



河南财政金融学院

2023 届本科毕业设计

基于 JavaScript GL API 的物流追踪系统

姓 名:	吴玉配
学 号:	20192502010341
专 业:	计算机科学与技术
班 级:	19 级计科 U(3)
指导教师:	赵景海
教学单位:	计算机与人工智能学院

二〇二三年六月十日

河南财政金融学院 毕业论文（设计）版权使用授权书

本人完全了解河南财政金融学院关于收集、保存和使用学位毕业论文（设计）的规定，同意如下各项内容：按照学校要求提交毕业论文（设计）的印刷本和电子版本；学校有权保存毕业论文（设计）的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存毕业论文（设计）；学校有权提供目录检索以及提供本毕业论文（设计）全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交毕业论文（设计）的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制毕业论文（设计）的部分或全部内容用于学术活动。

指导教师签名：

毕业论文（设计）作者签名：

2023 年 月 日

河南财政金融学院 毕业论文（设计）原创性声明

本人郑重声明：所呈交的毕业论文（设计），是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本毕业论文（设计）的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本毕业论文（设计）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位毕业论文（设计）原创性声明的法律责任由本人承担。

指导教师签名：

毕业论文（设计）作者签名：

2023 年 月 日

目录

毕业论文(设计)版权使用授权书	1
毕业论文(设计)原创性声明	1
摘 要	1
ABSTRACT	2
1 引言	3
1.1. 研究背景	3
1.2. 研究目的及意义	3
1.3. 国内外文献综述	4
2 系统分析	5
2.1 需求分析	5
2.1.1 功能需求	5
2.1.2 性能需求	5
2.1.3 环境需求	5
2.1.4 用户需求	5
2.1.5 系统数据流图及时序图	6
2.2 可行性分析	10
2.2.1 法律可行性	10
2.2.2 技术可行性	10
2.2.3 经济可行性	10
3 总体设计	11
3.1 系统结构设计	11
3.2 系统模块设计	11
3.3 数据库设计	12
3.4 算法设计	15
4 详细设计	16
4.1 用户登录模块及修改个人信息模块详细设计	16
4.2 货物录入模块详细设计	16
4.3 物流管理模块详细设计	16
4.4 物流可视化追踪模块详细设计	16
4.5 发布物流模块详细设计	17
4.6 智能物流模块详细设计	17
5 系统实现	18
5.1 用户登录及修改个人信息	18
5.2 货物录入	19
5.3 物流管理	21
5.4 发布物流	21
5.5 可视化追踪物流	23
5.6 智能物流	24

6 系统测试	26
6.1 测试环境	26
6.2 测试方法	27
6.3 测试用例	27
6.4 测试结论	41
6.4.1 用户登录及修改个人信息	41
6.4.2 货物录入	41
6.4.3 物流管理	42
6.4.4 发布物流	42
6.4.5 可视化追踪物流	42
6.4.6 智能物流	42
7 结论与展望	43
7.1 结论	43
7.2 展望	43
参考文献	44
致谢	45
附录	46

摘 要

本文旨在设计和实现一个基于地图 JavaScript GL API 的物流追踪系统，以解决物流追踪的准确性和效率问题。通过研究物流追踪的基本原理和需求，本文分析了当前物流追踪系统存在的问题和挑战，并提出了基于地图 JavaScript GL API 技术实现的解决方案。本文首先对 JavaScript GL API 技术进行了深入研究和分析，探讨了其特点、优势和应用。然后，基于该技术实现了物流追踪系统，并对系统进行了实际测试和验证。本文详细介绍了系统的设计和实现过程，阐述了使用的技术，给读者提供了一个完整的系统开发的案例，包括数据采集、数据处理、地图展示和用户交互等方面。通过对系统的实际应用和测试，本文评估了系统的性能和效果，并与其他常见的物流追踪系统进行了比较分析。研究表明，基于地图 JavaScript GL API 技术实现的物流追踪系统具有较高的准确性、实时性和可视化效果，能够满足不同物流公司的管理需求。同时，该系统还具有良好的用户体验和操作性，优化物流管理，提升其效率和经济效益。总的来说，本文的研究对于优化物流管理、提高物流追踪的准确性和效率具有重要的实际应用价值。此设计可作为其他物流领域的研究和实践的参考和借鉴，具有一定的指导意义和参考价值。然而，基于地图 JavaScript GL API 技术实现物流追踪系统仍存在一些局限性和挑战，未来研究可以进一步探索如何克服这些挑战，优化系统的设计和实现。

关键词：物流，追踪，JavaScript API GL，经济

ABSTRACT

This article aims to design and implement a logistics tracking system based on the Map JavaScript API GL to solve the accuracy and efficiency problems in logistics tracking. By studying the basic principles and requirements of logistics tracking, this article analyzes the problems and challenges of current logistics tracking systems, and proposes a solution based on the Map JavaScript API GL technology. This article first conducts in-depth research and analysis on the Map JavaScript API GL technology, discussing its characteristics, advantages, and applications. Then, based on this technology, a logistics tracking system is implemented and the system is tested and verified in practice. This article provides a detailed description of the system design and implementation process, explaining the technologies used and providing readers with a complete system development case, including data collection, data processing, map display, and user interaction. Through the practical application and testing of the system, this article evaluates the system's performance and effects and compares it with other commonly used logistics tracking systems. The research shows that the logistics tracking system based on the Map JavaScript API GL technology has high accuracy, real-time performance, and visualization effects, and can meet the management needs of different logistics companies. At the same time, the system also has good user experience and operability, optimizing logistics management and improving its efficiency and economic benefits. Overall, this research has important practical application value for optimizing logistics management and improving the accuracy and efficiency of logistics tracking. This design can serve as a reference and reference for other research and practices in the logistics field and has certain guiding significance and reference value. However, the implementation of logistics

tracking systems based on Map JavaScript API GL technology still has some limitations and challenges, and future research can further explore how to overcome these challenges and optimize the system design and implementation.

Keywords: logistics, tracking, JavaScript API GL, economy

1 引言

1.1. 研究背景

物流是中国经济发展迅速的代表行业之一，在如今时代，它无处不在，我们可以随时随地看到它的身影，为什么我们能在网上买的东西能在很短的时间里送到我们手中，都是因为非常多的物流企业存在，然而随着社会经济的发展，人们的购买能力的提升，物流企业信息管理面临着巨大的压力，如果还是照以前采用纸笔管理信息的方式，企业信息管理通常比较繁杂，企业工作的效率实在是太低，此时采用计算机来管理企业信息成为了一种趋势，计算机相比人为操作有更高的效率，更安全，也给企业工作人员降低了工作压力，工作管理人员只需要通过操作电脑，就可以管理企业的信息，但是我们如何通过计算机来管理企业信息呢，这时开发一个物流管系统成为了众多中小型物流企业的第一任务，它对于很多企业是发展中必不可少的一部。

1.2. 研究目的及意义

通过该毕业设计课题训练和规范的毕业设计撰写，试图掌握计算机科学与技术专业的基本知识和基本技能，能够运用软件工程思想，进行需求分析，并掌握软件工程的基本方法。研究了 WEB GIS 技术和车辆路径规划相关技术，有效的优化了物流资源配置，提高了物流配送信息化管理水平、物流配送效率和服务质量。该系统解决了实时追踪物流运营的问题，这对于高效及时地交付货物至关重要。随着电商和物流业务的不断增长，物流追踪系统成为了一个必不可少的应用。在当前国内外研究现状中，也有许多基于地图 API 的物流追踪系统，但它们多数采用基础地图 API，缺乏高级可视化功能，无法满足用户对交互性和实时性的需求。因此，本系统采用高德地图 API，利用其强大的可视化功能，提供更加友好和直观的用户界面，同时实现了实时物流追踪，使物流运营更加高效和便捷。具有一定的研究意义和实用价值。

1.3. 国内外文献综述

1. 基于移动 GIS 的共享物流系统的设计与实现

2019 年周文业《基于移动 GIS 的共享物流系统的设计与实现》^[1]，这篇文献介绍了一个基于移动 GIS 的共享物流配送管理系统的设计与实现，该系统结合共享物流的特点和我国货运物流现状，优化资源配置，提高物流配送效率和安全性。文献作者对货运物流现状和共享物流的创新模式进行了研究，提出了一种新的共享货运物流模式，设计和实现了共享物流管理系统。同时，作者分析了移动 GIS 的应用现状、相关理论和技术，搭建了一个服务于货运共享物流模式的移动 GIS 平台。此外，作者还对物流配送车辆路径问题进行了研究和分析，并结合共享货运物流的配送模式，设计实现了适合系统配送模式的车辆路径优化算法，通过优化算法有效的降低了物流配送成本，提高了货运物流配送的质量和效率。文献还进行了系统的测试，测试结果表明该系统功能和性能满足既定要求。

2. 二维码技术应用在农产品物流追溯系统中的分析和思考

2014 年李晶在现代农产品物流环节中^[2]，讨论了二维码技术在农产品物流追溯系统中的应用和优缺点。作者指出，建立农产品物流追溯系统是保障农产品质量安全的必要手段，而二维码技术能够为其提供技术支持。在详细分析了二维码技术的相关特点后，作者提出了农产品物流追溯系统中使用二维码技术可能会带来的问题^[3]，并对其在物流追踪领域的发展前景进行了预测。

3. 基于智慧供应链的电力物流跟踪系统

2019 年吴小力在物流跟踪系统^[4]中讲述了基于智慧供应链的电力物流跟踪系统。文章指出，智慧供应链是结合物联网技术和现代供应链管理理论、方法和技术，对促进电力物资管理水平提高和增益增效具有重要意义。其中，电力物流跟踪系统是电力智慧供应链的核心环节之一。为此，文章介绍了一种基于 RFID 的智慧电力物流跟踪系统，并对其进行了硬件和软件的详细设计。RFID 设备是该智慧电力物流跟踪系统的核心部件，由单片机、RFID 模块、隔离模块和 GPRS 模块组成。文章还介绍了 RFID 设备的硬件设计和软件流程图，说明了 RFID 设备的工作原理和操作流程。最后，文章提到该系统已经成功应用于地市级电网，运行结果证明了其有效性。

2 系统分析

2.1 需求分析

2.1.1 功能需求

1. 用户注册和登录功能: 用户需注册并登录后才能使用本系统的物流追踪服务。
2. 物流信息录入: 管理员可以录入物流信息, 包括物流单号、物品名称、发货人信息、收货人信息等。
3. 物流信息查询: 用户可以通过输入物流单号等信息查询物流信息, 包括物品当前位置和运输状态等。
4. 物流信息更新: 管理员可以更新物流信息, 包括物品当前位置和状态等。
5. 物流信息展示功能: 系统可以将查询到的快递物流信息以地图、列表、图表等多种方式进行展示。

2.1.2 性能需求

1. 系统的响应时间: 查询和展示快递物流信息需要在 1 秒内完成。
2. 并发访问量: 系统需要支持至少 100 个并发请求。
3. 系统稳定性: 系统需要保持 24 小时的稳定性, 确保用户可以随时使用系统。

2.1.3 环境需求

1. 部署环境: 部署在云服务器上, 保证系统的稳定性和可靠性。
2. 操作系统: Windows、Linux。
3. 数据库: MySQL 数据库管理系统。

2.1.4 用户需求

1. 易用性: 系统需要简单易用, 不需要太多的操作步骤。

2. 实时性:系统需要提供实时更新的物流信息,保证用户可以及时了解快递的运输情况。
3. 可靠性:系统需要提供准确、及时的物流信息。

2.1.5 系统数据流图及时序图

1. 数据流图

顶层数据流图:

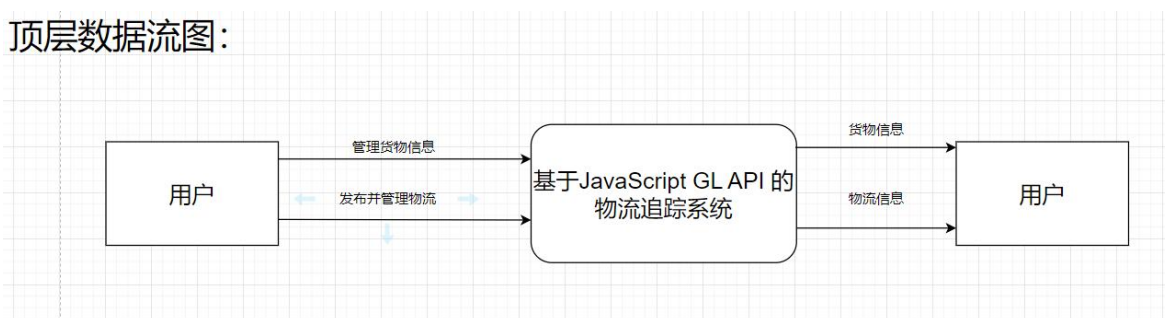


图 2 顶层数据流图

0层数据流图:

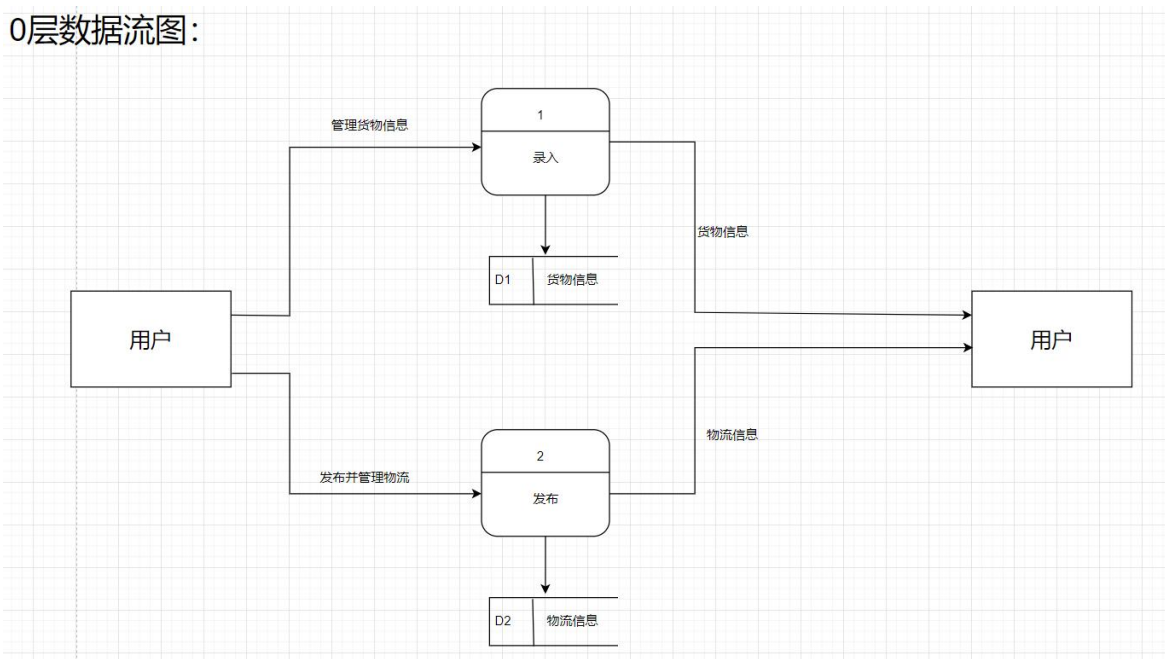


图 3 0层数据流图

1层数据流图：

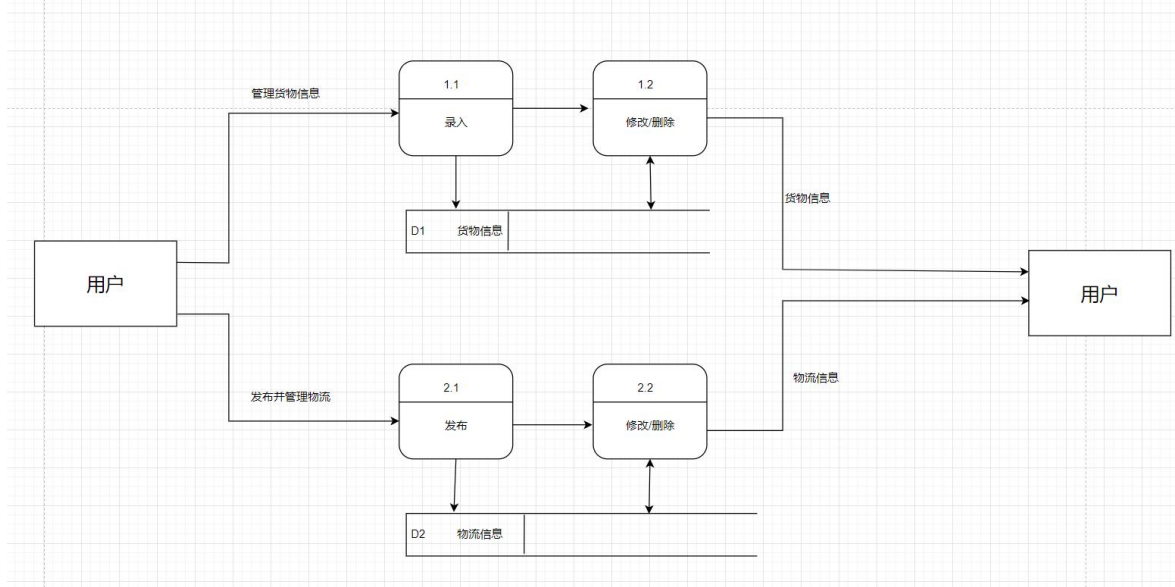


图 4 1 层数据流图

2. 时序图

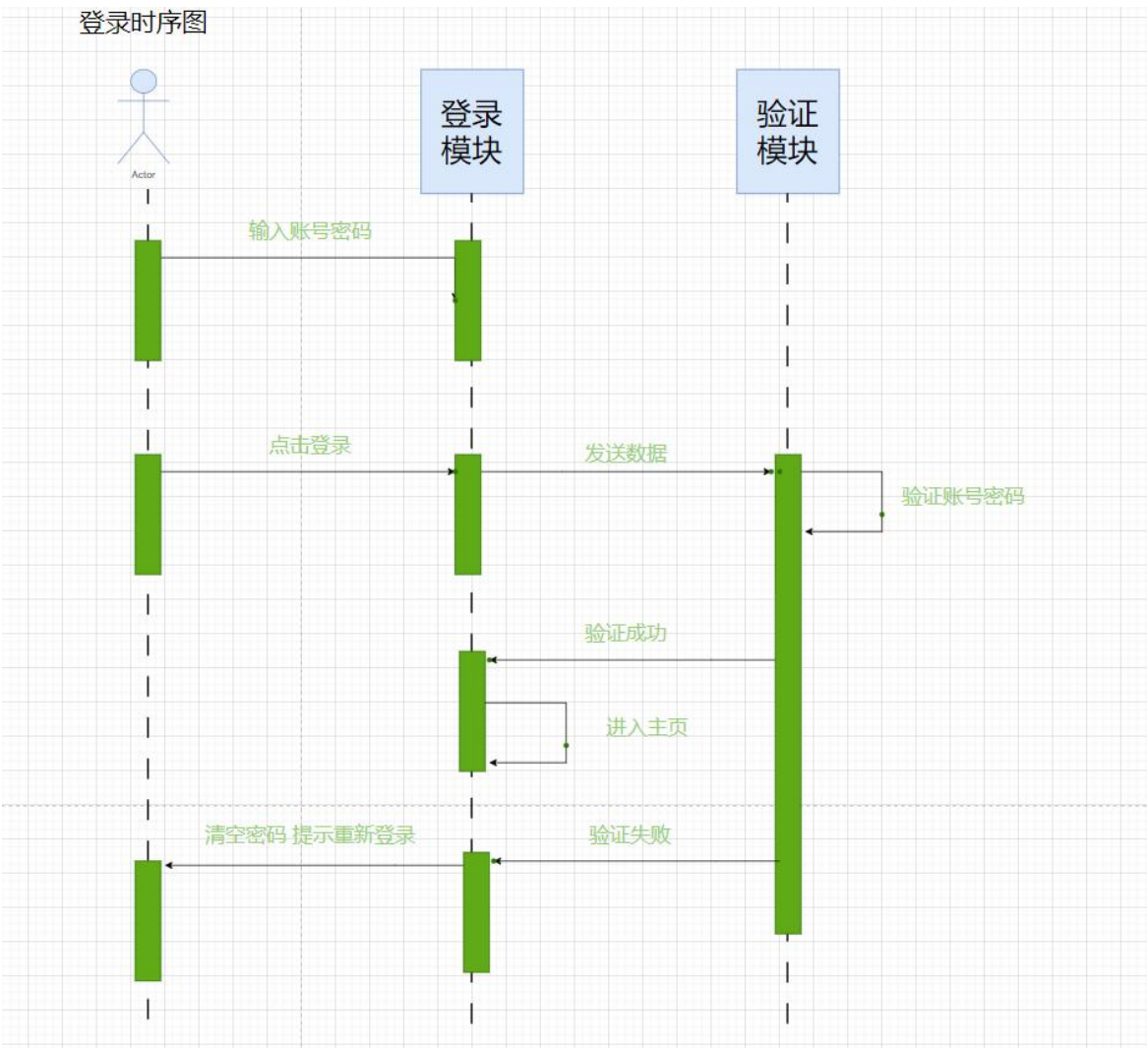


图 5 登录时序图

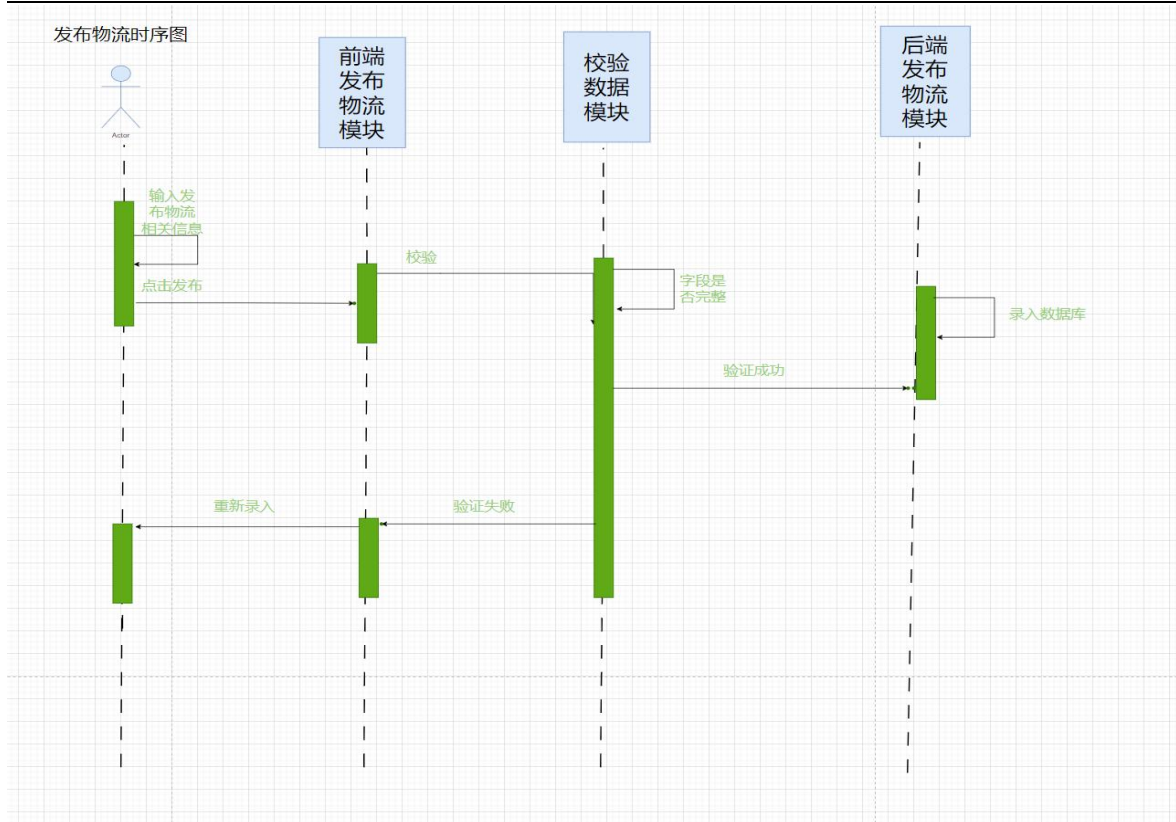


图 6 发布物流时序图

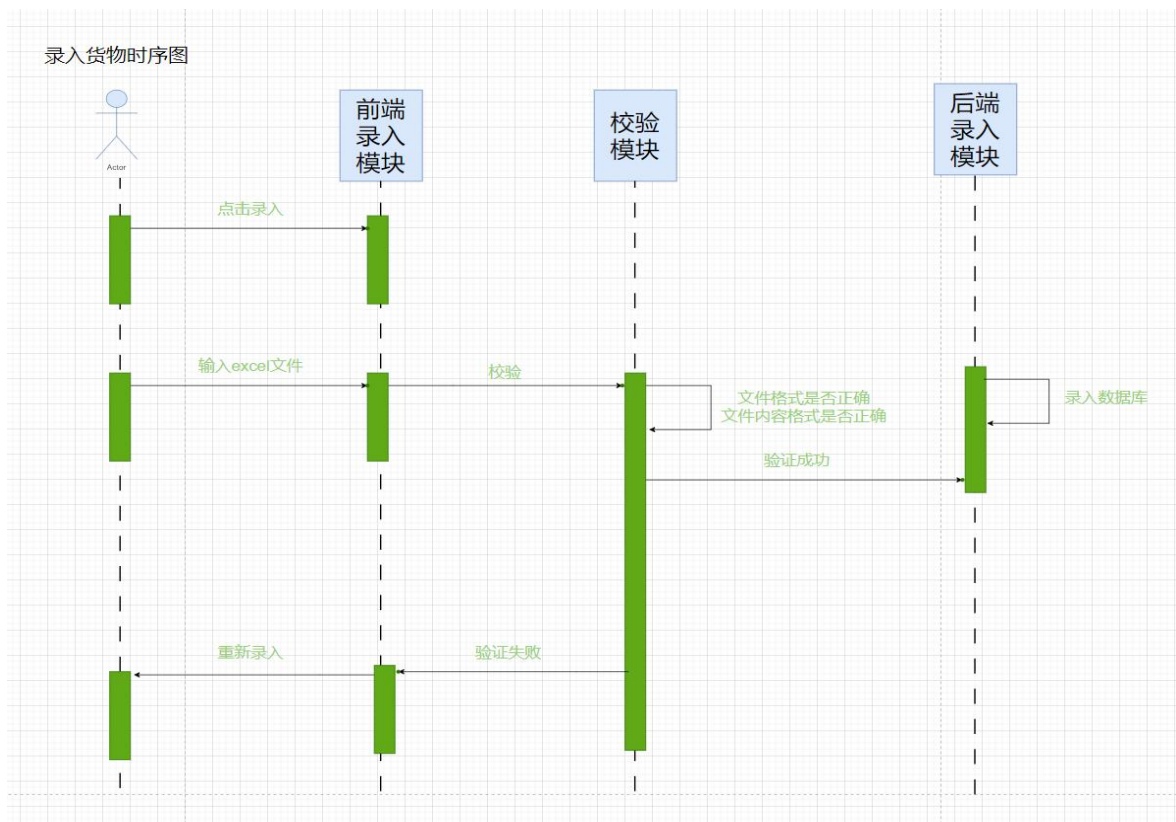


图 7 录入货物时序图

2.2 可行性分析

2.2.1 法律可行性

本系统需要遵守相关的法律法规，包括数据保护法律、互联网安全法律等。同时，还需要获得高德地图 API 的授权使用。因此，在法律可行性上，本系统需要严格遵守相关法律法规，确保系统的合法性和安全性。

2.2.2 技术可行性

本系统采用了流行的前后端分离技术，使用 Node.js 作为后端开发语言，采用 Koa 作为后端框架，使用 Vue3 作为前端框架。同时，为实现实时物流追踪，采用了高德地图 API。这些技术都是成熟的技术，有丰富的文档和支持，因此在技术可行性上不存在问题。

2.2.3 经济可行性

本系统的开发需要一定的经费投入，包括开发人员的工资和系统所需的硬件设备和软件授权等。同时，该系统的实施将会带来一定的经济效益，能够提高物流运营的效率 and 准确性，通过优化物流流程，减少物流成本，提高物流效率和服务质量，增强物流企业在市场上的竞争力。因此，在经济可行性上，本系统具有一定的优势。

3 总体设计

3.1 系统结构设计

该物流追踪系统采用前后端分离的架构，前端主要使用 Vue3 技术栈开发，后端采用 Node.js 和 Koa 框架，数据库使用 MySQL，缓存使用 Redis，地图 API 采用第三方高德地图 GL JS API。具体的系统结构设计如下：

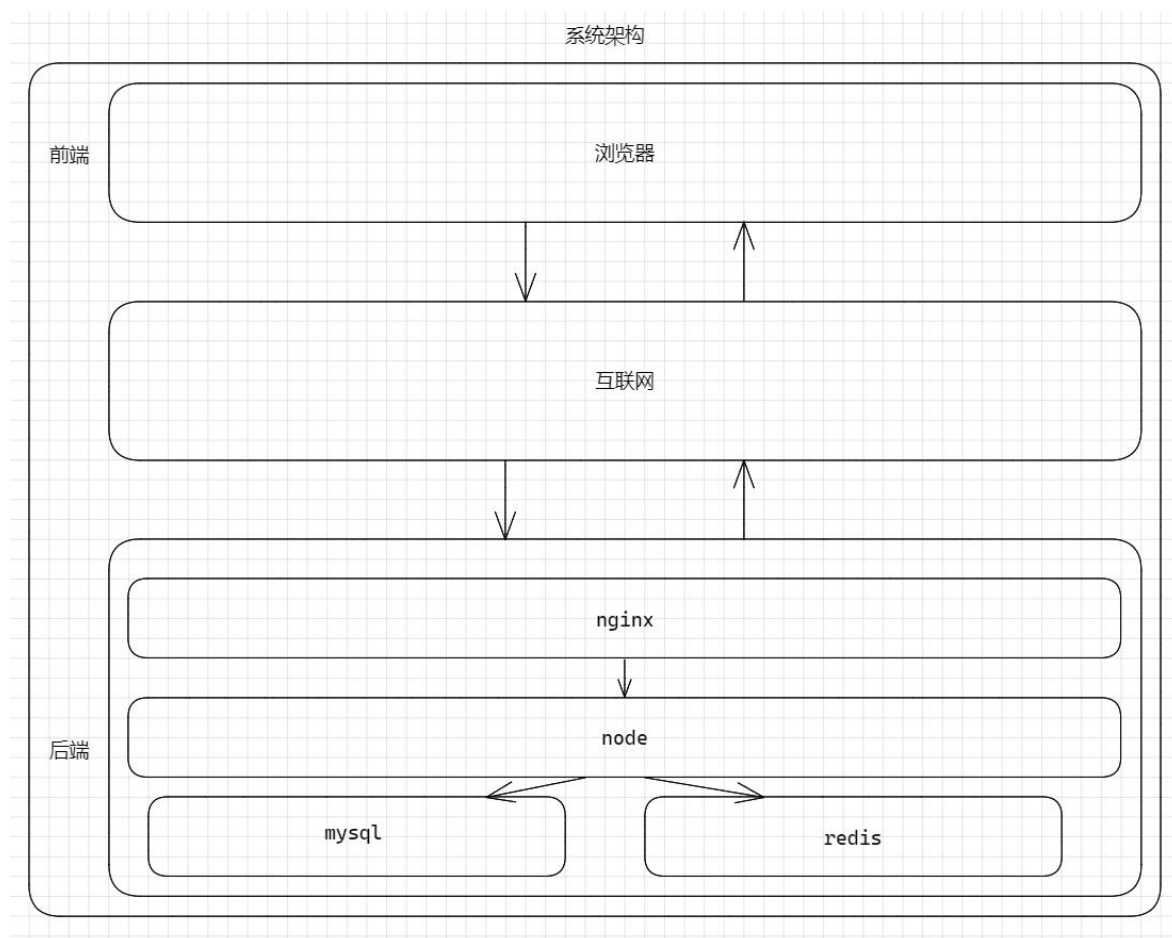


图 8 系统架构图

3.2 系统模块设计

该物流追踪系统的分为六个模块：

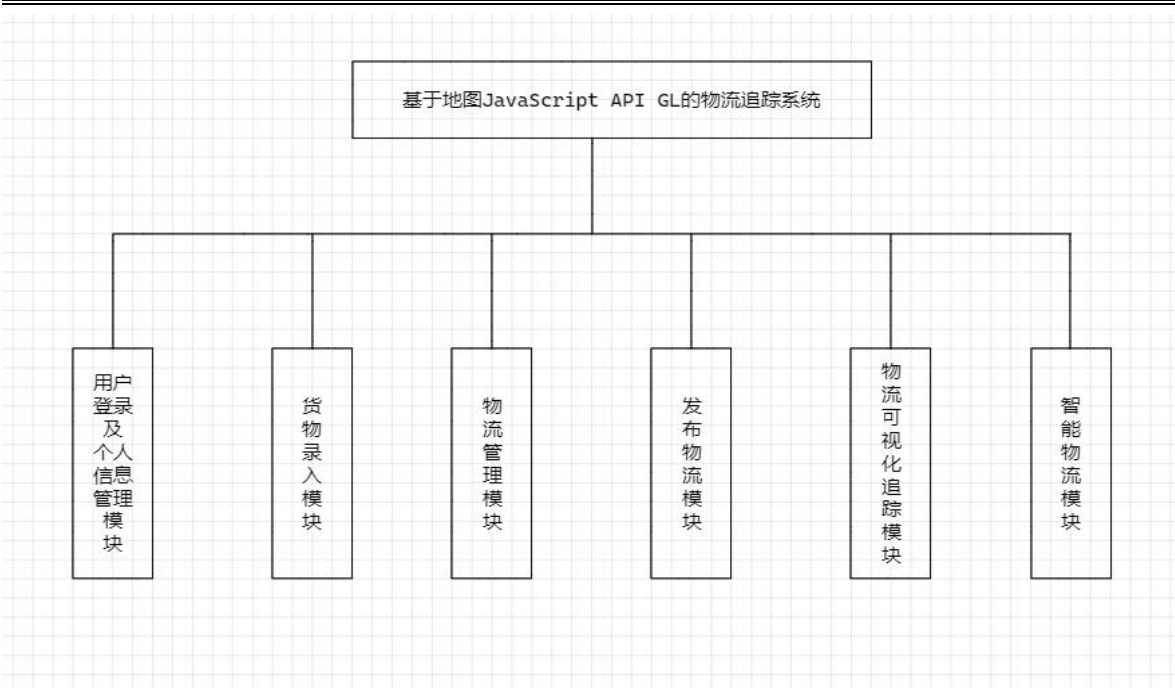


图9 模块设计图

- 1. 用户登录及个人信息管理模块：实现用户登录、注册、注销等功能，对用户进行身份认证。
- 2. 物流管理模块：实现物流查询、添加、修改和删除等功能，对物流进行数据操作和处理。
- 3. 物流可视化追踪模块：实现物流状态查询和展示，显示订单的实时物流轨迹，支持地图展示和路径规划。实现物流的多种可视化追踪
- 4. 发布物流模块:在系统内选择物品进行发布物流
- 5. 智能物流模块：一个智能物流机器人，可以向它询问某个物流的具体状态和信息。
- 6. 货物录入模块：录入货物信息，可以通过导入 excel 文件的方式导入。

3.3 数据库设计

- 1. 用户表

表 1

字段名	类型	长度	描述
id	int	10	用户 id
account	varchar	11	用户账号

uname	varchar	20	用户名
password	varchar	20	用户密码
img	varchar	30	用户图像
address	varchar	50	用户地址
address_geo	varchar	20	用户地理坐标
create_time	timestamp		创建时间

2. 物流表

表 2

字段名	类型	长度	描述
id	int	10	物流 id
uid	varchar	10	发货人 id
thing_id	int	20	用户名
current_time	timestamp		到达最新位置的时间
current_position	varchar	30	到达最新位置的地点
start_position	varchar	50	发货地点
end_position	varchar	20	收获地点
current_position_geo	varchar	20	到达最新位置的地理坐标
start_position_geo	varchar	20	发货位置的地理坐标
end_position_geo	varchar	20	收获位置的地理坐标
toName	varchar	20	收获人名字
toPhone	varchar	11	收获人手机号码
time	timestamp		时间
qr_code	varchar	20	物流二维码
status	int	10	物流状态

3. 物流位置记录表

表 3

字段名	类型	长度	描述
id	int	10	位置表 id

pid	varchar	11	物流 id
current_time	varchar	20	到达当前位置时间
current_position	varchar	20	到达当前位置地点名称
current_position_geo	varchar	30	到达当前位置地理坐标

4. 物品表

表 4

字段名	类型	长度	描述
id	int	10	物品 id
uid	varchar	11	用户 id
name	varchar	20	物品名称
price	varchar	20	物品价格
total	varchar	30	总量
count	Int	20	剩余数量

5. E-R 图

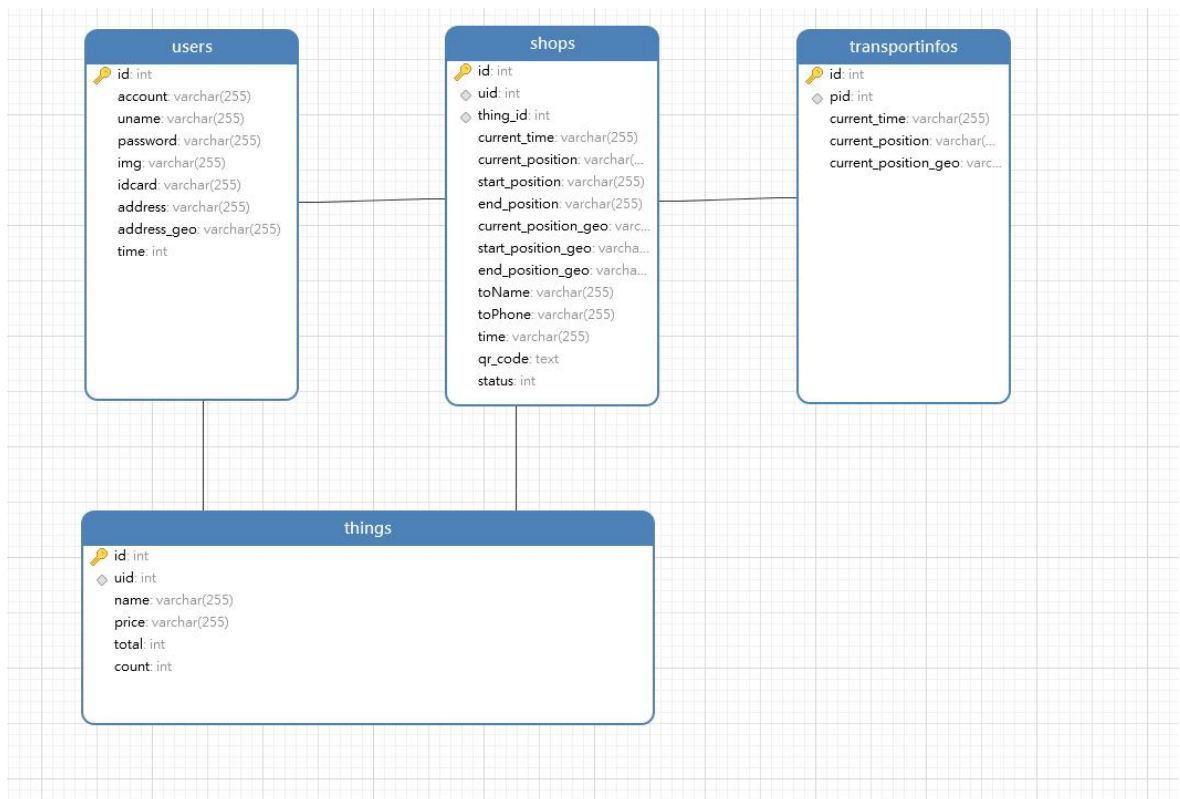


图 10 E-R 图

3.4 算法设计

1. 路径规划算法：高德地图 GL API 支持路径规划，可以根据起点、终点、途经点和交通方式等信息生成最优路径。算法设计的关键是根据不同的场景和需求选择合适的路径规划算法，例如 Dijkstra 算法、最短路算法等。
2. 可视化算法：高德地图 GL API 提供了丰富的地图可视化功能，可以展示地图数据、路线、标记点等。在设计可视化算法时，需要考虑数据的类型和数量、展示方式和效果等因素。例如，在展示路线时，可以使用平滑曲线算法来优化路线的显示效果

4 详细设计

4.1 用户登录模块及修改个人信息模块详细设计

1. 用户名和密码验证模块：通过输入用户名和密码进行验证，确保用户的身份准确性。
2. 用户信息存储模块：将用户的登录信息保存到数据库中，便于后续的查询和修改。
3. 用户信息修改模块：允许用户修改个人信息，包括用户名、密码、收获等。

4.2 货物录入模块详细设计

1. 货物导入模块:通过解析 excel 文件，将货物导入系统
2. 货物信息存储模块：将货物的信息保存到数据库中，方便后续的查询和管理。

4.3 物流管理模块详细设计

1. 货物信息输入模块：允许用户输入货物的基本信息，包括货物名称、数量、起始地、目的地、收货人、联系电话等。
2. 货物状态设置模块：用户可以根据货物的状态，设置货物的状态，如待发货、已发货、运输中、已送达等。
3. 货物信息存储模块：将货物的信息保存到数据库中，方便后续的查询和管理。

4.4 物流可视化追踪模块详细设计

1. 地图展示模块：物流管理系统可以通过地图展示，直观地显示货物的运输路线和状态。
2. 实时更新模块：物流管理系统可以实时更新货物的位置和状态，以便用户随时了解货物的最新情况。

3. 数据可视化模块：通过数据可视化技术，物流管理系统可以将货物的信息、状态等数据以图形化的方式呈现出来，方便用户快速了解货物的情况。

4.5 发布物流模块详细设计

1. 发布货物信息模块：用户可以将货物的信息发布到物流管理系统中，以便更好地管理和跟踪。

2. 生成物流单据模块：物流管理系统可以根据用户的要求，自动生成物流单据，二维码等方便用户进行打印和管理。

4.6 智能物流模块详细设计

1. 实现一个智能机器人，向机器人发送物流编号，机器人给你返回对应的物流信息。

5 系统实现

5.1 用户登录及修改个人信息

该功能实现了用户账号的登录和个人信息的修改，包括用户名、密码、联系方式等。改模块的部分代码如图 11 流程图如图 12

```
class LoginController {  
  async login(ctx: any, next: any) {  
    const { account, password, type } = ctx.request.body;  
    if (type == 1) {  
      ctx.body = await loginService.login(account, password);  
    } else {  
      ctx.body = await loginService.adminLogin(account, password);  
    }  
  }  
  
  async register(ctx: any, next: any) {  
    const { username, password } = ctx.request.body;  
    ctx.body = await loginService.register(username, password);  
  }  
  
  async change(ctx: any) {  
    const data = ctx.request.body;  
    ctx.body = await loginService.change(ctx.userInfo.id, data);  
  }  
}
```

图 11 登录模块部分代码

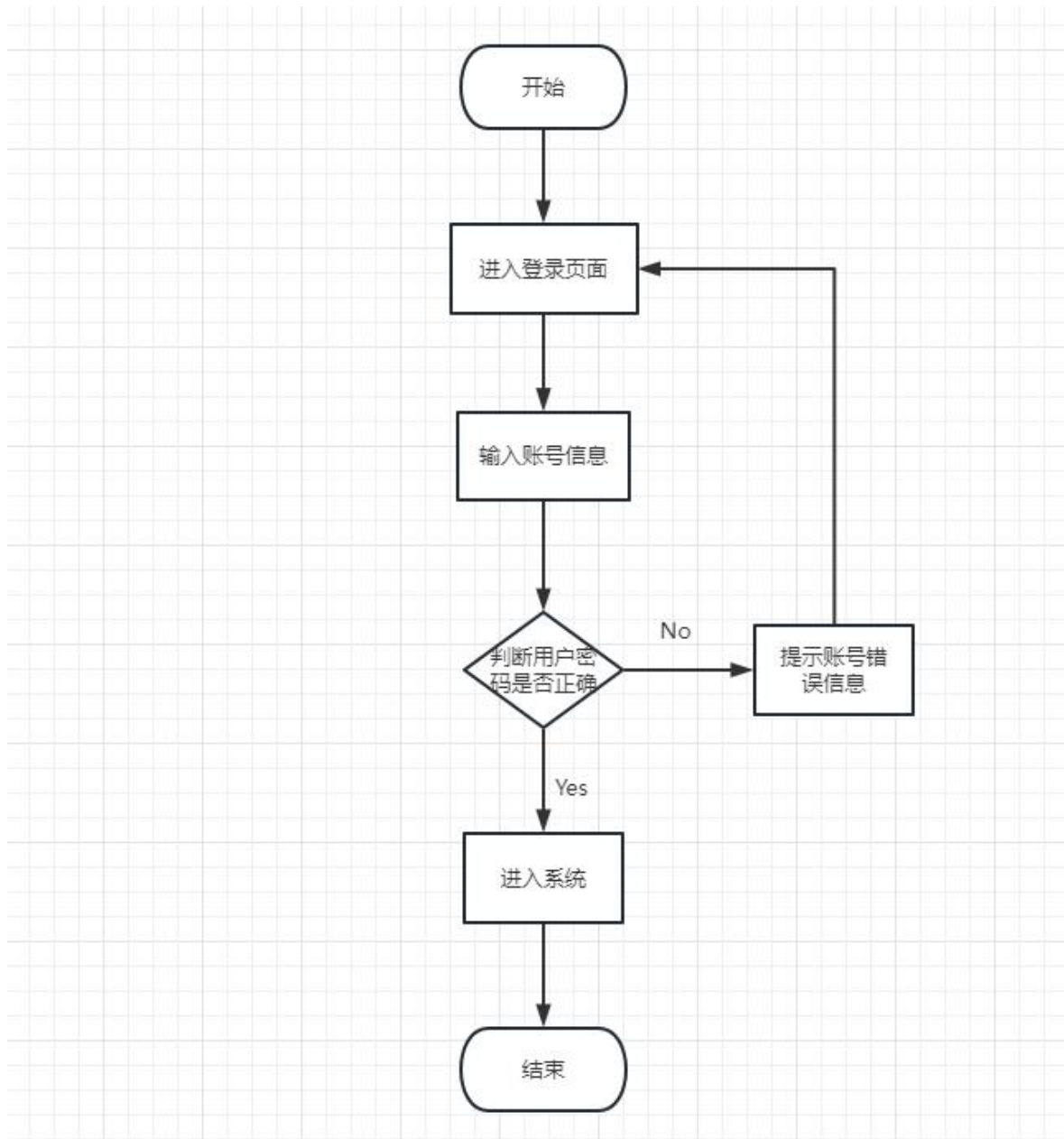


图 12 登录模块流程图

5.2 货物录入

该功能允许用户录入货物信息，包括货物名称、数量、价格等。该模块部分代码如图 13 流程图如图 14

```
async createThings(uid: any, data: any) {  
  const { name, price, total } = data;  
  try {  
    for (let i = 0; i < name.length; i++) {  
      await Thing.create({  
        uid: uid,  
        name: name[i],  
        price: price[i],  
        total: total[i],  
        count: total[i],  
      });  
    }  
  } catch (error) {  
    return { code: 200, msg: error };  
  }  
  
  return { code: 200, msg: '导入成功' };  
}
```

图 13 货物录入代码

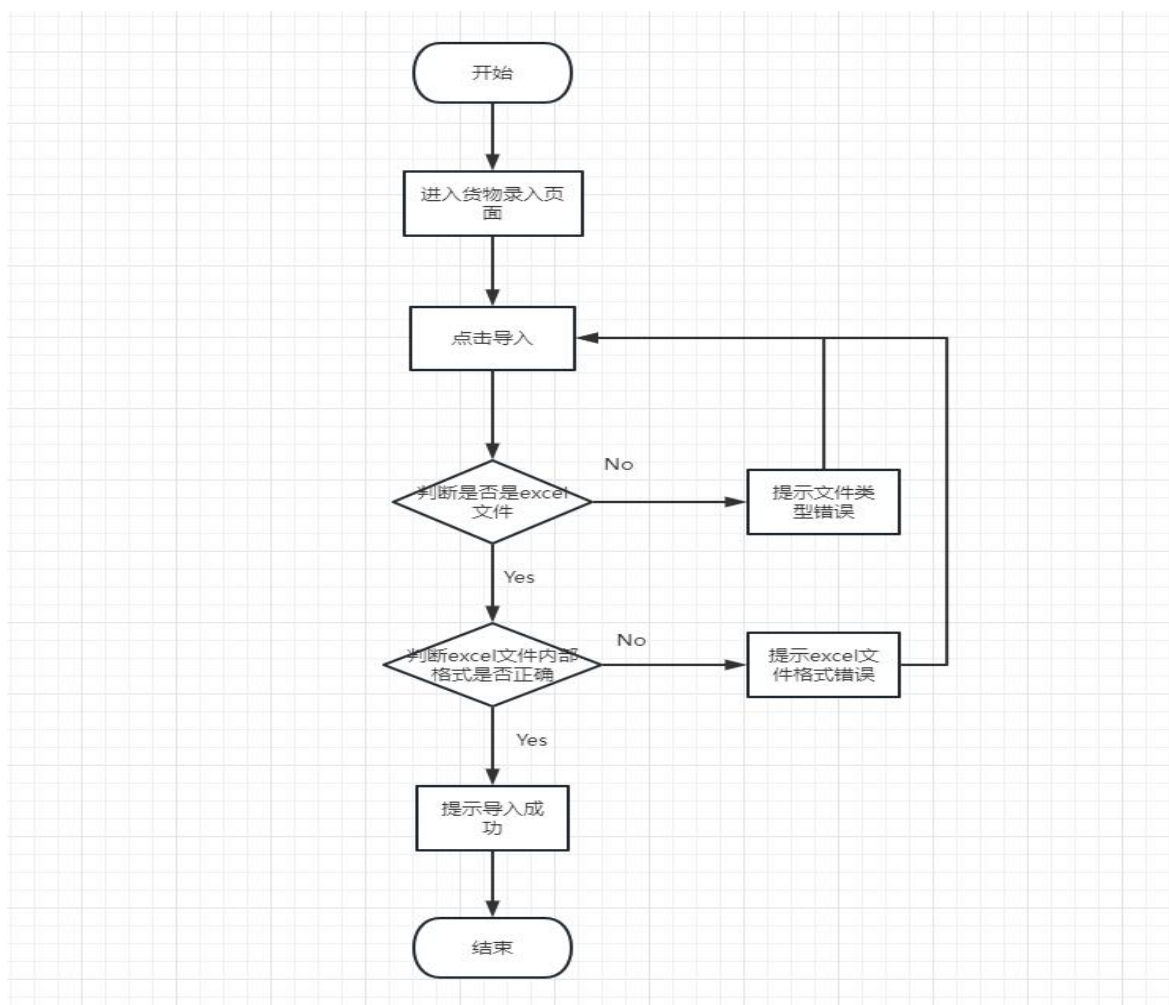


图 14 货物录入模块流程图

5.3 物流管理

该功能实现了对已发货物的管理，包括货物的状态跟踪、物流路线的规划和调整、状态、拦截、删除、更新等操作。该模块部分代码如图 15

```
// 删除
async delete(ctx: Context, next: any) {
  const { id } = ctx.query;
  const res = await shopService.delete(id);
  ctx.body = res;
}

// 更新地址
async updateAddress(ctx: Context, next: any) {
  const { id, lng, lat, current_position } = ctx.request.body;

  ctx.body = await shopService.updateAddress({ id, lat, lng, current_position });
}

// 获取物流详情信息
async getInfoById(ctx: Context, next: Next) {
  const { id } = ctx.request.body;
  const res = await shopService.getInfoById(id);
  ctx.body = res;
}

// 添加一个物流运输信息
async addATransportInfo(ctx: Context, next: Next) {
  const data = ctx.request.body;
  const res = await shopService.addATransportInfo(data);
  ctx.body = res;
}

// 查询
async getTransportInfo(ctx: Context) {
  const { id } = ctx.query;
  const res = await shopService.getTransportInfo(id);
  ctx.body = res;
}
```

图 15 物流管理代码

5.4 发布物流

该功能允许用户发布货物信息和物流需求，包括货物类型、起点、终点、时间、价格等。该模块部分代码如图 16 流程图如图 17

```
// 创建物流
async createShop(ctx: Context, next: any) {
  const { id } = ctx.userInfo;

  const resulte = await shopService.createShop({ ...ctx.request.body, uid: id });
  ctx.body = resulte;
}

async createShop(shopInfo: any) {
  console.log(shopInfo);

  const res: any = await ShopModel.create({
    ...shopInfo,
  });

  let qr_code = await QRCode.toDataURL(String(res.dataValues.id));
}
```

图 16 物流发布代码

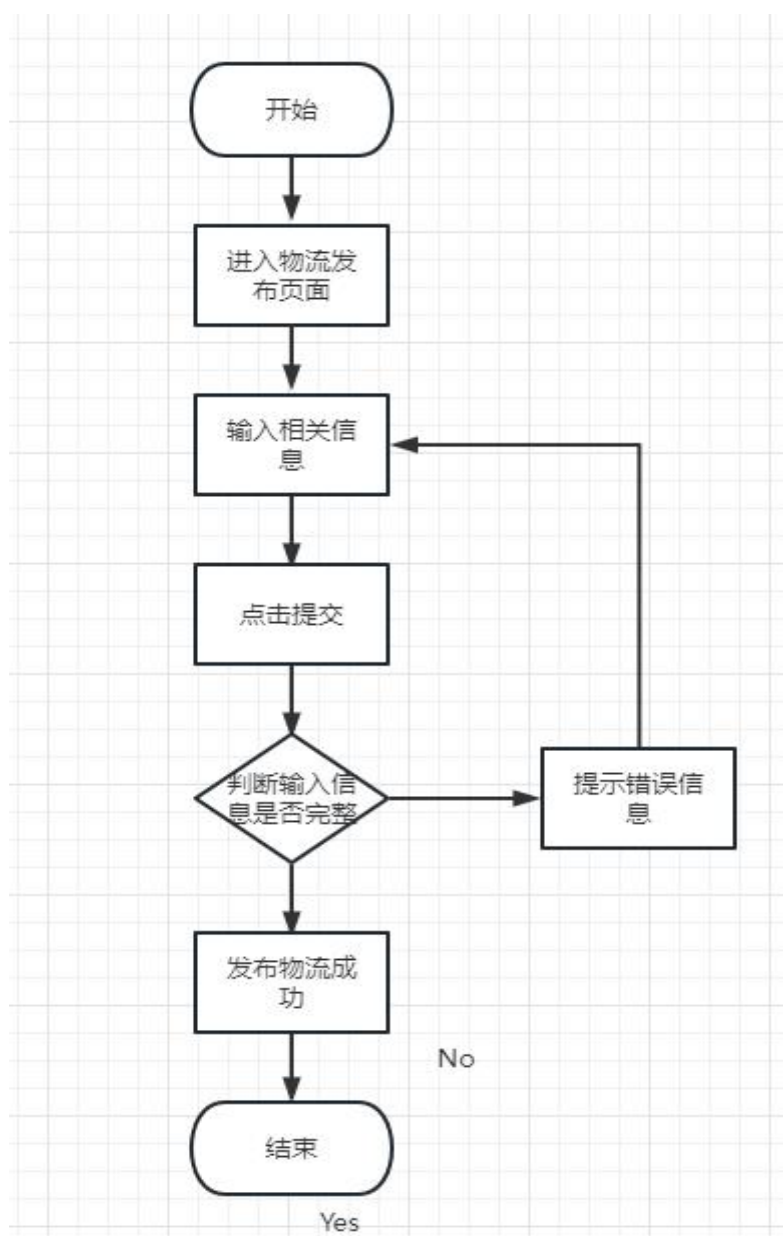


图 17 物流发布代码

5.5 可视化追踪物流

该功能允许用户对已发布的物流订单进行可视化追踪，包括货物的实时位置、运输情况等。该模块主要是展示功能,没有流程图。

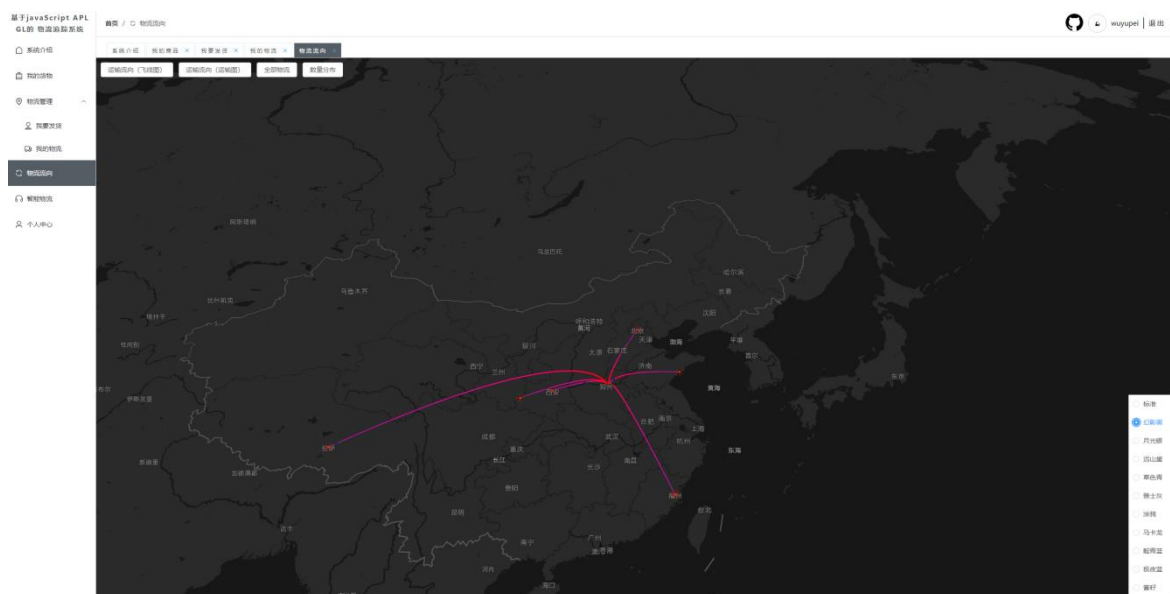


图 18 物流可视化追踪(飞线图)页面图

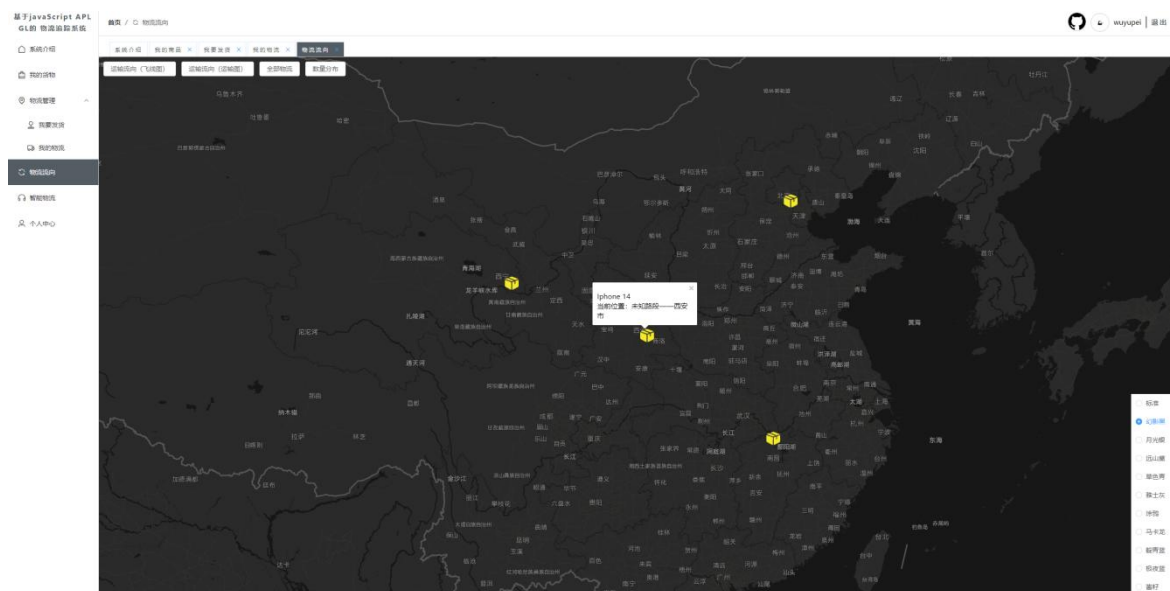


图 19 物流可视化追踪(物流分布)页面图

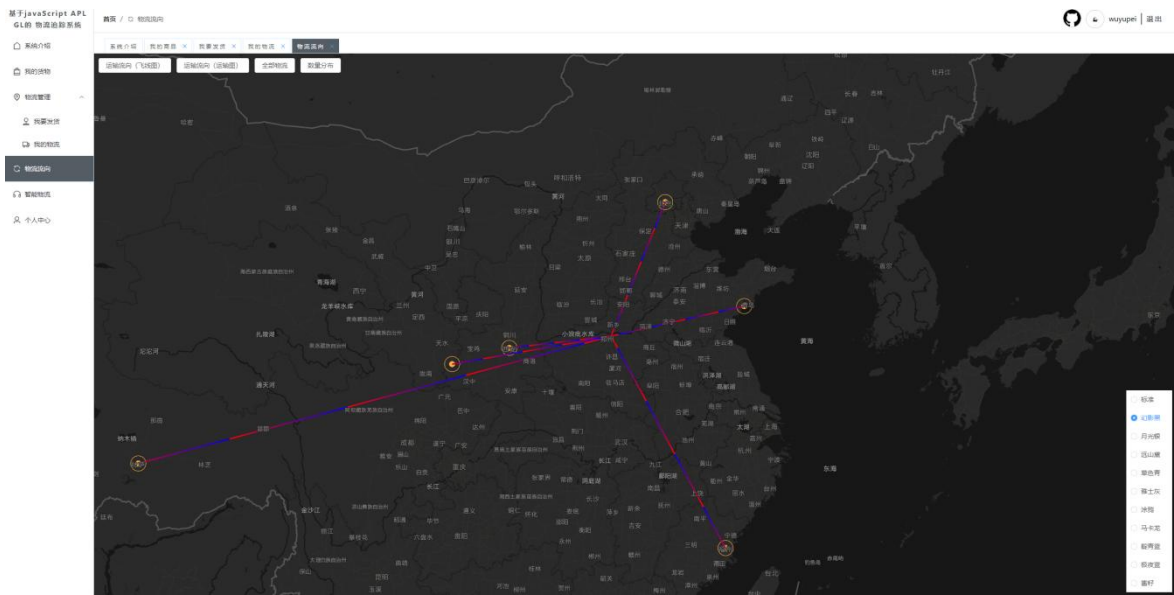


图 20 物流可视化追踪(流向)页面图

5.6 智能物流

该功能允用户根据物流订单号，查询物流信息。该模块部分代码如图 21 具体流程图如图 22

```
async getInfoById(ctx: Context, next: Next) {
  const { id } = ctx.request.body;
  const res = await shopService.getInfoById(id);
  ctx.body = res;
}
async getInfoById(id: any) {
  const res = await ShopModel.findOne({
    where: {
      id,
    },
  });
  if (!res) {
    return [];
  } else {
    return [res];
  }
}
```

图 21 智能物流代码

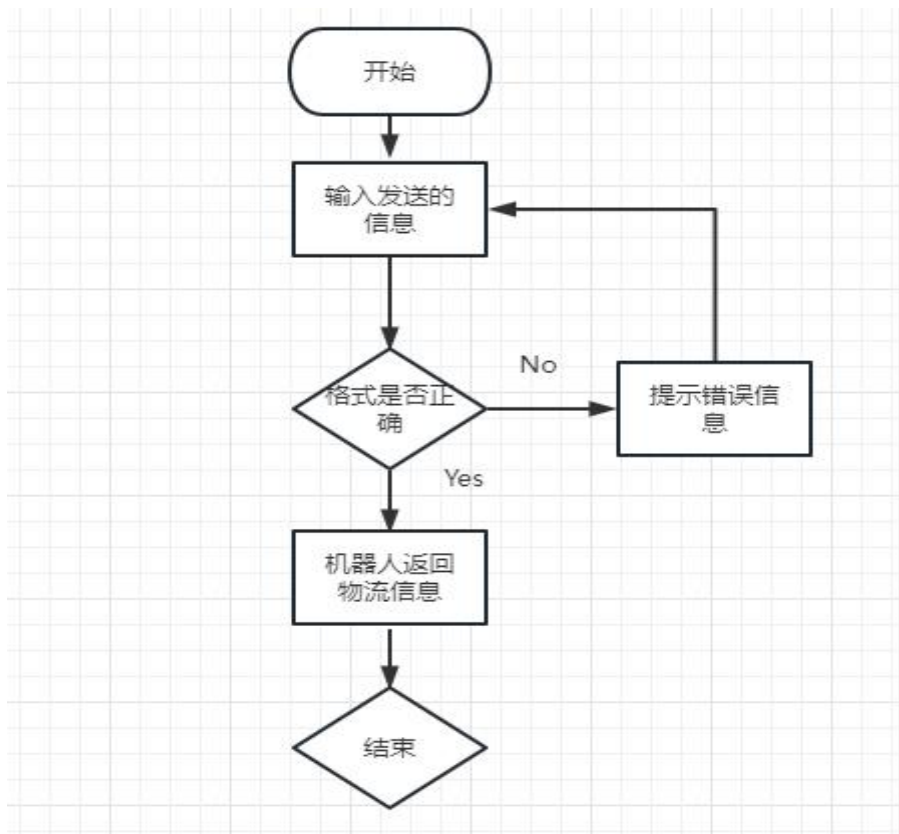


图 22 智能物流流程图

6 系统测试

6.1 测试环境

1. 数据库环境

表 5

硬件环境	服务器-cpu	2 核 1.00Ghz
	服务器-内存	2GB
软件环境	数据库:	MySQL_64bit
	数据库服务器 IP:	8.141.63.127
	数据库操作系统	CentOS 7.6 64Bit
	数据库实例名:	lch_db
	数据库用户:	root

2. 应运环境

表 6

硬件环境	服务器-cpu	2 核 1.0Ghz
	服务器-内存	2GB
软件环境	中间件:	Nginx
	数据库服务器 IP:	8.141.63.127
	数据库操作系统	CentOS 7.6 64Bit

3. 客户端环境

表 7

硬件	客户端-cpu	I7-7700HQ 四核 2.8Ghz
	客户端-内存	16GB
软件	操作系统	Windows10-64bit
	浏览器	IE10、EDGE、Chrome、360
网络环境	互联网	100/1000M 以太网

6.2 测试方法

根据开题报告中的需求，按照预定的测试计划进行测试，验证系统是否符合需求和功能是否可以正常使用。

6.3 测试用例

1. 用户登录及修改个人信息

表 8

测试功能点：用户登录及修改个人信息	
运行权限： 无	
测试初始条件：具有登录系统的账号	
<u>使用操作：</u> 1. 打开浏览器，输入网址 http://127.0.0.1:5173 2. 使用账号密码登录 3. 点击登录	<u>预期输出：</u> 1. 密码输入正确，进入系统 2. 密码输入错误，提示登录失败 3. 修改个人信息
备注： 登录失败截图：	

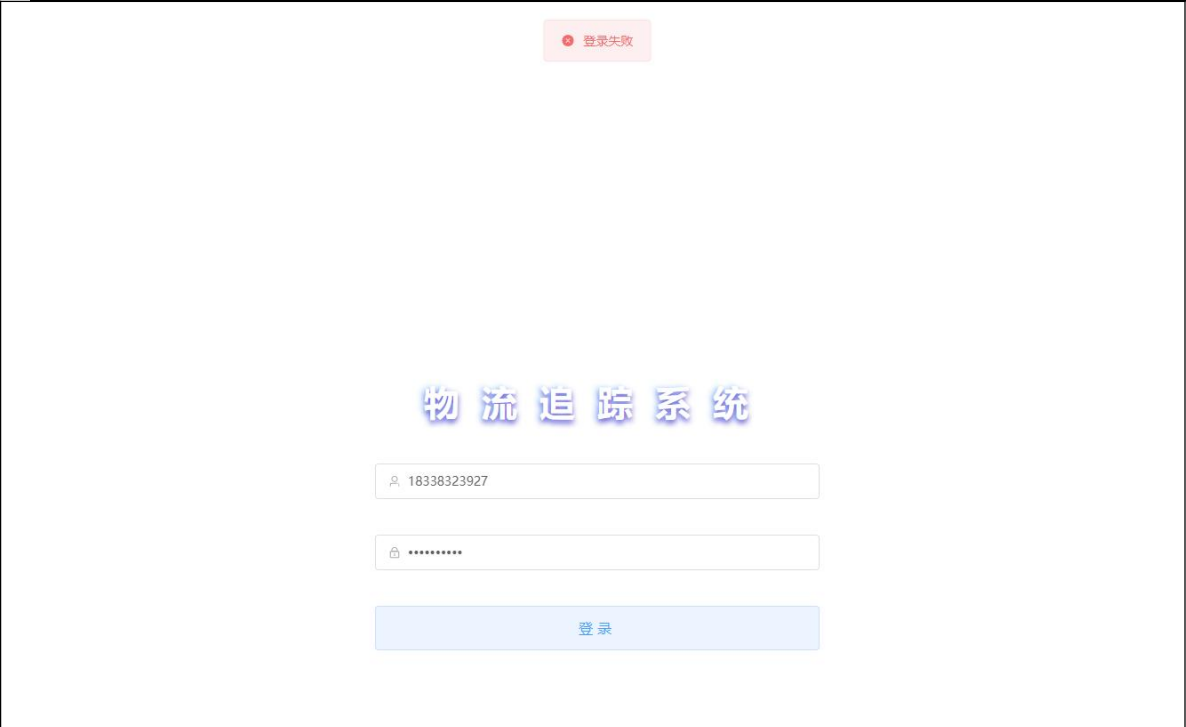


图 23 登录失败测试图

登录页面

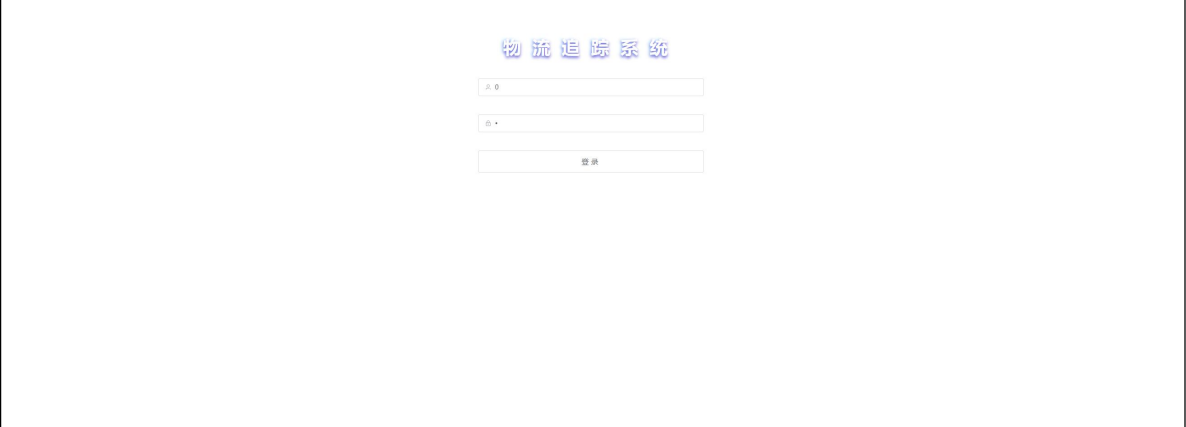


图 24 登录成功测试图

登录成功后进入系统界面截图

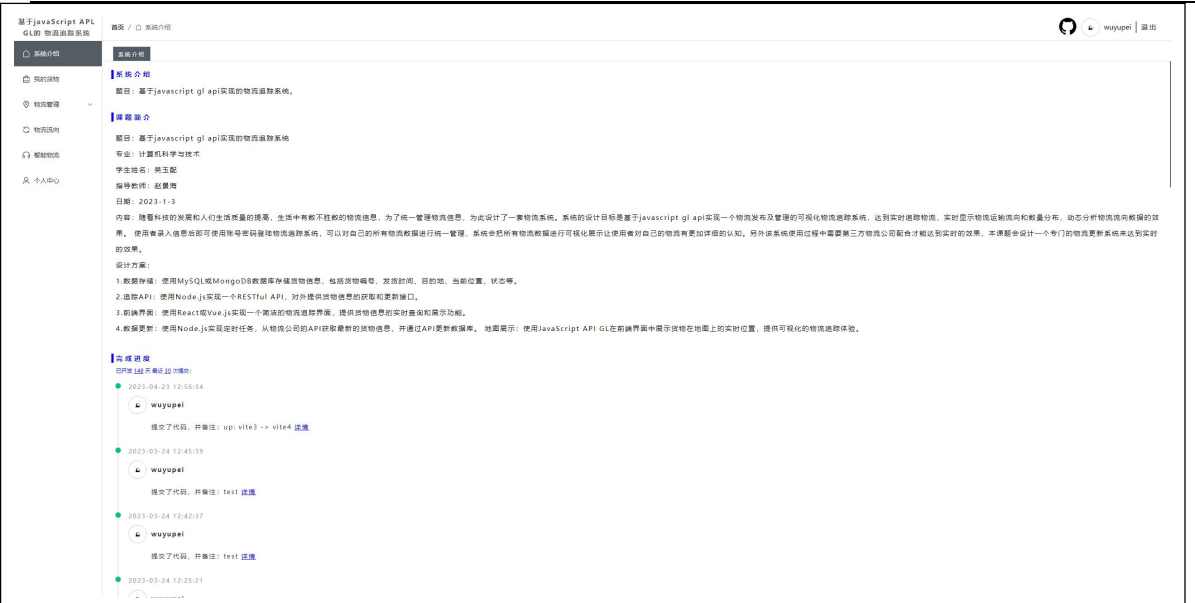


图 25 登录成功测试图

修改个人信息页面



图 26 修改个人信息图测试图

1 分钟内不可重复修改

系统介绍

我的货物

物流管理

我要发货

我的物流

物流流向

智能物流

个人中心

系统介绍

个人中心

个人中心

姓名:

wuyupe

发货地址:

请选择地址

联系电话:

0

操作:

1秒后可修改

图 27 修改个人信息图测试图

测试结果及分析：同预期输出是否一致： 是（√）否（ ） 测试人：吴玉配

2. 录入货物

表 9

测试功能点：录入货物	
运行权限： 无	
测试初始条件：具有登录系统的账号	
<div>使用操作：</div> <div>1. 打开浏览器，输入网址</div> <div>http://127.0.0.1:5173</div> <div>2. 使用账号密码登录</div> <div>3. 登入系统，点击我的货物，进行查看，修改，删除</div>	<div>预期输出：</div> <div>1. 查看用户在平台中的货物</div> <div>2. 录入货物</div> <div>3. 编辑货物信息</div> <div>4. 修改货物信息</div>

备注：

查看商品截图：

系统介绍

我的商品

我的商品

id	商品名称	现有库存	价格	操作
0	lphone	10	899	<div>编辑删除</div>
5	小米	50	6999	<div>编辑删除</div>
6	华为	30	7999	<div>编辑删除</div>

图 28 查看商品测试图

导入商品截图：

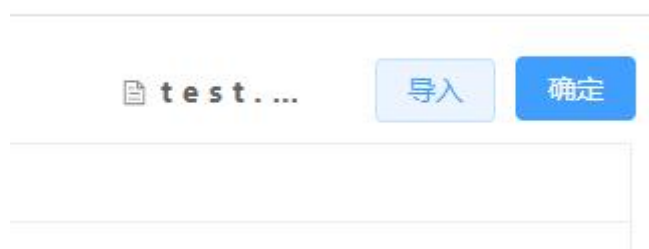


图 29 导入商品测试图



图 30 导入成功测试图



图 31 导入后测试图

修改商品截图:



图 32 编辑测试图

修改成功截图

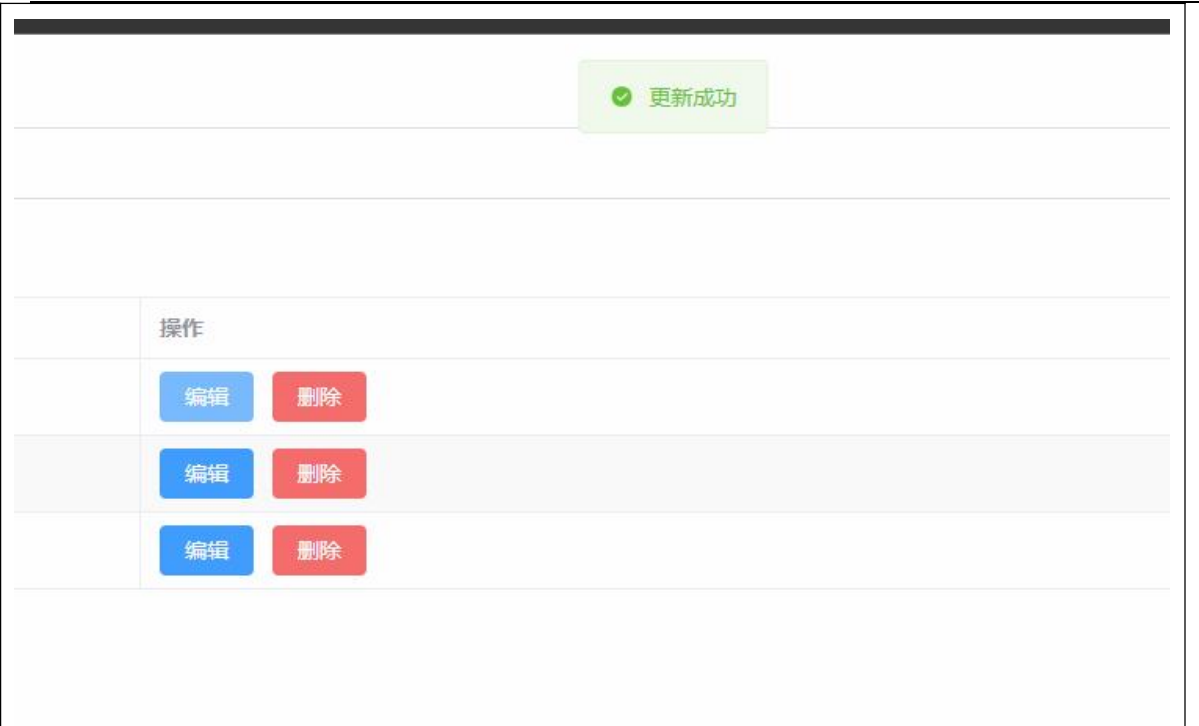


图 33 编辑成功测试图

删除成功截图:



图 34 删除成功测试图

测试结果及分析：同预期输出是否一致： 是（√）否（ ） 测试人：吴玉配

3. 发布物流

表 10

测试功能点：发布物流

运行权限： 无	
测试初始条件： 具有登录系统的账号	
<p>使用操作：</p> <p>1. 打开浏览器，输入网址</p> <p>http://127.0.0.1:5173</p> <p>2. 使用账号密码登录</p> <p>3. 登入系统，点击我的货物，进行查看，修改，删除</p>	<p>预期输出：</p> <p>1. 输入完整的相关信息，点击发货</p> <p>2. 信息输入完整，发货成</p> <p>3. 信息数不完整提示发货失败，请重新输入信息</p>

备注：
输入信息部分截图：



图 35 输入信息测试图

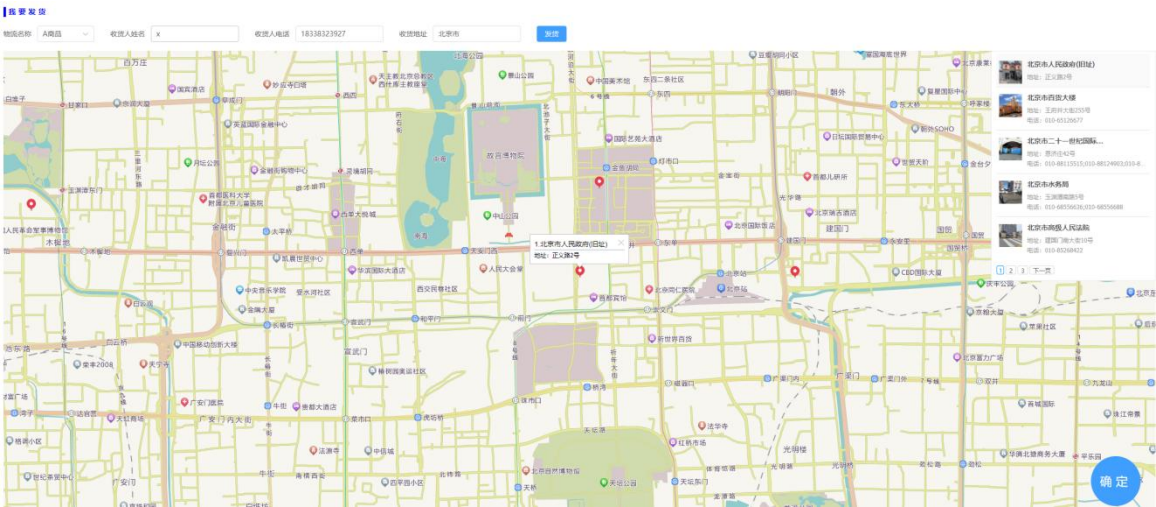


图 36 输入信息测试图

信息输入不完整截图

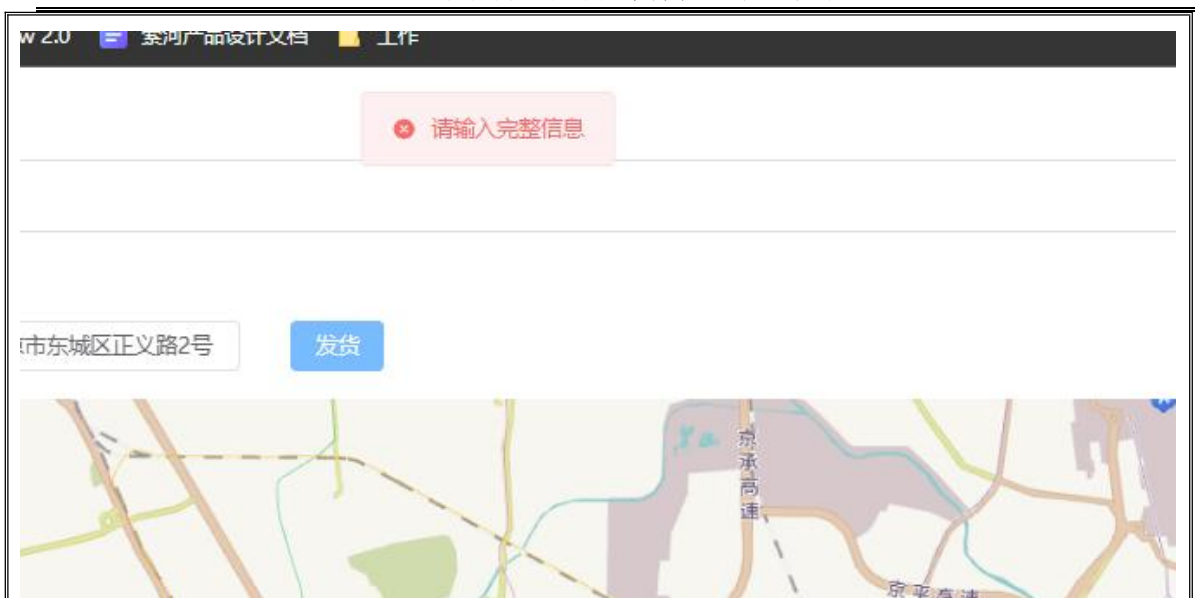


图 37 输入信息不完整测试图

测试结果及分析：同预期输出是否一致： 是（√）否（ ） 测试人：吴玉配

4. 物流管理及追踪

表 11

测试功能点：物流管理及追踪	
运行权限： 无	
测试初始条件：具有登录系统的账号	
<p>使用操作：</p> <ol style="list-style-type: none"> 1. 打开浏览器，输入网址 http://127.0.0.1:5173 2. 使用账号密码登录 3. 登入系统，点击我的物流，进行物流详信息的查看，物流拦截，物流更新位置，物流删除，查看物流运输路线等操作 	<p>预期输出：</p> <ol style="list-style-type: none"> 1. 可以对用户在此平台上发布的物流进行详细操作(查看，更新，修改，拦截)等

备注：
查看物流列表截图

物流 / 二维码 / 物流列表

系统设置

我的物流

物流列表

物流列表

id	物流名称	收货人	物流电话	物流流向	当前地址	时间	二维码	状态	操作
4	iphone	A	18338329327	河南财政金融学院 --> 北京市东城区正义路2号	正义路——北京城区	2023-03-20 17:20:38		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>
5	iphone			河南财政金融学院 --> 西安市未央区未央路	未央路段——西安市	2023-03-16 14:46:37		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>
6	iphone			河南财政金融学院 --> 拉萨市城关区城关区	峡口隧道——海东市	2023-04-23 21:05:56		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>
7	iphone			河南财政金融学院 --> 青岛市黄岛区黄岛区		1970-01-01 08:00:00		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>
8	iphone			河南财政金融学院 --> 福州市鼓楼区鼓楼区	峡口隧道——福州市	2023-03-16 14:46:37		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>
11	iphone			河南财政金融学院 --> 北京市东城区正义路2号		1970-01-01 08:00:00		运输中	<div><div>删除</div><div>拦截</div><div>更新位置</div><div>添加备注</div></div>

图 38 物流列表测试图

删除成功截图：

test拦截成功

物流流向	当前地址	时间	二维码	状态	操作
河南财政金融学院 --> 北京市东城区正义路2号	正义路——北京城区	2023-03-20 17:20:38		运输中	<div><div>删除</div><div>拦截</div></div>
河南财政金融学院 --> 西安市未央区未央路	未央路段——西安市	2023-03-16 14:46:37		运输中	<div><div>删除</div><div>拦截</div></div>
河南财政金融学院 --> 拉萨市城关区城关区	峡口隧道——海东市	2023-04-23 21:05:56		运输中	<div><div>删除</div><div>拦截</div></div>

图 39 删除物流测试图

更新物流位置截图：

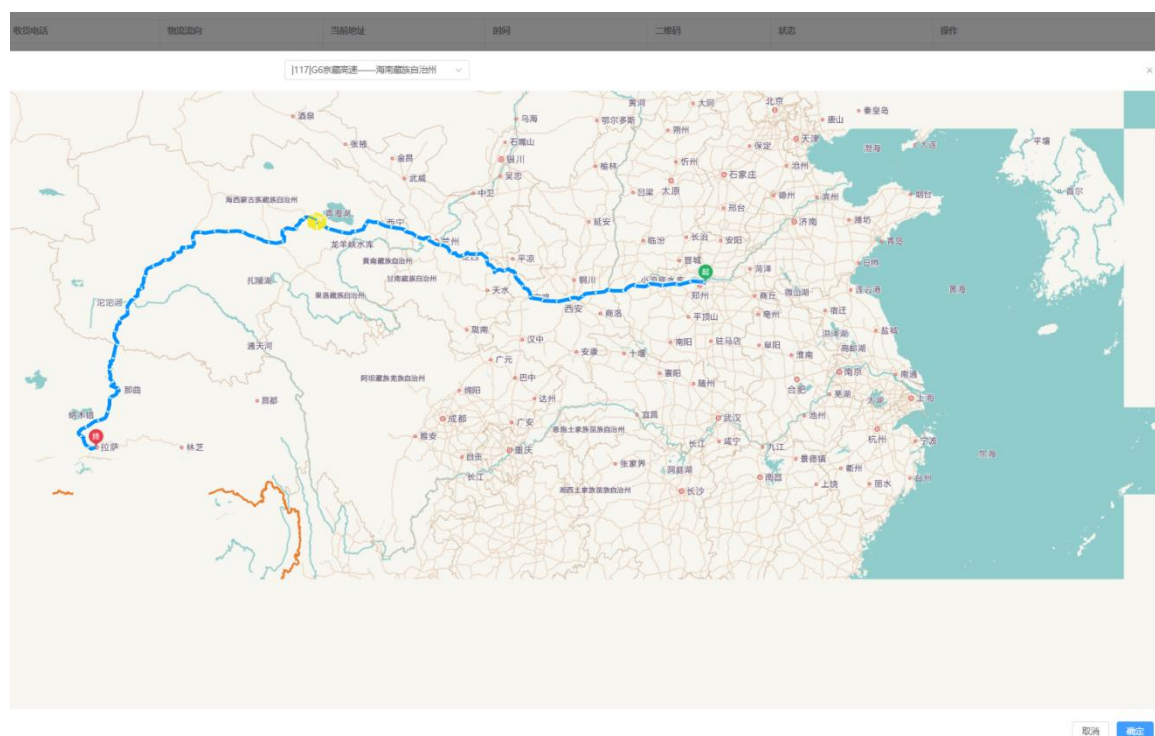


图 40 更新物流测试图

物流运输路线截图：



图 41 物流运输路线测试图

测试结果及分析：同预期输出是否一致： 是（√）否（ ） 测试人：吴玉配

5. 物流可视化追踪

表 12

测试功能点：物流可视化追踪

运行权限： 无	
测试初始条件： 具有登录系统的账号	
<p>使用操作：</p> <p>4. 打开浏览器，输入网址</p> <p>http://127.0.0.1:5173</p> <p>5. 使用账号密码登录</p> <p>6. 登入系统，点击物流流向</p>	<p>预期输出：</p> <p>1. 可以对用户在此平台上发布的物流进行可视化追踪</p> <p>2. 可以分别一不同的可视化效果，精准实时掌握物流流向，动态</p>
<p>备注：</p> <p>可视化追踪物流飞线图：</p> 	
<p>图 42 物流可视化(飞线图)测试图</p> <p>可视化追踪物流流向图：</p>	

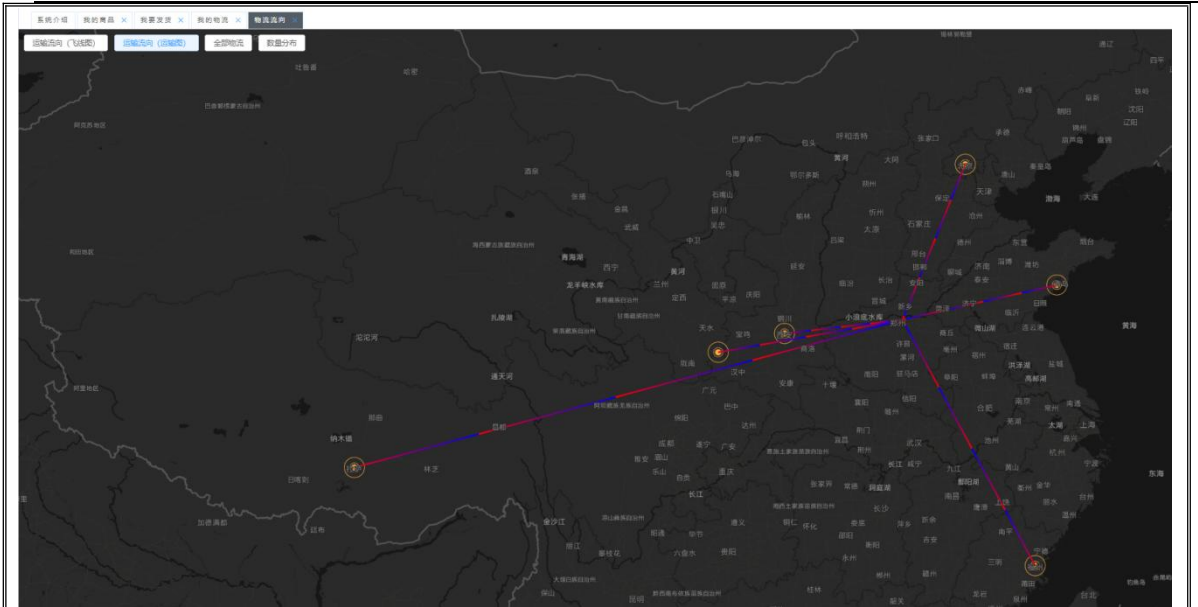


图 43 物流可视化(流向图)测试图

可视化追踪物流分布图:

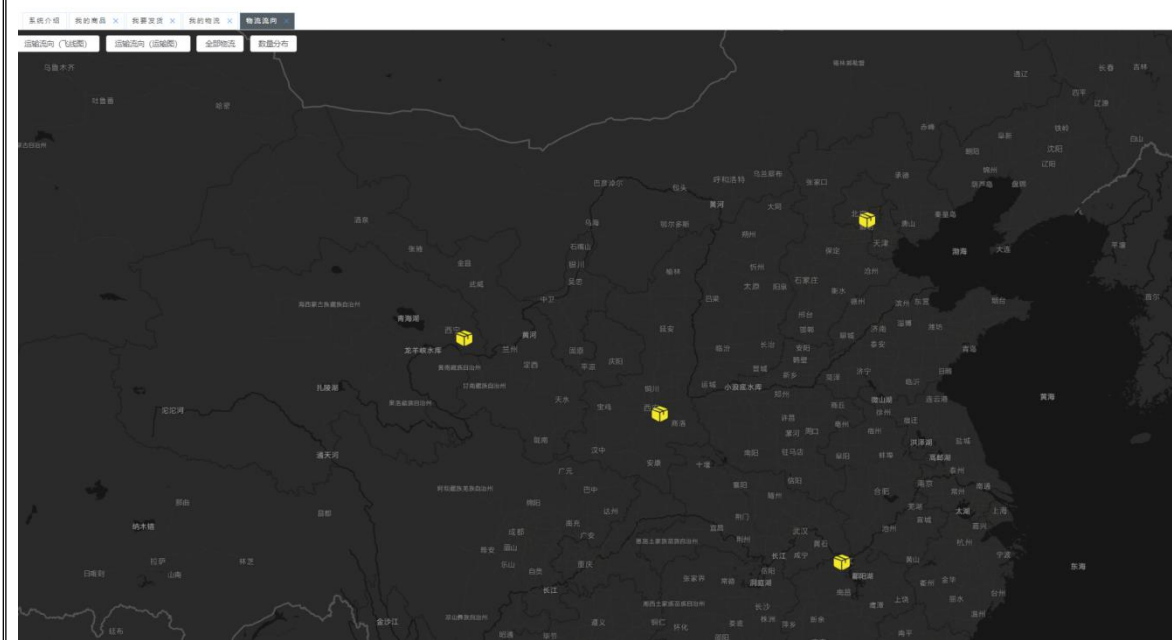


图 44 物流可视化(物流分布)测试图

运输过程中的物流信息:



6. 智能物流

表 13

测试功能点：智能物流	
运行权限： 无	
测试初始条件：具有登录系统的账号	
<p>使用操作：</p> <ol style="list-style-type: none">1. 打开浏览器，输入网址 http://127.0.0.1:51732. 使用账号密码登录3. 登入系统，点击智能物流	<p>预期输出：</p> <ol style="list-style-type: none">1. 可智能物流机器人发送物流编号2. 查询该物流编号的详细信息

备注：

发送物流编号截图：

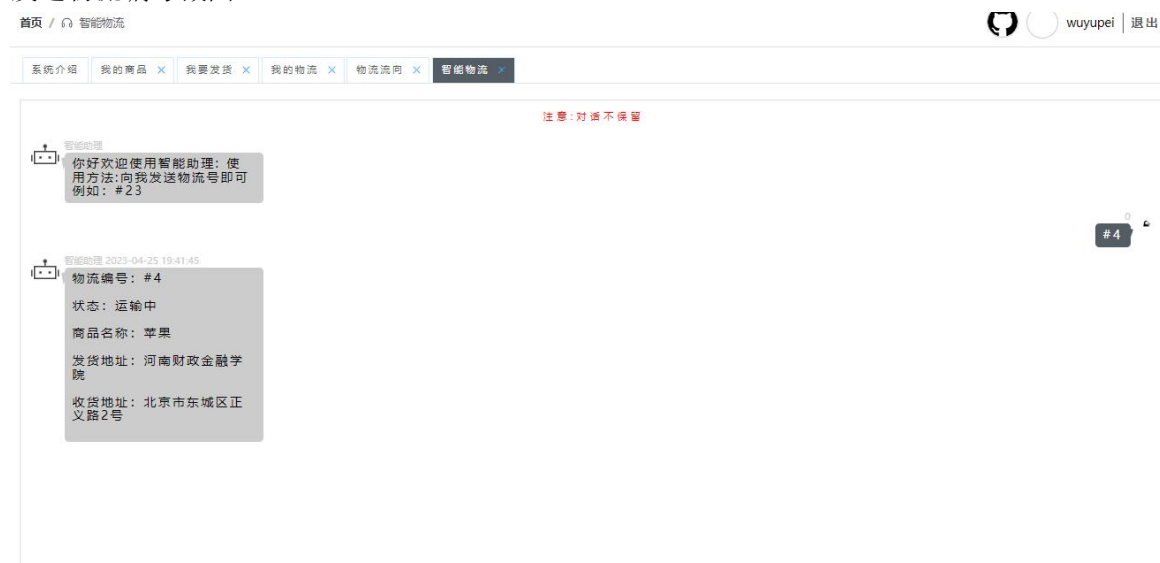


图 46 智能物流测试图

发送其他信息截图：



图 47 智能物流测试图

测试结果及分析：同预期输出是否一致： 是（√）否（ ） 测试人：吴玉配

6.4 测试结论

6.4.1 用户登录及修改个人信息

1. 用户登录功能正常，可以成功登录系统。
2. 修改个人信息功能正常，可以成功更新个人信息。

6.4.2 货物录入

1. 货物录入功能正常，可以成功添加货物信息。

2. 货物信息字段完整，录入信息符合要求。

6.4.3 物流管理

1. 物流管理功能正常，可以成功添加、编辑、删除物流信息。
2. 物流信息字段完整，录入信息符合要求。

6.4.4 发布物流

1. 发布物流功能正常，可以成功发布物流信息。
2. 物流信息显示清晰明了，用户可以轻松了解物流情况。

6.4.5 可视化追踪物流

1. 可视化追踪物流功能正常，可以实时显示货物位置和运输情况。
2. 地图显示清晰，可以准确反映货物位置和路线。

6.4.6 智能物流

1. 智能物流功能正常，可以发送不同 id 查询不同物流的详细信息

7 结论与展望

7.1 结论

通过对高德地图 GL API 物流追踪系统的设计与实现，我们实现了一个功能完善、性能优良的物流追踪系统。系统不仅支持用户登录、物流管理、物流追踪、可视化展示、智能物流以及发送邮箱提醒等多种功能，还通过采用 Node.js 和 Koa 框架、MySQL 和 Redis 数据库等技术栈，保证了系统的高性能和稳定性。同时，系统还采用了高德地图的 GL API，实现了物流地图展示和路径规划等功能。

7.2 展望

未来，随着物流行业的不断发展和创新，物流追踪系统也需要不断地更新迭代。因此，我们计划在未来的版本中加入更多的功能和技术。例如，我们可以采用人工智能技术来优化物流路径规划，提高运输效率和减少成本。同时，我们还可以引入更多的数据分析和可视化技术，以更好地展示物流数据，并为用户提供更好的体验和服务。此外，我们还将致力于提高系统的性能和稳定性，优化系统架构和数据库设计。我们相信，在未来的努力和创新下，我们的物流追踪系统将成为物流行业的重要工具之一，为用户带来更好的体验和服务。

参考文献

- [1]周文业. 基于移动 GIS 的共享物流系统的设计与实现[D].2019 年.05 期
- [2]李晶. 二维码技术应用在农产品物流追溯系统中的分析和思考[D].2014 年.19 期
- [3]朱健. 使用二维码要防信息泄露[D].2016 年.09 期
- [4]吴小力. 基于智慧供应链的电力物流跟踪系统[D].2019 年.48 期
- [5]Perera C, Zaslavsky A, Christen P, et al. Context Aware Computing for the Internet of Things: A Survey[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 414-454.
- [6]Gubbi J, Buyya R, Marusic S, et al. Internet of Things (Iot): A Vision, Architectural Elements, and Future Directions[J]. Future Generation Computer Systems, 2013, 29(7): 1645-1660.
- [7]Ngu A H H, Gutierrez M, Metsis V, et al. Iot Middleware: A Survey On Issues and Enabling Technologies[J]. IEEE Internet of Things Journal, 2017, 4(1): 1-20.
- [8]Zanella A, Bui N, Castellani A, et al. Internet of Things for Smart Cities[J]. IEEE Internet of Things Journal, 2014, 1(1).
- [9]谢欢, 陈继努. 基于 WebGIS 和 GPRS 的智能交通系统设计 与实现[J].2005(32)
- [10]孟志军, 赵春江, 王秀. 基于 GPS 的农田多源信息采集系统 的研究与开发 [J]2003(4).
- [11]钟聪儿, 邱荣祖. RFID 在茶叶物流追踪与追溯中的关键应用技术[J] 2016 年
- [12]柴凤伟. 货物安全的重要保障[J]2003 年.
- [13]胡江虹. 皮革企业智慧物流追踪与监管系统设计[J]2021 年.
- [14]闫锋. 浅谈物流追踪系统在国际 EPC 项目物流管理中的应用[J] 2018-04-28.
- [15]王道军. 基于北斗卫星的物流追踪系统研究[J] 2013-12-20.

致谢

在此我想向我的毕业设计导师、辅导员以及所有帮助过我的同学表达我的最深的感激之情。您的支持和鼓励是我完成这个项目的重要动力。

首先，我要感谢我的毕业设计导师，在我整个设计过程中提供了专业的指导和协助。在我遇到困难和疑惑的时候，导师总是耐心地解答我的问题，给我提供宝贵的建议和意见。他的认真负责和对我的指导和鼓励让我不断提高自己的能力，让我对未来充满信心。

同时，我也要感谢我的辅导员，她不仅在我毕业设计的过程中给予了我帮助，还在我的整个大学生活中给予我很多的关心和支持。她的热心帮助和耐心指导让我感受到大家庭的温暖，也让我在大学生活中学会了很多珍贵的经验和知识。

此外，我还要感谢我的同学们。在整个设计过程中，他们给予我很多帮助和支持。每当我遇到困难和疑惑的时候，他们总是毫不犹豫地伸出援手，帮助我克服困难。他们的鼓励和支持让我有信心和勇气克服困难，不断提升自己的能力和水平。

最后，我要再次感谢所有帮助过我的老师、同学和朋友们。没有你们的支持和鼓励，你们的支持和激励激发了我战胜困难的信心和勇气。在此，我向你们表达最深的谢意，也希望能够在未来的日子里继续与你们保持联系，共同进步，共同成长。

附录

后端部分代码:

启动代码:

```
import Koa from 'koa';

// database
import './config/database';

// middleware
import usePlugin from './middleware/usePlugin';
import useLogger from './middleware/logger';

// router
import useRouter from './router/index';

const app: Koa = new Koa();
app.proxy = true;
usePlugin(app);
useRouter(app);
useLogger(app);

app.listen(7552, () => {
  console.log('server is runing at 7552');
});
```

连接数据库代码:

```
import { Sequelize } from 'sequelize';
import { DB, USER, PASSWORD, HOST, DIALECT } from './index';

const sequelize = new Sequelize(DB, USER, PASSWORD, {
  host: HOST,
  dialect: DIALECT,
});

(async function () {
  try {
    await sequelize.authenticate();
    console.log('数据库链接成功....');
  } catch (err) {
    console.log('数据库链接失败....');
  }
})();
```



```

    } catch (error) {
      console.error('Unable to connect to the database:', error);
    }
  })();
}

```

```
export default sequelize;
```

登录代码:

```

loginRouter.post('/login', loginController.login);
loginRouter.post('/registe', loginController.register);

```

```
import loginService from './service/login.service';
```

```

class LoginController {
  async login(ctx: any, next: any) {
    const { account, password, type } = ctx.request.body;
    if (type == 1) {
      ctx.body = await loginService.login(account, password);
    } else {
      ctx.body = await loginService.adminLogin(account, password);
    }
  }

  async register(ctx: any, next: any) {
    // const { username, idcard, phonenumber, password, money } = ctx.request.body;
    // ctx.body = await loginService.register(username, idcard, phonenumber,
password, money);
  }

  async change(ctx: any) {
    const data = ctx.request.body;
    ctx.body = await loginService.change(ctx.userInfo.id, data);
  }
}

```

```
export default new LoginController();
```

```

import UserModel from './model/user.model';
import AdminModel from './model/admin';
import { genToken } from './utils/index';

```

```

async login(account: string, password: string) {
  const res: any = await UserModel.findOne({
    where: {
      account,
      password,
    },
  });

  if (res) {
    const token = genToken({ account, password, id: res.id });
    return Object.assign({}, res.dataValues, { msg: '登录成功', code: 200, token,
password: " " });
  }

  return { code: 210, msg: '登录失败' };
}

async adminLogin(account: string, password: string) {
  const res: any = await AdminModel.findOne({
    where: {
      account,
      password,
    },
  });

  if (res) {
    const token = genToken({ account, password, id: res.id });
    return Object.assign({}, res.dataValues, { msg: '登录成功', code: 200, token,
password: " " });
  }

  return { code: 210, msg: '登录失败' };
}

async register(username: any, idcard: any, ppassword: any, money: any) {
  const res = await UserModel.create({
    uname: username,
    money,
    idcard,
  });
  console.log(res);
}

async change(id: any, data: any) {

```

```

    }
  }

  async change(id: any, data: any) {
    let res = await UserModel.update(
      { ...data },
      {
        where: {
          id,
        },
      }
    );

    if (res.length) {
      return { msg: '修改成功', code: 200 };
    } else {
      return { msg: '修改失败', code: 400 };
    }
  }
}
}

```

```
export default new LoginService();
```

其他接口代码:

```

import KoaRouter from 'koa-router';
import shopController from '../controller/shop.controller';
import veryToken from '../middleware/veryToken';

const shopRouter: KoaRouter = new KoaRouter();

shopRouter.get('/shop', veryToken, shopController.findById);

shopRouter.post('/createshop', veryToken, shopController.createShop);

shopRouter.get('/delete', veryToken, shopController.delete);

shopRouter.post('/updateAddress', veryToken, shopController.updateAddress);

shopRouter.post('/getInfoById', veryToken, shopController.getInfoById);

shopRouter.post('/addATransportInfo', veryToken,
shopController.addATransportInfo);

```



```
shopRouter.post('/createThings', veryToken, shopController.createThings);

shopRouter.get('/getAllThings', veryToken, shopController.getAllThings);

shopRouter.delete('/deleteOneThing', veryToken, shopController.deleteOneThing);

shopRouter.post('/changeOneThing', veryToken, shopController.changeOneThing);

export default shopRouter;
```

前端代码:

导航栏代码:

```
<template>
  <div class="aside-body">
    <el-menu :router="true" :default-active="defaultActive"
class="el-menu-vertical-demo">
      <template v-for="item in menu" :key="item.index">
        <template v-if="item.children">
          <el-sub-menu :index="item.index">
            <template #title>
              <el-icon><location /></el-icon>
              <span>{{ item.title }}</span>
            </template>
            <el-menu-item v-for="sItem in item.children" :index="sItem.index">
              <el-icon>
                <component :is="sItem.icon"></component>
              </el-icon>
              <span>{{ sItem.title }}</span>
            </el-menu-item>
          </el-sub-menu>
        </template>
        <template v-else>
          <el-menu-item :index="item.index">
            <el-icon>
              <component :is="item.icon"></component>
            </el-icon>
            <span>{{ item.title }}</span>
          </el-menu-item>
        </template>
      </template>
    </div>
  </template>
```

```

</template>

<script setup>
import { ref } from 'vue';

const menu = [
  { title: '系统介绍', index: '/', icon: 'House' },
  { title: '我的货物', index: '/MyShops', icon: 'ShoppingBag' },
  {
    title: '物流管理',
    index: '/WoYaoFahuo',
    icon: 'House',
    children: [
      {
        title: '我要发货',
        index: '/WoYaoFahuo',
        icon: 'Coordinate',
      },
      {
        title: '我的物流',
        index: '/WoDeFahuo',
        icon: 'Van',
      },
    ],
  },
  { title: '物流流向', index: '/WuLui', icon: 'Refresh' },
  { title: '智能物流', index: '/ZhiNengWuLui', icon: 'Headset' },
  { title: '个人中心', index: '/Self', icon: 'User' },
];

let defaultActive = ref(menu[0].index);
const route = useRoute();

function findIndexByPath(menu, path) {
  const map = menu.reduce((pur, cur, index) => {
    if (cur.children) {
      pur.push(...cur.children);
    } else {
      pur.push(cur);
    }
  }, []);
  return map.find((item) => item.index === path)?.index || menu[0].index;
}

```

```

route,
() => {
  const path = route.fullPath;
  defaultActive.value = findIndexByPath(menu, path);
},
{
  immediate: true,
}
);
</script>

```

```

<style scoped lang="less">
.aside-body {
  height: calc(100% - 60px);
}

.el-menu {
  width: 100%;
  height: 100%;
  .el-menu-item {
    &.is-active {
      color: #fff;
      background-color: #545c64;
    }
    &:hover {
      color: #fff;
      background-color: #545c64;
    }
  }
}
</style>

```

可视化地图代码

```

<template>
  <div class="content">
    <Map ref="mapRef"></Map>
    <div class="btns">
      <el-button @click="showShopFlow">运输流向（飞线图）</el-button>
      <el-button @click="showShopTransport">运输流向（运输图）</el-button>
      <el-button @click="showAllShop">全部物流</el-button>
      <el-button @click="showCount">数量分布</el-button>
    </div>
  </div>
</template>

```

```

    <div class="styles">
      <el-radio-group v-model="currentStyle" @change="radioChange">
        <el-radio v-for="item in styles" :label="item.name"
size="large">{{ item.label }}</el-radio>
      </el-radio-group>
    </div>
  </div>
</template>

```

```

<route lang="yaml">
meta:
  name: 物流流向
</route>

```

```

<script setup>
import useAppStore from '../store/app';

import { findShop } from '../service/home';
const mapRef = ref(null);

let appStore = useAppStore();
let currentStyle = ref('normal');
let styles = [
  { label: '标准', name: 'normal' },
  { label: '幻影黑', name: 'dark' },
  { label: '月光银', name: 'light' },
  { label: '远山黛', name: 'whitesmoke' },
  { label: '草色青', name: 'fresh' },
  { label: '雅士灰', name: 'grey' },
  { label: '涂鸦', name: 'graffiti' },
  { label: '马卡龙', name: 'macaron' },
  { label: '靛青蓝', name: 'blue' },
  { label: '极夜蓝', name: 'darkblue' },
  { label: '酱籽', name: 'wine' },
];
function radioChange(style) {
  console.log(`amap://styles/${style}`);
  mapRef.value.getMap().setMapStyle(`amap://styles/${style}`);
}

let shops = reactive({});

findShop(appStore.userInfo.id).then((res) => {
  shops = res.data;

```