|  |  |
| --- | --- |
| **项 目 编 号** |  |
| **文 档 编 号** |  |
| **密 级** |  |

**小型超市管理系统-系统设计**

V1.0

**三峡大学 计算机科学与技术专业 第3组**

评审日期： 2024 年 12 月 6 日

**目 录**

[1. 导言 1](#_Toc186284985)

[1.1 编写目的 1](#_Toc186284986)

[1.2 作用范围 1](#_Toc186284987)

[1.3 参考资料 1](#_Toc186284988)

[2. 项目设计原则简介 2](#_Toc186284989)

[3. 体系结构设计 3](#_Toc186284990)

[3.1 View视图层 4](#_Toc186284991)

[3.2 Controller层 4](#_Toc186284992)

[3.3 Service层 5](#_Toc186284993)

[3.4 数据持久层 6](#_Toc186284994)

[3.5 Model模型层 7](#_Toc186284995)

[4. 功能模块设计 8](#_Toc186284996)

[4.1 功能模块设计总述 8](#_Toc186284997)

[4.2 基本业务模块设计 8](#_Toc186284998)

[4.2.1 模块1：用户认证与授权 8](#_Toc186284999)

[4.2.2 模块2：仓库管理 9](#_Toc186285000)

[4.2.3 模块3：商品管理 10](#_Toc186285001)

[4.2.4 模块4：配送管理 10](#_Toc186285002)

[4.2.5 模块5：系统日志 11](#_Toc186285003)

[4.2.6 模块6：邮件验证码登录 11](#_Toc186285004)

[5. 数据库设计 12](#_Toc186285005)

[5.1 数据库种类特点 12](#_Toc186285006)

[5.2 数据库逻辑结构 12](#_Toc186285007)

[5.3 物理结构设计 13](#_Toc186285008)

[5.3.1 admin表 13](#_Toc186285009)

[5.3.2 distribution表 13](#_Toc186285010)

[5.3.3 driver表 14](#_Toc186285011)

[5.3.4 inventory\_record表 14](#_Toc186285012)

[5.3.5 inventory表 14](#_Toc186285013)

[5.3.6 sale表 15](#_Toc186285014)

[5.3.7 employee表 15](#_Toc186285015)

[5.3.8 vehicle表 15](#_Toc186285016)

[5.3.9 commodity表 16](#_Toc186285017)

[5.3.10 code表 16](#_Toc186285018)

[5.3.11 company表 16](#_Toc186285019)

[5.3.12 login\_log表 16](#_Toc186285020)

[5.3.13 system\_log表 17](#_Toc186285021)

[5.3.14 user表 17](#_Toc186285022)

[5.3.15 warehouse表 17](#_Toc186285023)

[6. 界面设计 18](#_Toc186285024)

[6.1 登录注册页面设计 18](#_Toc186285025)

[6.2 基础管理页面设计 18](#_Toc186285026)

[6.3 销售管理页面设计 20](#_Toc186285027)

[6.4 配送管理界面设计 21](#_Toc186285028)

[6.5 运输管理界面设计 22](#_Toc186285029)

[6.6 图表分析界面设计 23](#_Toc186285030)

[6.7 系统管理界面设计 24](#_Toc186285031)

[6.8 日志管理界面设计 26](#_Toc186285032)

# 导言

## 编写目的

该文档根据小型超市管理系统的功能和性能，阐述了小型超市管理系统的系统设计，包括框架设计，功能模块设计，数据库设计，界面设计等部分。

本文档的预期读者包括：

* 开发人员
* 项目管理人员
* 测试人员

## 作用范围

该文档的目的是解决项目如何开发的问题，根据需求分析与实际的情况设计出项目开发的整体流程，让开发变得更有条理，让开发人员合理有序的进行开发。

## 参考资料

[1] 软件工程：实践者的研究方法（原书第八版·本科教学版）/（美）罗杰 S.普莱斯曼（Roger S.Pressman），（美）布鲁斯 R.马克西姆（Bruce R.Maxim）著；郑仁杰等译.—北京：机械工业出版社；

[2] UML2面向对象的分析与设计/谭火彬著.—北京：清华大学出版社；

[3] 人月神话（40周年中文纪念表）/(美)布鲁克斯（Brooks,F.P.）著；UML China翻译组，汪颖译.—北京：清华大学出版社。

# 项目设计原则简介

随着经济的发展和人民收入的提高，小型超市在日常生活中扮演着越来越重要的角色。作为社区消费的重要组成部分，小型超市不仅为消费者提供了方便快捷的购物体验，还为商家创造了良好的商业机会。然而，随着经营规模的扩大和市场竞争的加剧，传统的手工管理方式逐渐暴露出效率低下、数据不准确以及管理难度大的问题。为了提升商家的管理效率、优化顾客的购物体验，同时为商家提供科学的决策支持，一个功能全面且易于操作的小型超市管理系统应运而生。该系统通过高度集成化的管理流程，不仅可以有效改善超市的日常运营，还能通过统计分析等功能帮助商家了解市场动态，制定更具竞争力的经营策略，从而在市场中占据优势地位。

在整个系统设计的过程中遵循以下设计原则：

**1.实用性：**实用性是系统的主要设计原则，系统设计必须最大可能地满足用户的需求，做到操作方便、界面友好、可即时更新，能适应不同情况下的需求。

**2.先进性：**信息技术发展迅速，系统设计尽可能采用先进的技术标准和技术方法。

**3.人性化：**要有设计良好的人机交互界面，即使系统的操作界面简单易用，又能具有较强的适用性，满足不同计算机使用水平的用户使用。

**4.灵活和易维护：**采用开放的体系架构，基于开放源代码的技术框架和数据库系统，使用高效率的开源和免费开发工具，具备完整的文档说明。在维护方面，主要考虑两个层面，一是对于开发人员来讲，系统编码容易调整，可适应需求的变化和调整；二是对于系统管理维护人员来说，能够对系统进行便捷的维护和管理。

**5.安全性好：**选择安全可靠的软硬件运行平台，并在系统设计和实现的时候关注系统的安全控制和执行效率，提供相应的安全防护功能，保证系统具有较高的安全性和可靠性。安全性方面，要考虑系统的安全、数据管理的安全、网络安全。保证用户权限、数据安全和系统的稳定性。

# 体系结构设计

本项目采用了典型的三层架构设计，包括View 视图层、Controller 控制层和Model 模型层，以实现功能模块的高效分离和灵活扩展。

View 视图层主要负责用户界面的展示与交互，通过Vue2技术构建支持多角色操作的界面，为管理员、收银员、销售员及会员提供清晰的导航和友好的用户体验。前端与后端通过RESTful API通信，采用JSON格式传输数据，确保接口的通用性和易用性。

Controller 控制层基于Spring Boot框架开发，负责系统的核心逻辑处理。通过Spring MVC模式实现请求路由和响应，确保功能模块之间的高效协作，同时加入了严格的权限管理机制，保证用户只能访问授权的功能模块。

Model 模型层使用MySQL数据库存储商品、订单和库存等信息，通过JPA实现高效的数据操作。整体技术栈为Spring Boot + JPA，结合事务管理、日志记录和异常处理机制，确保系统的高可靠性、可扩展性和易维护性。

三层架构的设计使系统具有良好的可维护性和扩展性，能够适应未来功能的升级与调整需求。

此外，Spring boot是spring的优化，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。相比传统的spring框架，springboot配置更少，可以减少开发的难度和时间，提高开发的效率。

Spring boot的四个主要特性：

* **SpringBoot Starter：**他将常用的依赖分组进行了整合，将其合并到一个依赖中，这样就可以一次性添加到项目的Maven或Gradle构建中。
* **自动配置：**SpringBoot的自动配置特性利用了Spring4对条件化配置的支持，合理地推测应用所需的bean并自动化配置他们。
* **命令行接口：**（Command-line-interface, CLI）：SpringBoot的CLI发挥了Groovy编程语言的优势，并结合自动配置进一步简化Spring应用的开发。
* **Actuatir：**它为SpringBoot应用的所有特性构建一个小型的应用程序。但首先，我们快速了解每项特性，更好的体验他们如何简化Spring编程模型。

Spring Data JPA 是基于 JPA 规范的一个强大持久层框架，旨在简化数据访问层的开发。它通过继承 JpaRepository 或其他接口，提供开箱即用的 CRUD 操作，支持基于方法名解析的查询、分页和排序功能，同时通过 @Query 注解或 Specification API 支持复杂查询和动态条件查询。Spring Data JPA 深度集成 Spring 框架，具备内置事务管理、依赖注入等功能，并提供对实体之间一对一、一对多、多对多关系的良好支持。它适合快速开发基于数据库的业务应用，尤其是在 CRUD 频繁、表结构相对简单的场景中；然而，对于需要精确控制 SQL 性能的复杂查询场景，灵活性可能略显不足。结合 Spring 的生态系统，Spring Data JPA 是现代 Java 开发中提高生产效率的重要工具。

## View视图层

视图是用户看到并与之交互的界面。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

视图层功能包括：

根据用户身份信息进行识别登录，支持使用邮箱验证码和人脸识别登录。

根据用户身份信息进行识别实用系统，进行不同的数据和逻辑交互，满足用户需求 。

通过页面控件，完成统一风格的页面部署，形成较好的页面风格。

对前端提交数据进行合规性校验，如登入时账号密码长度，类型，格式等等。

将页面请求发送到验证层，并接受验证层返回，将结果在前端展示。

提供过滤功能，将非法的请求给拦截掉。

视图层主要由下面组件构成：

前端模板组件：根据控制层的需求提供前端模板页面，给用户展示出正确的前端页面。部分前端模板页面如表 3‑1所示。

表 3‑1部分前端模板页面

|  |  |  |
| --- | --- | --- |
| 文件名称 | VUE | 说明 |
| 首页(商品管理) | Commodity.vue | 显示首页(商品管理)界面 |
| 登录 | Login.vue | 显示登录界面 |
| 员工管理 | Employee.vue | 显示员工管理界面 |
| 仓库管理 | Warehouse.vue | 显示仓库管理界面 |
| 图表分析 | In.vue  Out.vue | 显示销售分析出入库界面 |
| 系统管理 | Security.vue  Admin.vue  Role.vue | 显示系统安全管理界面  显示系统操作员管理界面  显示系统权限管理界面 |
| 日志管理 | LoginLog.vue  SystemLog.vue | 显示登录日志界面  显示系统日志界面 |

## Controller层

 Controller，从字面上理解是控制器，所以它是负责业务调度的，所以在这一层应写一些业务的调度代码。

而且Controller层功能包括：

接收并处理从表现层传入的各种输入以及输出各种异常提示信息或处理结果信息。

对于输入的数据进行数据校验，过滤非法数据。

向业务控制层发送处理请求。

接收业务逻辑层请求处理结果。

返回状态符，实现页面跳转。

控制包部分结构如表3‑2所示。

表3‑2 controller结构说明

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.ctgu.api.controller | AdminController | 管理员控制器，主要处理Admin的业务调度 |
| CommodityController | 商品控制器，主要负责商品相关的业务调度 |
| DistributionController | 配送控制器，主要负责配送相关的业务调度 |
| DriverController | 司机控制器，主要负责司机相关的业务调度 |
| EmailController | 邮箱控制器，主要负责邮箱验证码的业务逻辑 |
| EmployeeController | 员工控制器，主要负责员工相关的业务调度 |
| InventoryController | 库存控制器，负责商品库存相关的业务调度 |
| LogController | 日志控制器，负责各种日志相关的业务调度 |
| RoleController | 角色控制器，负责系统角色的业务调度 |
| SaleController | 销售控制器，负责商品销售相关业务调度 |
| UserController | 用户控制器，负责平台用户的业务调度 |
| VehicleController | 车辆控制器，负责配送车辆的业务调度 |
| WarehouseController | 仓库控制器，主要负责商品仓库的业务调度 |

## Service层

即业务逻辑层，service不单纯是对于dao的增删改查调用，service是业务层，所以应该更切近于具体业务功能要求，所以在这一层，一个方法所体现的是一个可以对外提供的功能。

业务逻辑层功能包含：

* 实现各种业务处理逻辑或处理算法，比如用户筛选，分页查询等等。
* 接受来自控制层的信息，并做出相应的处理。
* 向持久层发送数据操作的请求，进行对数据信息的增删改查操作。
* 作为控制层的服务层，提供接口供周边系统调用。

由负责处理各类业务逻辑的 service 组成，由于系统业务逻辑层的实现类较多， 此处列举一些核心的实现类，如表 3‑3所示。

表 3‑3 业务逻辑层结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.ctgu.api.service | AdminService | 管理员管理模块Service接口 |
| CommodityService | 商品管理模块Service接口 |
| DistributionService | 配送管理模块Service接口 |
| DriverService | 司机管理模块Service接口 |
| EmailService | 邮箱发送管理模块Service接口 |
| EmployeeService | 员工管理模块Service接口 |
| InventoryService | 库存管理模块Service接口 |
| LogService | 日志管理模块Service接口 |
| RoleService | 系统角色管理模块Service |
| SaleService | 销售管理模块Service接口 |
| UserService | 用户管理模块Service接口 |
| VehicleService | 车辆管理模块Service接口 |
| WarehouseService | 仓库管理模块Service接口 |

## 数据持久层

智慧超市管理系统的数据持久层通过整合 Spring Boot 框架和数据访问技术（如 JPA/Hibernate、MyBatis 或 MyBatis-Plus）实现高效的数据库操作。持久层主要通过实体类与数据库表的映射完成数据存储、查询、更新和删除功能，利用 Repository 或 Mapper 接口简化常见 CRUD 操作，同时结合事务管理（@Transactional）确保数据一致性。此外，为提高性能，持久层可引入分页查询、数据库索引优化，以及 Redis 缓存等机制，有效支持系统的稳定运行和高效扩展。

数据持久层事务的四个特性：

* **原子性**（Atomicity）**：**操作这些指令时，要么全部执行成功，要么全部不执行。
* **一致性**（Consistency）**：**事务的执行使数据从一个状态转换为另一个状态，但是对于整个数据的完整性保持稳定。
* **隔离性**（Isolation）**：**隔离性是当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。
* **持久性**（Durability）**：**当事务正确完成后，它对于数据的改变是永久性的。

数据持久层包结构如表 3‑4所示。

表 3‑4 数据持久层结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 说明 |
| com.ctgu.api.repository | AdminRepository | 管理员管理模块持久层接口 |
| CodeRepository | 验证码模块持久层接口 |
| CommodityRepository | 商品管理模块持久层接口 |
| DistributionRepository | 配送管理模块持久层接口 |
| DriverRepository | 司机管理模块持久层接口 |
| EmployeeRepository | 员工管理模块持久层接口 |
| InventoryRecordRepository | 库存记录管理模块持久层接口 |
| InventoryRepository | 库存管理模块持久层接口 |
| LoginLogRepository | 登录日志管理模块持久层接口 |
| SaleRepository | 销售管理模块持久层接口 |
| SystemLogRepository | 系统日志管理模块持久层接口 |
| UserRepository | 用户管理模块持久层接口 |
| VehicleRepository | 车辆管理模块持久层接口 |
|  | WareHouseRepository | 仓库管理模块持久层接口 |

## Model模型层

智慧超市管理系统的模型层通过实体类定义系统的数据结构，与数据库表进行一一映射，并通过注解（如 @Entity、@Table、@Column）指定字段和关系（如一对多 @OneToMany、多对多 @ManyToMany）。模型层不仅封装了核心业务数据，还支持字段校验（如 @NotNull、@Size）和关系映射，确保数据合法性和一致性，同时可以扩展额外逻辑属性（如计算折扣、格式化时间），为持久层和服务层提供基础支持，方便系统功能的实现与维护。

模型层结构如表 3‑5所示。

表 3‑5 模型层接结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.ctgu.api.model.entity | Admin | 管理员管理的存储 |
| Code | 验证码的存储 |
| Commodity | 商品管理的存储 |
| Company | 公司管理的存储 |
| Distribution | 配送管理的存储 |
| Driver | 司机管理的存储 |
| Employee | 员工管理的存储 |
| Inventory | 库存管理的存储 |
| InventoryRecord | 库存记录管理的存储 |
| Loginlog | 登录日志的存储 |
| Sale | 销售管理的储存 |
| SystemLog | 题目评论信息的储存 |
| User | 用户管理的存储 |
| Vehicle | 车辆管理的存储 |
| Warehouse | 仓库管理的存储 |

# 功能模块设计

## 功能模块设计总述

本系统采用模块化设计思想，基于Spring Boot框架构建，遵循面向对象设计原则，将功能划分为六大核心模块。系统以用户认证与授权模块为基础，通过JWT和Spring Security实现安全控制；以仓库管理和商品管理模块为核心，实现库存和商品的精细化管理；配合配送管理模块完成物流配送环节；通过系统日志模块实现全方位的操作监控；并集成邮件服务模块提供验证码等功能支持。各模块之间职责明确、松耦合、高内聚，通过统一的数据访问层和服务层实现模块间的协同，确保了系统的可扩展性和可维护性。整体设计注重实用性和安全性，采用统一的异常处理和响应机制，为用户提供规范化的接口服务。

其中，图 4‑1是基本业务模块图。

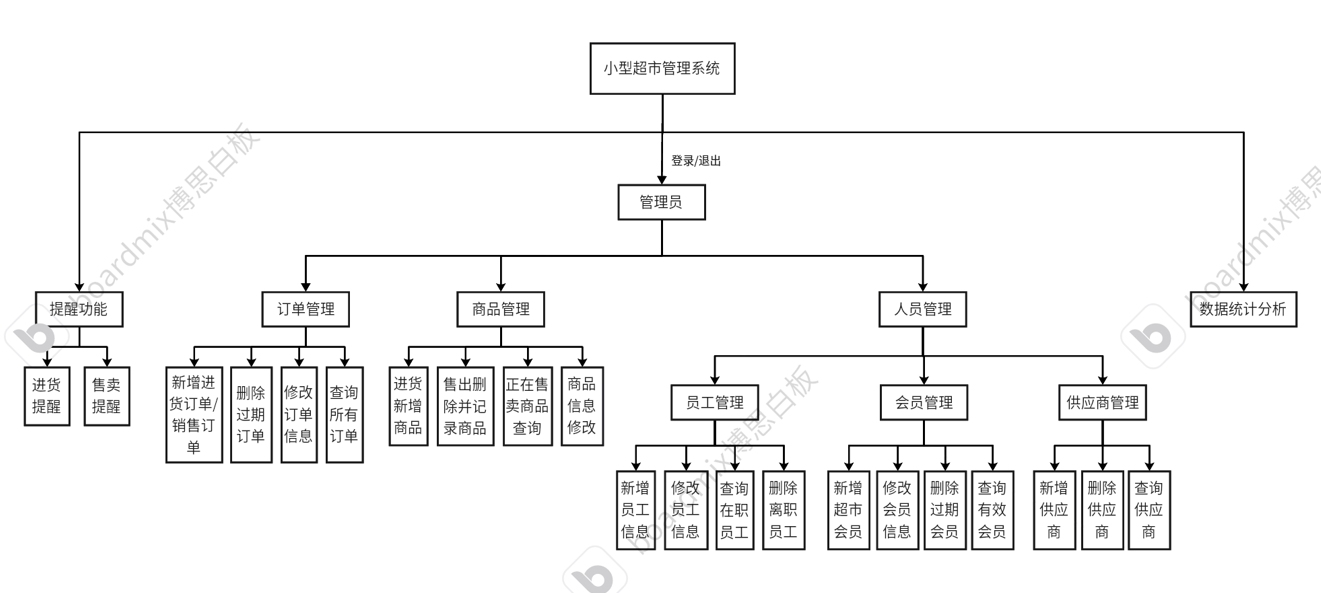


图 4‑1 基本业务模块图

## 基本业务模块设计

### 模块1：用户认证与授权

**功能简介：**采用JWT令牌实现无状态的身份认证，支持邮箱验证码和密码两种登录方式，基于Spring Security实现细粒度的角色权限控制，包含超级管理员、普通管理员等多种角色，并对用户的登录行为进行日志记录，保障系统安全性。

在**操作流程：**如图 4‑2所示：

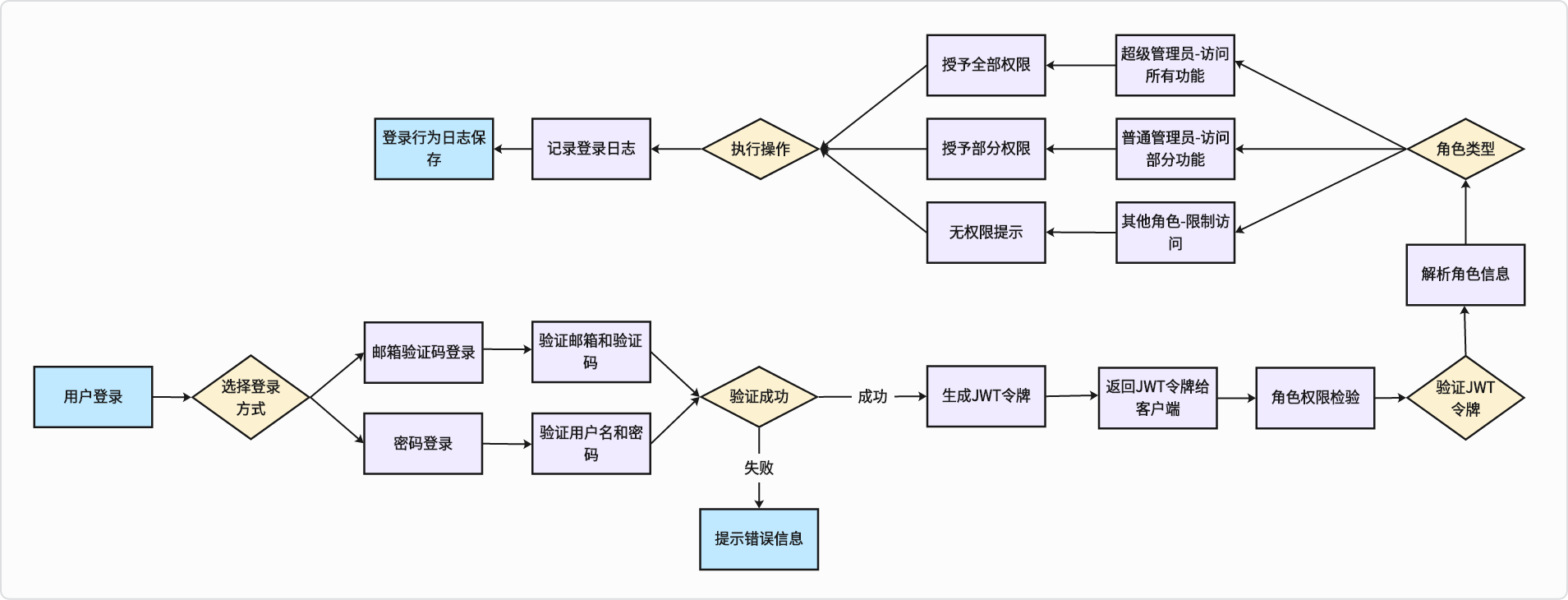


图 4‑2 用户认证授权流程图

### 模块2：仓库管理

**功能简介：**提供仓库基础信息的增删改查功能，实现库存实时监控和管理，记录商品出入库明细，支持多仓库管理，并能够生成库存分析统计报表，帮助管理者及时了解库存状况和流动情况。

**操作流程：**如图 4‑3所示。

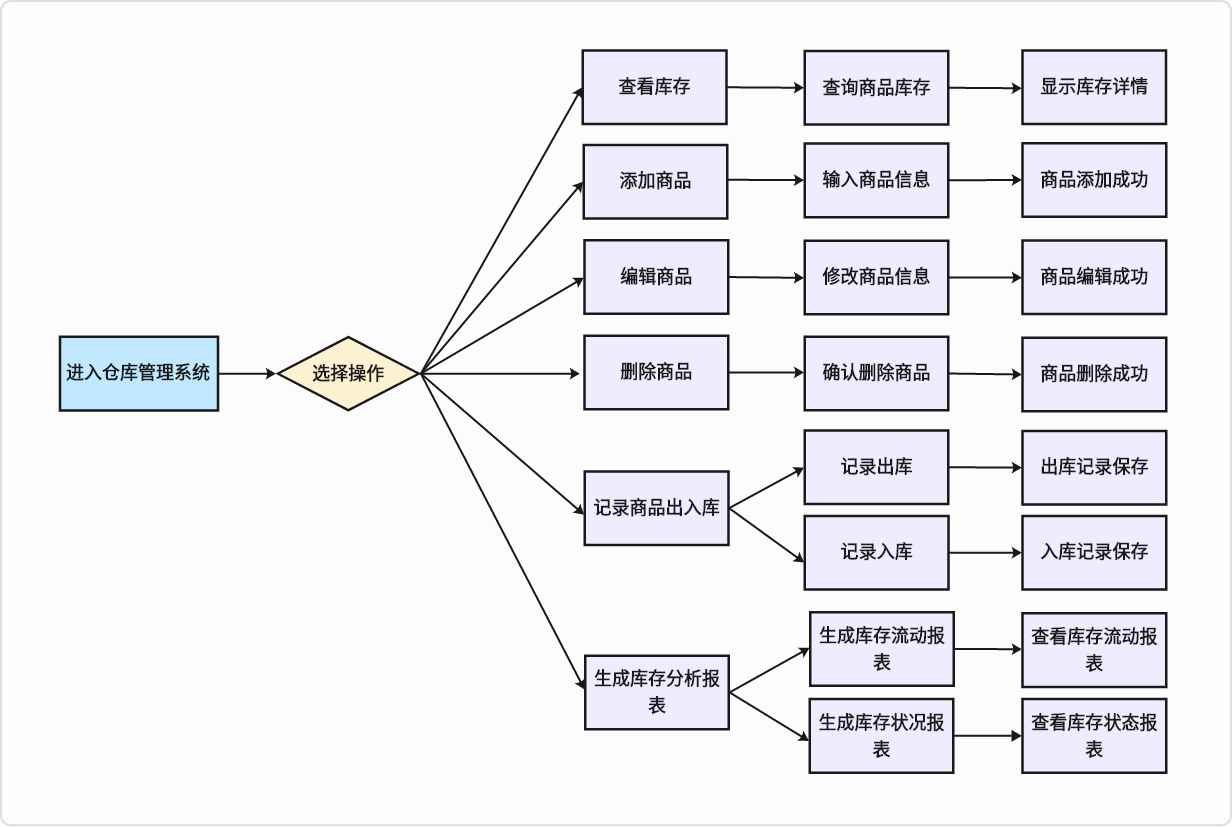


图 4‑3 仓库管理流程图

### 模块3：商品管理

**功能简介：**实现商品信息的全生命周期管理，包括商品的添加、修改、删除和查询，支持按商品名称模糊搜索，记录商品的创建和更新时间，跟踪商品库存变动，为销售和仓储决策提供数据支持。

**操作流程：**如图 4‑4所示。

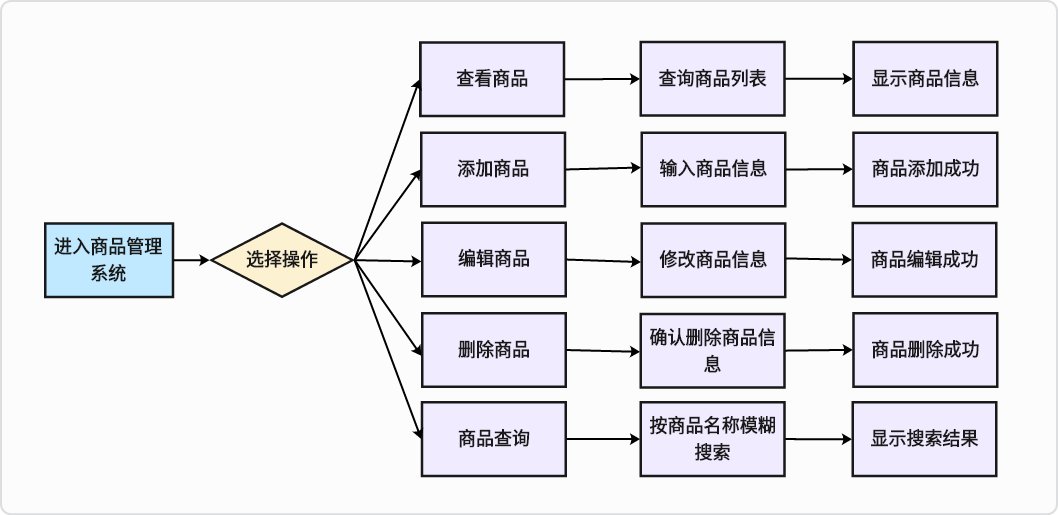


图 4‑4 商品管理流程图

### 模块4：配送管理

**功能简介：**整合配送单管理、车辆调度和司机分配功能，支持配送状态实时更新，包含配送单的创建、分配和跟踪，可查看可用车辆和司机信息，实现配送资源的高效调度。

**操作流程：**如图 4‑5所示。

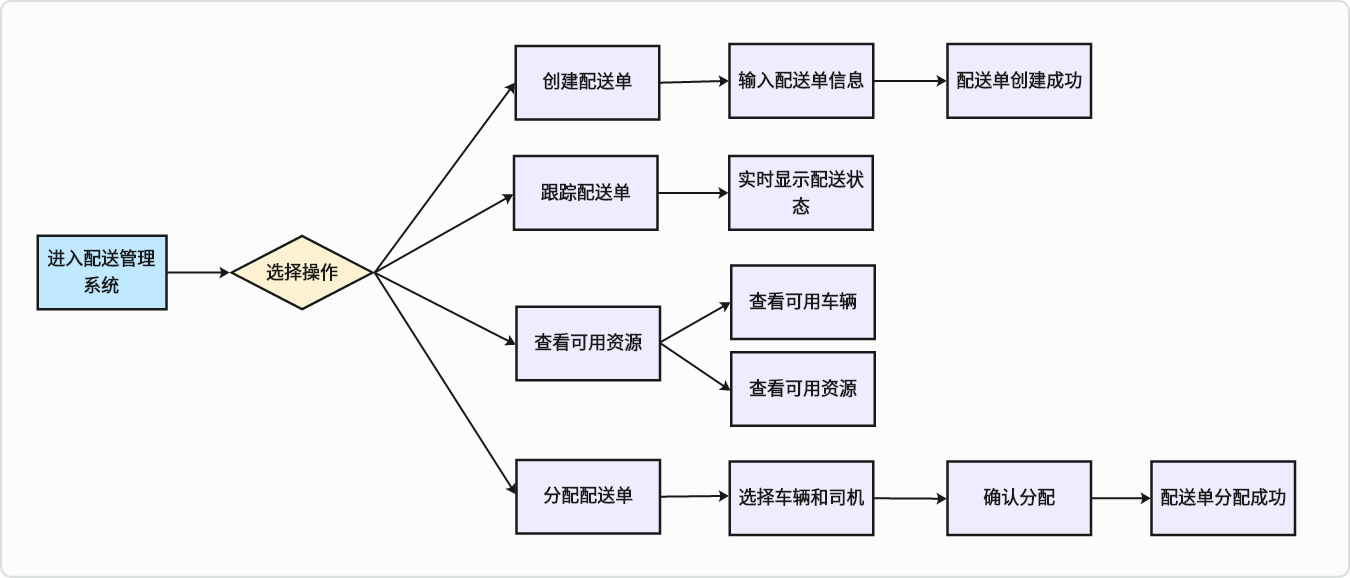


图 4‑5 配送管理流程图

### 模块5：系统日志

**功能简介：**通过AOP技术实现系统操作日志的自动记录，包括用户的登录日志和业务操作日志，记录操作人、操作时间、IP地址等信息，支持日志查询和统计分析，方便系统监控和问题追踪。

**操作流程：**如图 4‑6所示。

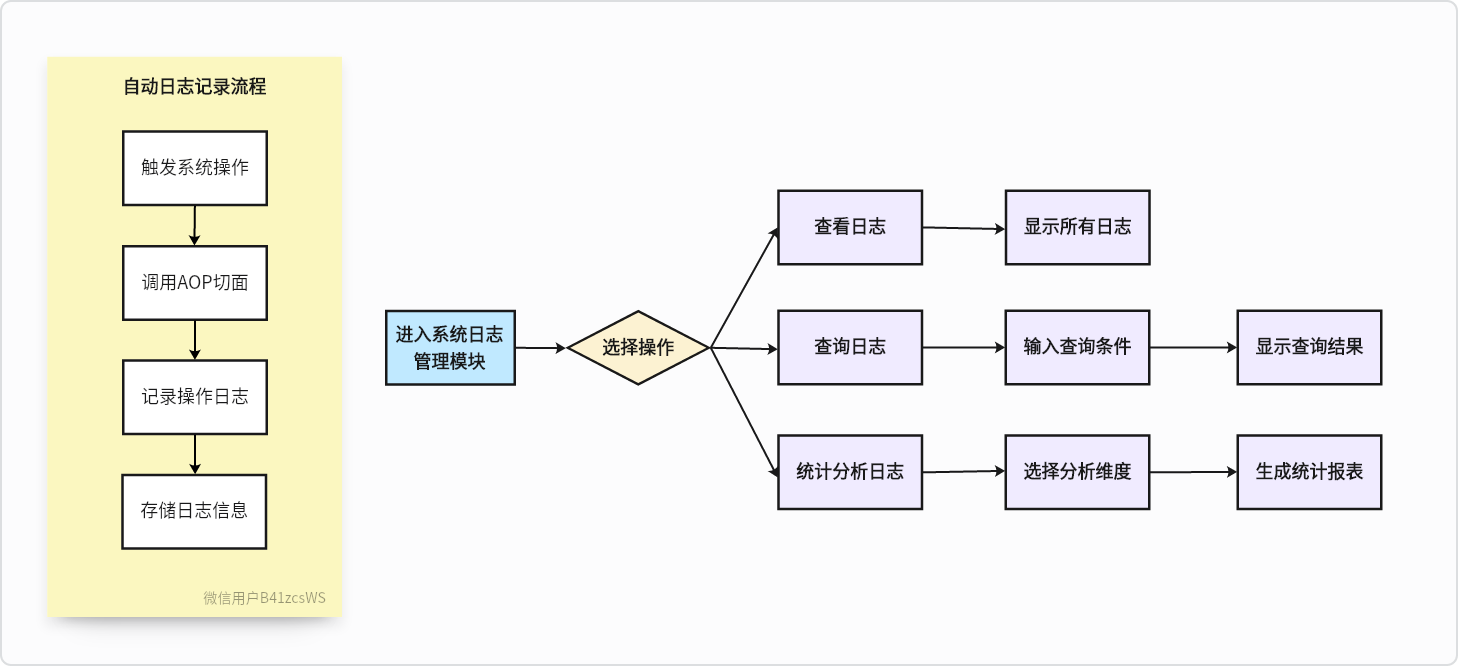


图 4‑6 系统日志记录流程图

### 模块6：邮件验证码登录

**功能简介：**基于Spring Mail实现邮件发送功能，主要用于登录验证码的发送和验证，支持验证码的有效期设置和校验，确保登录安全性，同时为系统通知提供邮件发送支持。

**操作流程：**如图 4‑7所示。

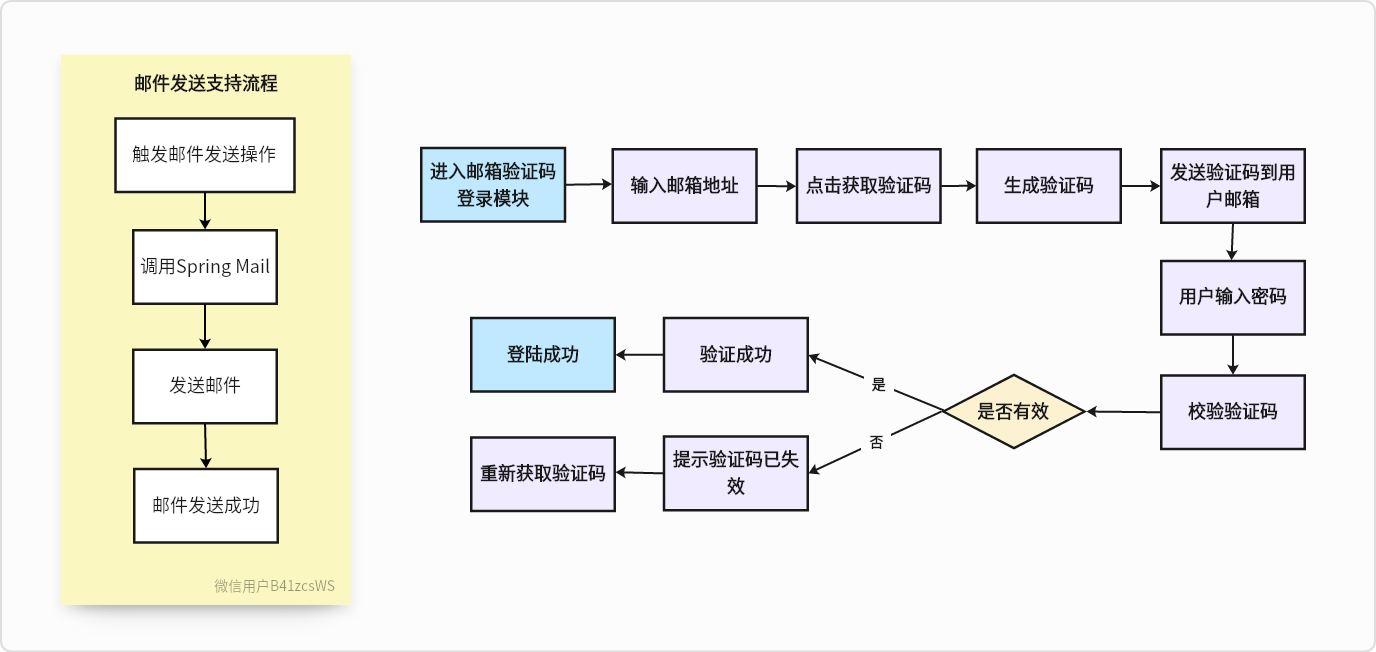


图 4‑7 邮件验证码登录流程图

# 数据库设计

## 数据库种类特点

本项目采用了MySQL 8.0.40的数据库。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL具有以下几个特性：

（1）MySQL为多种编程语言提供了API。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby,.NET和Tcl等。

（2）优化的 SQL查询算法，有效地提高查询速度。

（3）提供多语言支持，常见的编码如中文的GB2312、BIG5，日文的 Shift\_JIS等都可以用作数据表名和数据列名。

（4）提供 TCP/IP、ODBC和JDBC等多种数据库连接途径。

（5）提供用于管理、检查、优化数据库操作的管理工具。

（6）支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

（7）MySQL使用标准的SQL数据语言形式。

## 数据库逻辑结构

经过充分的讨论和分析，我们将“小型超市管理系统”的数据中设计了15个实体，其对应的实体关系如图 5-8：数据库表结构图所示。



图 5-8：数据库表结构图

其中包括admin表，distribution表，driver表，inventory\_record表，inventory表，sale表，employee表，vehicle表，commodity表，code表, company表，login\_log表，system\_log表, user表, warehouse表。

## 物理结构设计

### admin表

管理员表标识的是管理员信息，其表结构如表 5‑1所示：

表 5‑1 管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 管理员ID |
| create\_at | varchar(255) | Y | 注册时间 |
| password | varchar(255) | N | 密码 |
| email | varchar(255) | N | 注册邮箱 |
| roles | varchar(255) | Y | 角色 |

### distribution表

配送表标识的是商品配送的信息，其表结构如表 5‑2所示：

表 5‑2 配送表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 配送ID |
| address | varchar(255) | Y | 客户地址 |
| care | varchar(255) | Y | 注意事项 |
| did | varchar(255) | Y | 司机ID |
| driver | varchar(255) | Y | 司机 |
| number | varchar(255) | Y | 车牌号 |
| phone | varchar(255) | Y | 司机手机号 |
| status | int | Y | 运输状态 |
| time | datetime | Y | 预计送达时间 |
| urgent | bit | N | 是否加急 |
| vid | varchar(255) | Y | 车辆ID |

### driver表

司机信息表标识的是司机的个人信息，其表结构如表 5‑3所示：

表 5‑3 配送员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 司机ID |
| address | varchar(255) | Y | 家庭住址 |
| create\_at | varchar(255) | N | 注册时间 |
| driving | bit(1) | N | 是否在驾驶 |
| gender | varchar(255) | N | 性别 |
| id\_card | varchar(255) | N | 身份证号 |
| license | varchar(255) | N | 驾照编号 |
| name | varchar(255) | N | 姓名 |
| phone | varchar(255) | Y | 手机号 |
| score | varchar(255) | Y | 驾照分数 |
| update\_at | varchar(255) | Y |  |

### inventory\_record表

库存记录表标识的是所有存放到库存里面的商品记录，其表结构如表 5‑4所示：

表 5‑4 库存记录表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 库存记录ID |
| cid | varchar(255) | Y | 商品id |
| count | int | Y | 现有数量 |
| creat\_at | varchar(255) | Y | 创建时间 |
| description | varchar(255) | Y | 描述 |
| name | varchar(255) | N | 名称 |
| type | int | N | 出入库状态 |
| wid | varchar(255) | N | 仓库ID |

### inventory表

库存表标识的是仓库里现存的商品信息，其表结构如表 5‑5所示：

表 5‑5 库存表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 库存ID |
| cid | varchar(255) | N | 商品ID |
| count | int | Y | 现有数量 |
| location | varchar(255) | Y | 位置 |
| name | varchar(255) | N | 名称 |
| wid | varchar(255) | N | 仓库ID |

### sale表

销售表标识的是商品的销售信息，其表结构如表 5‑6所示：

表 5‑6 销售表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 销售ID |
| commodity | varchar(255) | N | 商品 |
| company | varchar(255) | Y | 供应商 |
| count | varchar(255) | Y | 数量 |
| create\_at | varchar(255) | N | 创建时间 |
| description | varchar(255) | Y | 备注信息 |
| number | varchar(255) | N | 打款账号 |
| pay | bit(1) | N | 是否结款 |
| phone | varchar(255) | N | 电话 |
| price | double | N | 价格 |

### employee表

员工表标识的是工作人员的信息，其表结构如表 5‑7所示：

表 5‑7 员工表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 员工ID |
| address | varchar(255) | Y | 家庭住址 |
| create\_at | varchar(255) | N | 注册时间 |
| department | varchar(255) | Y | 所属部门 |
| gender | varchar(255) | N | 性别 |
| id\_card | varchar(255) | N | 身份证号 |
| name | varchar(255) | N | 姓名 |
| phone | varchar(255) | N | 手机号 |
| update\_at | varchar(255) | Y | 更新时间 |

### vehicle表

车辆表标识的是运输车辆的信息，其表结构如表 5‑8所示：

表 5‑8 车辆信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 车辆ID |
| create\_at | varchar(255) | N | 创建日期 |
| driving | bit(1) | N | 是否驾驶 |
| number | varchar(255) | N | 车牌号 |
| type | varchar(255) | Y | 车型 |

### commodity表

商品表标识的是所售商品信息，其表结构如表 5‑9所示：

表 5‑9 商品表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 商品ID |
| count | int | N | 商品数量 |
| create\_at | varchar(255) | N | 创建日期 |
| description | varchar(255) | Y | 描述 |
| name | varchar(255) | N | 商品名 |
| price | double | N | 价格 |

### code表

验证码表存放的是注册时期所使用的验证码，其表结构如表 5‑10所示：

表 5‑10 验证码表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| email | varchar(255) | N | 邮箱 |
| exp | bigint | N | 经验值 |
| value | varchar(255) | Y | 验证码 |

### company表

合作公司表标识的是与本商店合作的公司的信息，其表结构如表 5‑11所示：

表 5‑11 合作公司表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 公司ID |

### 5.3.12 login\_log表

登录日志表标识的是所有尝试注册并登录的账号信息，其表结构如表 5‑12所示：

表 5‑12 登录日志表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 登录ID |
| browser | varchar(255) | Y | 浏览器 |
| date | datetime(6) | N | 登录日期时间 |
| email | varchar(255) | N | 邮箱 |
| ip | varchar(255) | Y | IP地址 |
| status | int | Y | 状态 |

### 5.3.13 system\_log表

系统操作日志表标识的是所有账号的操作记录，其表结构如 [表5-13](#_5.3.13_system_log表) 所示：

表 5‑13 系统操作日志表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 系统日志ID |
| account | varchar(255) | N | 邮箱账号 |
| busincess\_type | varchar(30) | N | 操作 |
| ip | varchar(255) | Y | IP地址 |
| method | varchar(255) | Y | 调用接口方法 |
| module | varchar(255) | Y | 模块 |
| time | datetime(6) | N | 时间 |

### 5.3.14 user表

用户信息表存放的是用户信息，其表结构如 [表5-14](#_5.3.14_user表) 所示：

表 5‑14 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 用户ID |
| create\_at | varchar(255) | N | 创建时间 |
| password | varchar(255) | N | 密码 |
| update\_at | varchar(255) | Y |  |
| username | varchar(255) | Y | 用户姓名 |

### 5.3.15 warehouse表

仓库表存放的是仓库信息，其表结构如 [表5-15](#_5.3.15_warehouse表) 所示：

表 5‑15 仓库表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| id | varchar(255) | N | 仓库ID |
| create\_at | varchar(255) | N | 创建时间 |
| name | varchar(255) | Y | 仓库名 |
| principle | varchar(255) | Y | 负责人 |

# 界面设计

## 登录注册页面设计

图形用户界面, 网站

描述已自动生成

图 6‑1 登录界面

图片包含 自然, 游戏机, 田地, 标志

描述已自动生成

图 6‑2 注册界面

## 基础管理页面设计



图 6‑3 商品管理界面

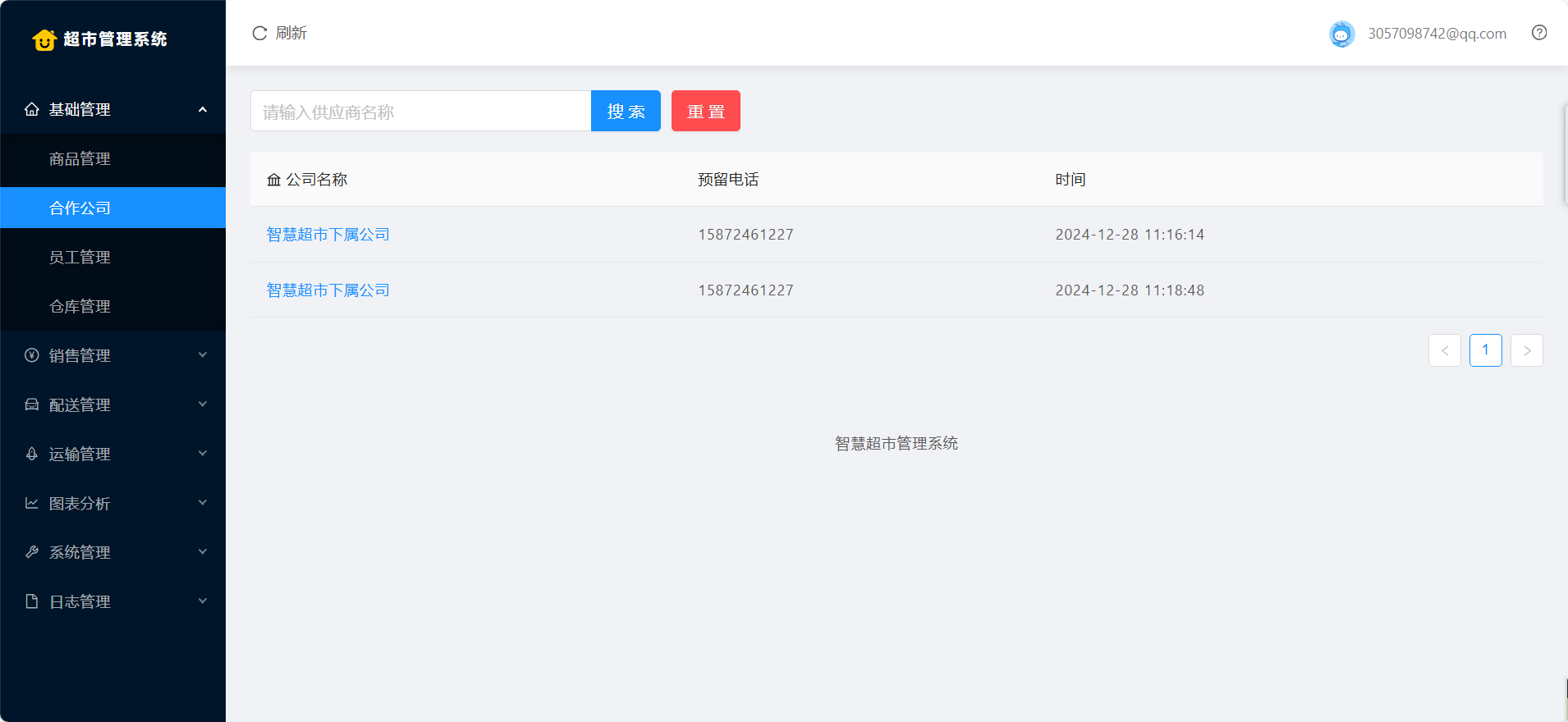


图 6‑4 合作公司界面

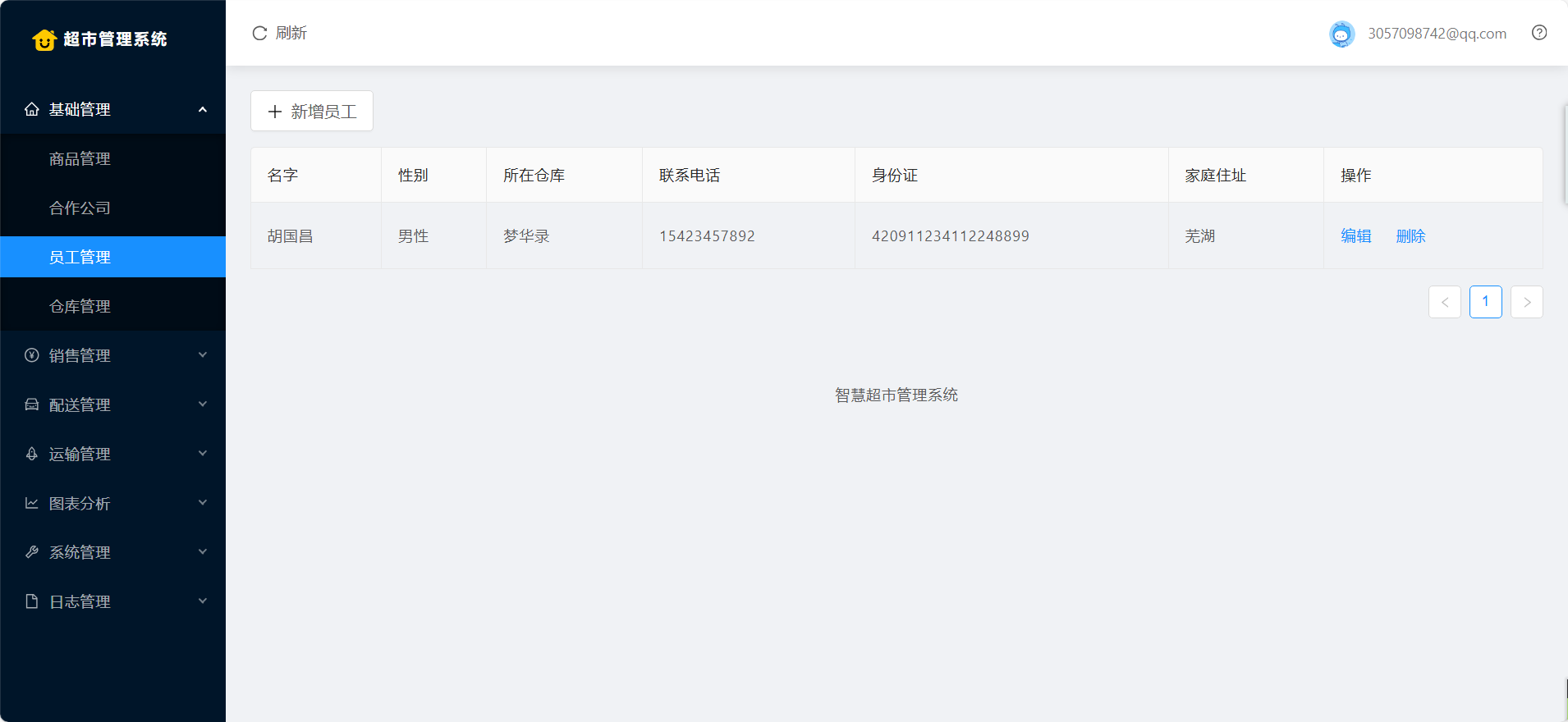


图 6‑5 员工管理界面

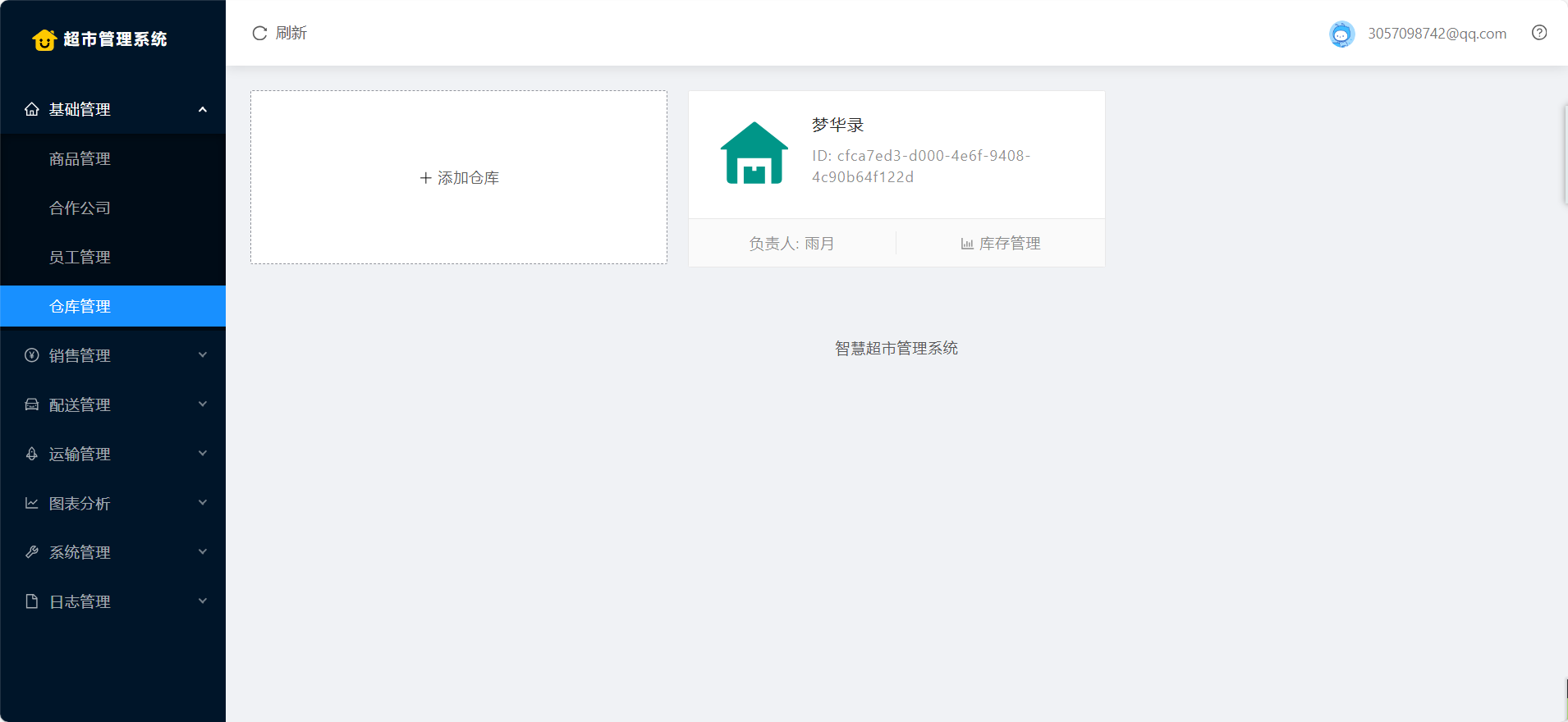


图 6‑6 仓库管理界面

## 销售管理页面设计

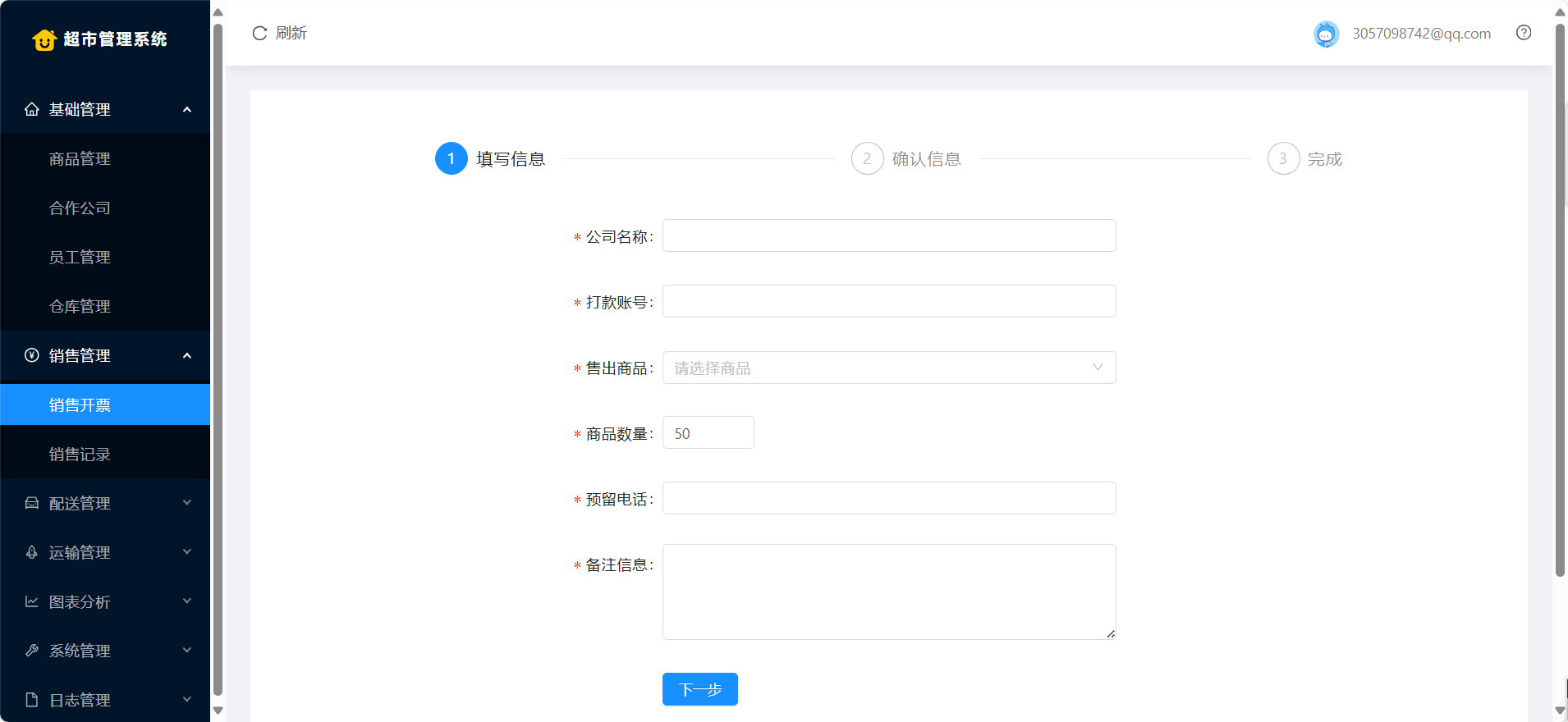


图 6‑7 销售开票界面



图 6‑8 销售记录

## 配送管理界面设计

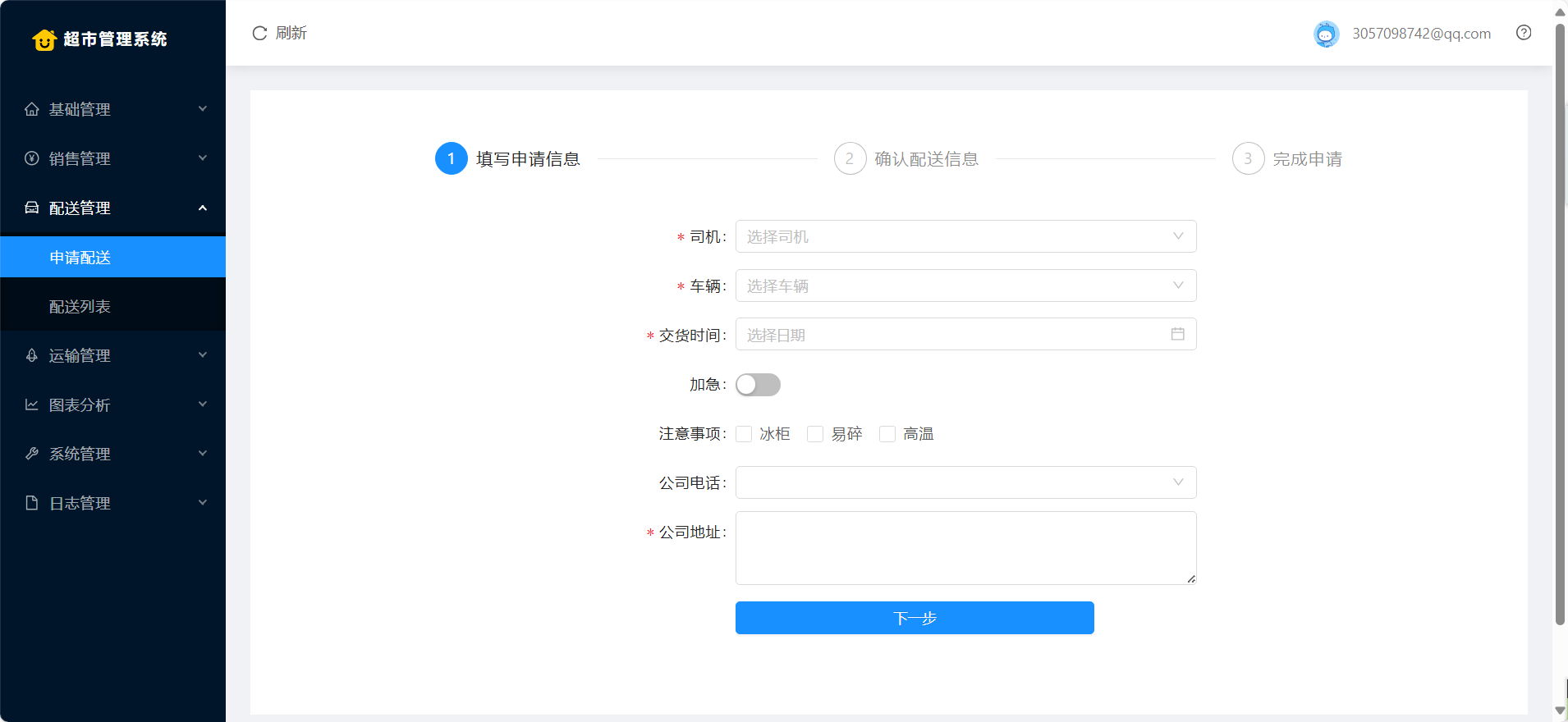


图 6‑9 申请界面



图 6‑10 配送列表界面

## 运输管理界面设计



图 6‑11 车辆资料界面

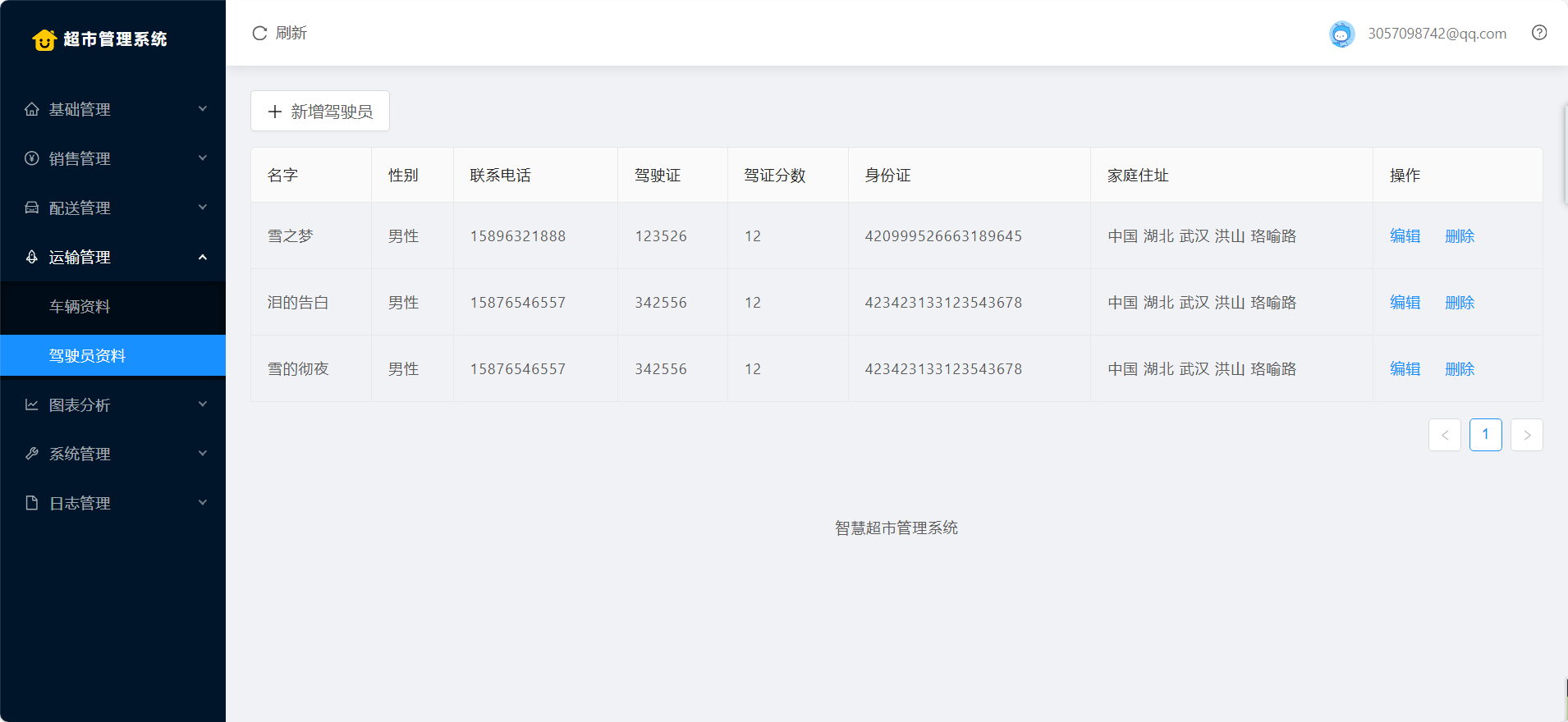


图 6‑12 配送员资料界面

## 图表分析界面设计

图表, 饼图

描述已自动生成图 6‑13 入库分析界面



图 6‑14 出库分析界面

## 系统管理界面设计



图 6‑15 安全设置界面

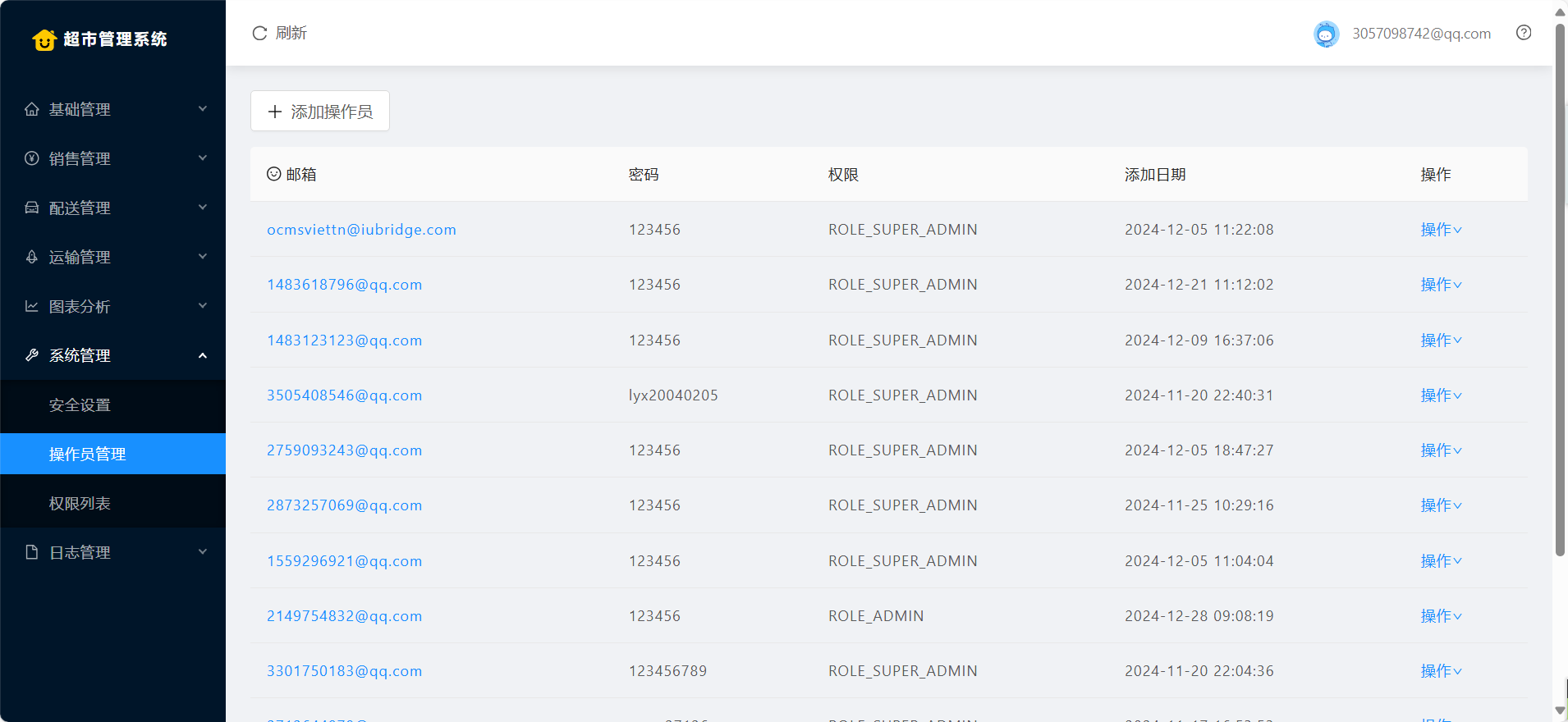


图 6‑16 操作员管理界面



图 6‑17 权限列表界面

## 日志管理界面设计



图 6‑18 登录日志界面

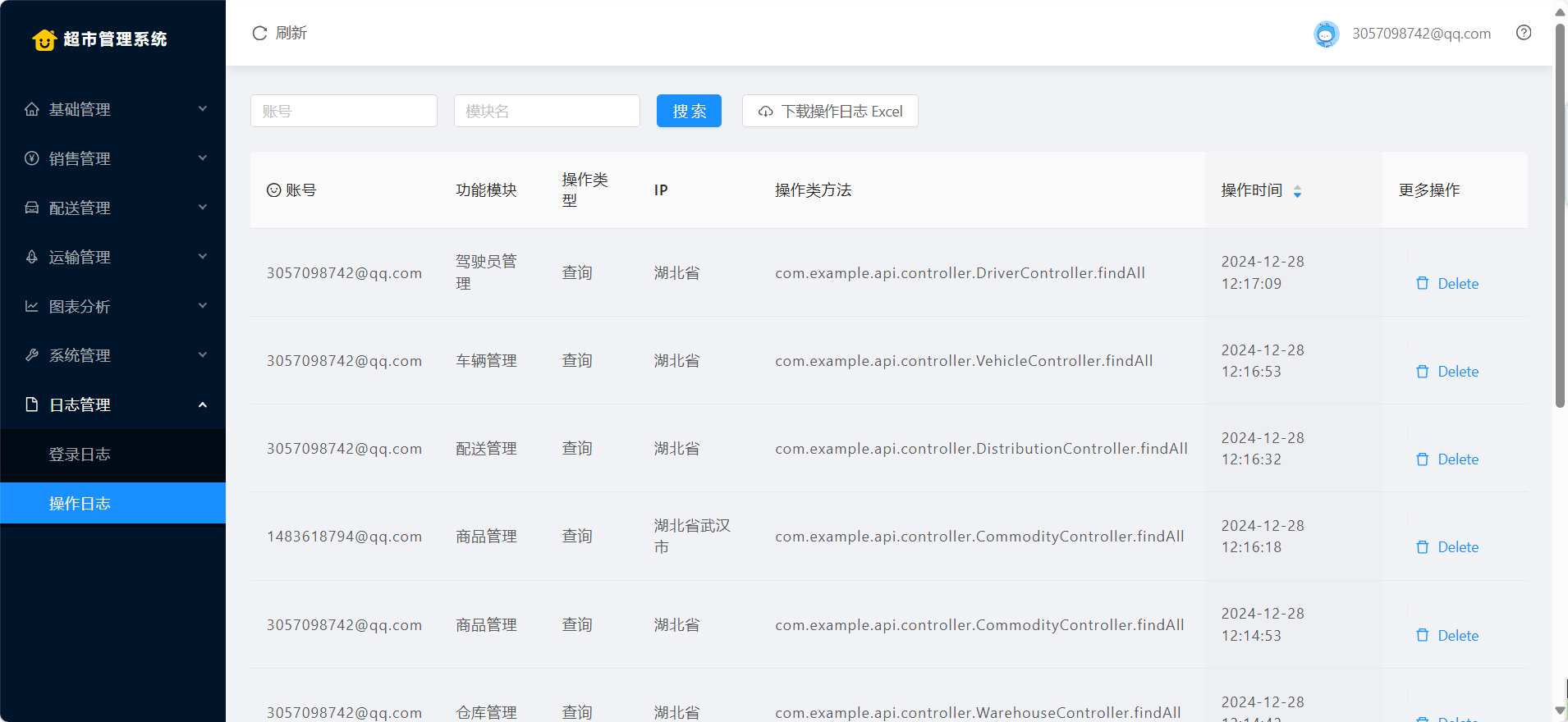


图 6‑19 操作日志界面