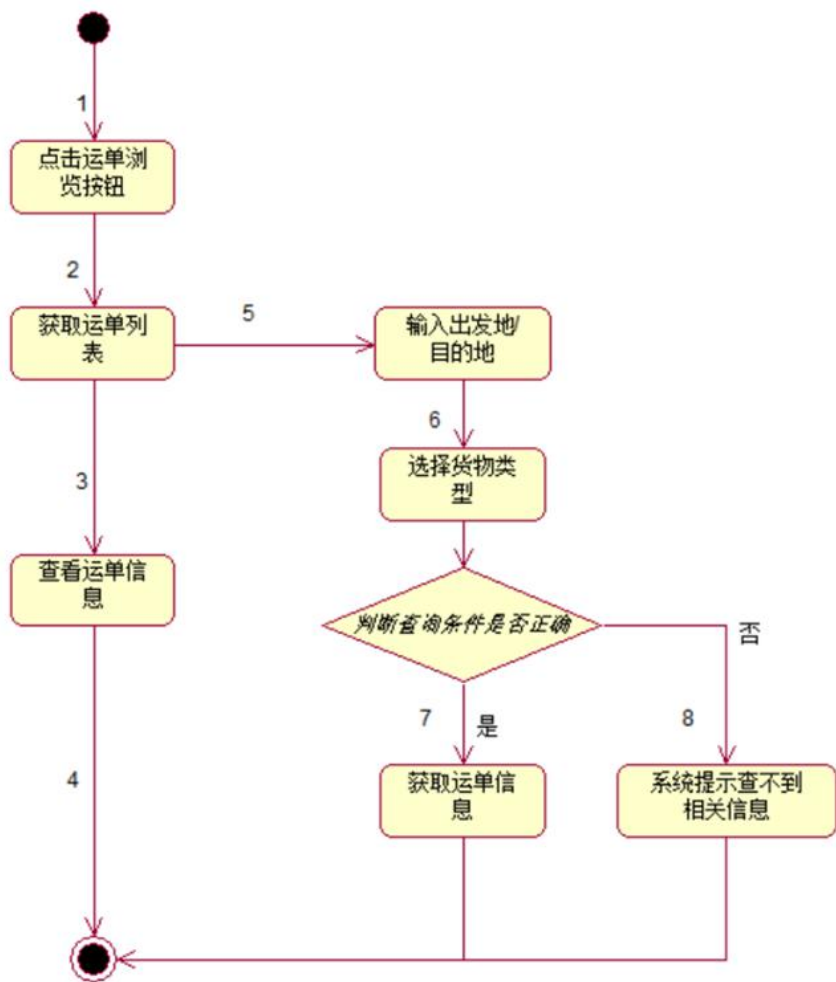


表 3-4 运单浏览模块测试记录

编号	路径标识	输入	期望输出	输出内容	发现问题	测试结果	测试时间	测试人
1	1、2、3、4	无	进入运单信息列表	运单信息列表	无	√	12.4	王超

2	1、2、5、6、7	河工大,黑龙江,零食	输出该运单信息	输出该运单信息	无	√	12.4	王超
3	1、2、5、6、8	河工大,北京,零食	系统提示查不到相关信息	系统提示查不到相关信息	无	√	12.4	王超

## 3.2 功能测试

### 3.2.1 系统功能需求

表 3-5 功能需求点列表

编 号	功 能 名 称	使 用 人	功 能 描 述	输 入 内 容	输 出 内 容
1	注册	货主或司机	选择身份并进行注册	用户点击“注册”按钮,填入所需信息进行注册	系统主界面
2	登录	货主、司机、系统管理员	正确输入账号密码进行登录	货主点击登录按钮,输入账号密码	系统主界面
3	完善个人信息	货主或司机	在登录后,完善个人的基本信息	用户点击“完善个人信息”,填入所需完善的个人信息	用户信息表
4	发布货物信息	货主	货主发布货物信息等待承运	货主点击发布货物信息,填入货物信息发布在网站上	货物信息表



5	查看货物信息	司机	司机查看网站上已发布的货物信息	司机点击“查看货物信息”	货物信息表
6	选择司机	货主	货主选择申请承运的司机	货主点击“查看司机”	司机信息表
7	修改运单状态	司机	司机实时更新货物承运状态	司机点击“修改运单状态”	订单表
8	查看运单状况	货主	货主获取实时运单状态	货主点击“查看运单状态”	订单表
9	确认收货	货主	货主确认收到货物	货主点击“确认收货”	订单表
10	管理账户	管理员	管理员管理账户	管理员点击“账户管理”	用户表
11	管理订单	管理员	管理员管理订单	管理员点击“订单管理”	订单表

### 3.2.2 功能测试报告

表 3-6 功能测试记录

编号	功能名称	输入内容	期望输出	输出内容	发现问题	测试结果	测试时间	测试人
----	------	------	------	------	------	------	------	-----

1	注册	个人信息，密码	获得相应的身份账号信息	获得相应的身份账号信息	无	√	12.4	王永航
2	登录	账号，密码	登录成功，进入系统主界面	登录成功，进入系统主界面	无	√	12.4	刘景贤
3	完善个人信息	手机号，车型，车牌号，运输类型	修改成功	修改成功	无	√	12.4	王永航
4	发布货物信息	货物名，货物类型，货物重量，起点，终点，价格区间，留言	发布成功	发布成功	无	√	12.4	王凡
5	查看货物信息	起点，终点，类型	查询货物信息	查货物信息	无	√	12.4	王超
6	选择司机	留言	选择成功	选择成功	无	√	12.4	刘景贤
7	修改运单状态	运单状态	修改成功	修改成功	无	√	12.4	王永航

8	查看 运单状况	货主 点击“查看 运单状态”	运单 状态	运单 状态	无	√	12.4	王 凡
9	确认 收货	货主 点击“确认 收货”	订单 界面	订单 界面	无	√	12.4	王 超
10	管理 账户	管理员 点击“账户 管理”	删除 成功	删除 成功	无	√	12.4	刘 景贤
11	管理 订单	管理员 点击“订单 管理”	删除 成功	删除 成功	无	√	12.4	王 永航

### 3.3. 其它测试（根据测试方案具体化）

#### 3.3.1 登录模块

表 3-7 等价类划分

输入数据	有效等价类	无效等价类
账号	①系统已有账号	②系统中没有的账号 ③空
密码	④正确的密码	⑥错误的密码 ⑦空

表 3-8 合理等价类测试用例

测试数据		期望结果	覆盖的等价类
账号	密码	登陆成功，进入系统主界面	①④
15510267085	111111		

表 3-9 不合理等价类测试用例

测试数据		期望结果	覆盖等价类
账号	密码	系统提示密码错误	①⑥
17320027713	111111		
17320027711	123456	系统提示不存在此用户	②
(空)	123456	系统提示请填写账号	③
17320027713	(空)	系统提示请填写密码	⑦

表 3-10 登录功能测试表

编号	功能名称	输入内容	期望结果	输出结果	发现问题	测试结果	测试时间	测试人
----	------	------	------	------	------	------	------	-----

1	登 录	15510267085, 111111	登 陆 成 功, 进 入 系 统 主 界 面	登 陆 成 功, 进 入 系 统 主 界 面	无	√	12.5	王 永 航
2	登 录	17320027713, 111111	系 统 提 示 密 码 错 误	系 统 提 示 密 码 错 误	无	√	12.5	王 永 航
3	登 录	17320027713, 123456	系 统 提 示 不 存 在 此 用 户	系 统 提 示 不 存 在 此 用 户	无	√	12.5	王 永 航
4	登 录	(空), 123456	系 统 提 示 请 填 写 账 号	系 统 提 示 请 填 写 账 号	无	√	12.5	王 永 航
5	登 录	17320027713, (空)	系 统 提 示 请 填 写 密 码		无	√	12.5	王 永 航

### 3.3.2 发布货物模块

表 3-11 等价类划分

输入数据	有效等价类	无效等价类
货物名称	①长度<40 的字符串	②长度>40 的字符串 ③空

货物类别	④长度<40 的字符串	⑤长度>40 的字符串 ⑥空
货物重量	⑦任意大的数	⑧空
运输起点	⑨<50 的字符串	⑩空
运输终点	(1)<50 的字符串	(2)空
最低价格	(3)>0 任意整数	(4)空
最高价格	(5)<9999 任意整数	(6)空
留言	(7)长度小于 255 的字符串 (8)空	无

表 3-12 合理等价类测试用例

测试数据								期望结果	覆盖的价
货物名称	货物类别	货物重量	运输起点	运输终点	最低价格	最高价格	留言		
零食	产 品、食 品及饮 品	112	河工 大	北京	0	123	(空)	发布成功	① ⑦ (1) (5)

一些东西	纺织原料及制品	1	石家庄	河南	0	132	事情紧急	发布成功	① ⑦ (1) (5)

表 3-13 不合理等价类测试用例

测试数据								期望结果	覆盖的等价类
货物名称	货物类别	货物重量	运输起点	运输终点	最低价格	最高价格	留言		
(空)	煤炭	12	河工大	北京	0	136	(空)	请输入货物名称	③④⑦ ⑨(1)(3) (5)(8)
零食	产品、食品及饮品	(空)	河工大	北京	0	123	(空)	请输入货物重量	①④⑧ ⑨(1)(3) (5)(8)
零食	产品、食品及饮品	112	(空)	(空)	0	123	(空)	请输入运输起点、终点	①④⑦ ⑩(2)(3) (5)(8)
零食	产品、食品及饮品	112	河工大	北京	(空)	(空)	(空)	请输入价格	①④⑦

									⑨)(1)(4 (6)(8)
--	--	--	--	--	--	--	--	--	-------------------

表 3-14 发布货物功能测试表

编号	功能名称	输入内容	期望结果	输出结果	发现问题	测试结果	测试时间	测试人
1	发布货物	零食，农副产品、食品及饮品，112，河工大，北京，0，123，（空）	系统提示发布成功	系统提示发布成功	无	√	12.5	刘景贤
2	发布货物	一些东西，纺织品原料及制品，1，石家庄，河南，0，132，事情紧急	系统提示发布成功	系统提示发布成功	无	√	12.5	刘景贤
3	发布货物	（空），煤炭，12，河工大，北京，0，136，（空）	系统提示请输入货物名称	系统提示请输入货物名称	无	√	12.5	刘景贤
4	发布货物	零食，农副产品、食品及饮品，（空），河工大，北京，0，123，	系统提示请输入货物重量	系统提示请输入货物重量	无	√	12.5	刘景贤
5	发布货物	零食，农副产品、食品及饮品，112，（空），（空），0，123，	系统提示请输入运输起点、	系统提示请输入运输起	无	√	12.5	刘景贤



			终点	点、终点				
6	发布货物	零食，农副产品、食品及饮品，112，河工大，北京，（空），（空），（空）	系统提示请输入价格	系统提示请输入价格	无	√	12.5	刘景贤

3.3.3 运单浏览模块

表 3-15 等价类划分

输入数据	有效等价类	无效等价类
运输起点	①系统已有发布货物信息的起点 ②空	③系统中没有的运输起点
运输终点	④系统已有发布货物信息的对应的终点 ⑤系统已有发布货物信息的起点 ⑥空	⑦系统已有发布货物信息的不对应的终点
货物类型	⑧系统中的货物类型	⑨空

表 3-16 合理等价类测试用例

测试数据			期望结果	覆盖的等价类
运输起点	运输终点	货物类型		

河工大	黑龙江	零食	查询成功,返回相应的货物信息	①④⑧
河工大	(空)	零食	查询成功,返回相应的货物信息	①⑥⑧
(空)	黑龙江	零食	查询成功,返回相应的货物信息	②⑤⑧
(空)	(空)	零食	查询成功,返回相应的货物信息	②⑥⑧

表 3-17 不合理等价类测试用例

测试数据			期望结果	覆盖的等价类
运输起点	运输终点	货物类型		
南极	黑龙江	零食	系统提示没有该货物	③④⑧
河工大	北京	零食	系统提示没有该货物	①⑦⑧
河工大	黑龙江	(空)	系统提示请输入货物类型	①④⑨

表 3-18 浏览运单功能测试表

编号	功能名称	输入内容	期望结果	输出结果	发现问题	测试结果	测试时间	测试人
1	浏览运单	河工大, 黑龙江, 零食	查询成功, 返回相应的货物信息	查询成功, 返回相应的货物信息	无	√	12.5	王超
2	浏览运单	河工大, (空), 零食	查询成功, 返回相应的货物信息	查询成功, 返回相应的货物信息	无	√	12.5	王超
3	浏览运单	(空), 黑龙江, 零食	查询成功, 返回相应的货物信息	查询成功, 返回相应的货物信息	无	√	12.5	王超
4	浏览运单	(空), (空), 零食	查询成功, 返回相应的货物信息	查询成功, 返回相应的货物信息	无	√	12.5	王超
5	浏览运单	南极, 黑龙江, 零食	系统提示没有该货物	系统提示没有该货物	无	√	12.5	王超
6	浏览运单	河工大, 北京, 零食	系统提示没	系统提示	无	√	121.5	王超

			有该货物	没有该货物				
7	浏览运单	河工大，黑龙江，（空）	系统提示请输入货物类型	系统提示请输入货物类型	无	√	12.5	王超

### 3.4 测试结论(Test Verdict)

测试日期：2021.12.8

测试地点：河北工业大学北辰校区实验室

测试环境：Windows10, firefox 浏览器

列出系统的强项：界面美观，交互性强，系统并发性强。

列出系统的弱项：在进行系统设计时，没有考虑到统计的设计。

列出不符合项的统计结果：无

测试人员签字：王永航，刘景贤，王凡，王超

### 4、制作心得与实验总结

**刘景贤：**

**完成任务：**确定任务选题，明确项目目的与目标，数据库设计，系统前端页面设计，系统前端界面实现，协调开发进度，系统后端接口开发，系统前后端接口调试，实验报告的修改，项目系统检查

**完成情况：**认真完成自己负责的工作，与小组其他成员积极配合，最终完成整个项目的设计、开发与测试。

**收获与不足：**通过本次综合实验，我个人收获颇多，也学习到了许多开发的有关知识。更重要的是我可以在实践中学习了更多的开发知识，编程能力也大大提高，不足的是设计系统中还存在一些考虑不周的情况，导致数据库在编程实现过程中加以修改。需要在之后的设计中多加注意。

**王永航：**

**完成的任務：**

確定任務選題、明確項目目的與目標、部分用例分析、角色分析，在系統實現中負責系統前後端部分代碼編寫、前後端部分接口調試以及系統部分功能測試。

**完成情況：**

能夠完成組內分配的各项任務，與小組其他成員積極配合最終完成整個項目的設計、開發與測試以及修復在測試中遇到的一些問題。

**收穫及不足：**

通過本次實驗，讓我了解到了軟件開發的整個流程，也体会到了軟件設計與開發的不易。正因為如此，我在本次實驗中學習到了很多東西，從實驗報告編寫上，我鞏固了課上所學的知识。從編碼實現上，我學到了很多設計方法以及編程思想，極大提高了我的編碼能力。從小組協作上，加強了團隊合作精神。此次實驗收穫很大，希望我們以後可以學到更多，一步一步變得強大。

但本次實驗還是存在一些不足，由於我們每個人都是第一次來實現一個系統，我們每個人都盡自己最大努力去讓系統變得更好，但無可厚非，由於時間和技術能力的限制，系統仍有许多方面待改進。

**王凡：**

**完成的任務：**

確定任務選題、畫用例圖、部分用例分析、參與數據庫的設計，設計界面原型，在系統實現過程中提出部分意見，在系統實現中完成了部分系統後端代碼編寫，以及系統部分功能測試。

**完成情況：**

較為成功的完成組內分配的各项任務，在組內成員完成部分不對應時，及時與小組成員溝通協調，共同商討最終結果，與小組其他成員積極配合最終完成整個項目的設計、開發與測試以及修復在測試中遇到的一些問題。

**收穫及不足：**

在系統前期設計階段考慮不周全，致使後面完成時出現各種問題，但好在及時協商修正，較為良好的完成了系統開發。通過本次實驗，學習到了軟件開發的基本流程，每一個環節都是必不可少的，了解基本框架，學會了畫用例圖、用例表、以及設計、測試相關文檔的書寫。提升了一定的編程能力。未來再次參與系統開發，一定在起步階段考慮周全，降低反復的難度。

**王超：**

**完成的任務：**

確定任務選題、明確項目目的與目標、參與需求分析、寫用例表、部分用例分析、進行角色分析，完成類設計，參與了系統實現，進行了系統功能

测试。

**完成情况：**

成功的完成组内分配的各项任务，得到了大家的帮助，与小组其他成员积极配合最终完成整个项目的设计、开发与测试以及修复在测试中遇到的一些问题。

**收获及不足：**

通过本次实验，我学习到了软件开发的基本流程，对课本上的知识有了更深刻的理解，了解了系统的基本框架，学会了画鲁棒图、序列图、类设计和测试相关文档的书写，提升了编程能力，对将来的学习与系统开发提供了许多经验。

**实验总结：**对于本次实验，可以说我们非常用心去做也花费了不少时间，但是得到的快乐与收获也更多。整个过程之中，小组的每一个人都全力以赴的发挥自己最大的价值，大家不断的彼此提建议，一起讨论，共同合作，除了对于课程的收获之外，大家也拥有着一份团队的凝聚力与小组合作的经验收获，实验的结束，也让我们更珍惜彼此，更珍惜一起努力的日子。

整个系统的完成度还是较高的，但还存在着不足。比如可以在货主和司机的操作界面加入数据更加直观的数据统计功能，使其可以查看自己一个月内发出了多少货物之类的，这样会使本系统更加完整。