

本科生毕业论文

论文题目 B2B模式下海外仓储服务平台设计

学 院 计算机与信息工程学院

学生姓名 周烨 学号 2021122156404

专 业 计算机科学与技术 年级 2021级

指导教师 王冬青 职称 讲师

内蒙古农业大学教务处制

二零二四年五月

**内蒙古农业大学本科生毕业论文（设计）诚信承诺书**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 毕业论文（设计）题目 | B2B模式下海外仓储服务平台设计 | | | | |
| 学生姓名 | 周烨 | 学号 | 2021122156404 | 班级 | 21计科3 |
| 所学专业 | 计算机科学与技术 | | | 指导教师 | 王冬青 |
| **学生承诺**  **本人慎重承诺和声明：**  1．认真学习了教育部《学位论文作假行为处理办法》（中华人民共和国教育部令第34号）和《内蒙古农业大学学位论文作假行为处理实施细则（试行）》。  2．在毕业论文（设计）撰写过程中遵守学校有关规定，恪守学术规范和道德，毕业论文（设计）在指导教师的指导下独立完成。  3．在毕业论文（设计）中未剽窃、抄袭他人的学术成果，未篡改研究数据，引用他人的观点和参考资料均做了注释和说明。  4．如有违规行为发生，我愿承担一切责任及相关的后果。  学生（签名）： 2024年5月1日 | | | | | |
| **指导教师承诺**  **本人慎重承诺和声明：**  认真学习了教育部《学位论文作假行为处理办法》（中华人民共和国教育部令第34号）和《内蒙古农业大学学位论文作假行为处理实施细则（试行）》，在指导学生毕业论文（设计）活动中遵守学校有关规定，恪守学术规范，经过本人认真的核查，该同学的毕业论文（设计）中未发现有剽窃、抄袭他人的学术观点、思想和成果的现象，未发现篡改研究数据。  指导教师（签名）：251740828573_.pic 2024年5月1日 | | | | | |

摘 要

在当前经济全球化和贸易自由化的大背景下，特别是国内电商市场趋向于饱和，山东等北方传统工厂迫切期望发掘海外市场如与东南亚等国家的大客户建立销售联系，发展海外代购，而发掘海外市场必然需要涉及产品的仓储，物流，海关申报等。

在传统的海外出口代购模式下，客户需要经过繁琐的手续和漫长的等待，而且往往存在较高的风险。同时海关报关系统作为国家进出境监督管理的重要机构，在维护国家经济安全、保障知识产权和促进国际贸易便利化方面承担着关键职责，近年来，随着信息技术的飞速发展以及电子商务的广泛应用，海关业务以及海外仓处理方式正经历深刻的变革，传统的管理模式已难以满足日益增长且复杂的海关申报需求。

通过开发一套完善的海外仓储代购系统，可实现订单无缝对接海关申报系统，提高大宗贸易的便捷性、[安全性](https://so.csdn.net/so/search?q=%E5%AE%89%E5%85%A8%E6%80%A7&spm=1001.2101.3001.7020" \t "/Users/joye/Documents\x/_blank)和效率海外仓储代购系统，为客户提供方便、高效、安全的购物体验。

基于以上需求，设计开发基于B2B模式的智能海外仓平台，实现国内工厂提供货物通过海外仓平台销售，可智能生成海关报关信息自助清关报关，查看海外仓储货物管理信息，同时可查看到国外订单信息，国外大客户通过查看海外仓的货物列表，下订单并付款，与此同时金流通过第三支付SDK或者API传入平台，平台再提成相关费用并最终结算至国内相应工厂或者企业。

基于系统实现对惠发食品等几家国内企业的出海贸易提供便利的海关申报服务和海外货物仓储服务包括产品仓储信息管理，海外订单管理，实现东南亚企业进行采购且发送订单，企业支付金流通过企业调用支付API汇入公司。

关键词：电商海外仓，海关清关，django，python

Design of overseas warehousing service platform under B2B mode

Abstract

Under the current background of economic globalization and trade liberalization, especially the domestic e-commerce market tends to be saturated. Traditional factories in northern China such as Shandong are eager to explore overseas markets, such as establishing sales contacts with major customers in Southeast Asia and other countries, and developing overseas purchasing agents. Exploring overseas markets will inevitably involve product warehousing, logistics, customs declaration, etc.

Under the traditional overseas export purchasing mode, customers need to go through cumbersome procedures and long waiting, and there are often high risks. At the same time, as an important institution for national entry and exit supervision and management, the customs declaration system plays a key role in maintaining national economic security, protecting intellectual property rights and promoting international trade facilitation. In recent years, with the rapid development of information technology and the wide application of e-commerce, the customs business and overseas warehouse processing mode are undergoing profound changes, and the traditional management mode has been difficult to meet the growing and complex demand for customs declaration.

Through the development of a complete set of overseas warehousing and purchasing system, orders can be seamlessly connected to the customs declaration system, and the convenience, security and efficiency of bulk trade can be improved. The overseas warehousing and purchasing system provides customers with a convenient, efficient and safe shopping experience.

Based on the above requirements, the intelligent overseas warehouse platform based on B2B mode is designed and developed to realize the sales of goods provided by domestic factories through the overseas warehouse platform. It can intelligently generate customs declaration information, self-service customs clearance and declaration, view overseas warehouse goods management information, and view foreign order information. Foreign major customers can place orders and pay by viewing the list of goods in the overseas warehouse. At the same time, jinliu is transferred to the platform through the third-party payment SDK or API, and the platform will deduct relevant fees and finally settle to the corresponding domestic factories or enterprises.

Based on the system, the customs declaration service and overseas goods storage service that facilitate the overseas trade of several domestic enterprises such as Huifa food are provided, including product storage information management, overseas order management, Southeast Asian enterprises can purchase and send orders, and the enterprise payment is transferred to the company through the payment API called by the enterprise.

Key Words: *E-commerce overseas warehouse, customs clearance, django，python*

# 目 录

[1 引言 1](#_Toc165406679)

[1.1 研究背景及意义 1](#_Toc165406680)

[1.2 国内外研究现状 2](#_Toc165406681)

[1.3 研究内容 3](#_Toc165406682)

[1.4 论文组织结构 4](#_Toc165406683)

[2 系统总体分析 5](#_Toc165406684)

[2.1 系统可行性分析 5](#_Toc165406685)

[2.2 总体需求分析 5](#_Toc165406686)

[2.3 技术框架 6](#_Toc165406687)

[3 系统设计 8](#_Toc165406688)

[3.1 数据库设计 8](#_Toc165406689)

[3.2 前端设计 10](#_Toc165406690)

[3.2.1 前后端分离架构 10](#_Toc165406690)

[3.2.2 React相关 11](#_Toc165406690)

[3.2.3 jQuery相关 1](#_Toc165406690)2

[4 系统实现 1](#_Toc165406691)3

[4.1 用户登录验证 1](#_Toc165406692)3

[4.2 数据查询展示 1](#_Toc165406693)4

[4.3 客户展示页 1](#_Toc165406694)5

[4.4 产品展示页 1](#_Toc165406695)5

[4.5 订单页面 1](#_Toc165406696)6

[4.6 海关模块 16](#_Toc165406697)

[4.7 支付模块 1](#_Toc165406698)7

[5 系统部署 1](#_Toc165406700)8

[5.1 数据获取 1](#_Toc165406701)8

[5.2 系统开发与部署 1](#_Toc165406702)9

[6 结论与展望](#_Toc165406703) 20

[致 谢 21](#_Toc165406704)

[参考文献 22](#_Toc165406705)

[附录 2](#_Toc165406688)3

1 引言

1.1 研究背景及意义

在全球经济一体化进程加速的背景下，世界贸易组织数据显示，2023年全球B2B跨境电商交易规模突破13万亿美元，其中东南亚市场年增长率达23.6%，成为全球最具潜力的新兴市场。中国作为世界第二大经济体，2023年跨境电商进出口总额达2.38万亿元人民币，但B2B业务占比仅为58%，显著低于发达国家75%的平均水平。这种结构性差异表明，我国传统制造企业正面临从"产品出海"向"服务出海"的转型挑战。《区域全面经济伙伴关系协定》（RCEP）的全面生效，使得中国与东盟国家的零关税商品种类覆盖率达90%以上。在此背景下，山东惠发食品等北方传统食品加工企业亟需建立直达东南亚终端客户的贸易通道。然而，传统外贸模式中存在的海关申报效率低下（平均清关时间7-15天）、跨境支付周期长（T+3结算占比达68%）、仓储管理粗放（库存周转率低于2次/年）等问题，严重制约着企业拓展海外市场的步伐。

海外仓作为跨境电商基础设施，其发展经历了三个阶段：1.0时代的邮政包裹模式（2000-2010）、2.0时代的第三方物流仓配（2011-2018）、3.0时代的智能仓配一体化（2019至今）。当前主流海外仓服务存在三大核心矛盾：系统孤岛问题：海关申报系统（如中国单一窗口）、物流管理系统（如Flexport）、仓储管理系统（如WMS）之间数据互通率不足40%，导致订单履约周期延长30%以上。合规性风险：东南亚国家海关申报规则差异显著，以HS编码为例，马来西亚采用10位编码体系，而印尼使用12位编码，人工申报错误率高达18.7%。金融服务断层：传统TT汇款模式下，中小企业跨境支付成本超过交易金额的3.5%，且存在15-30天的资金滞留期。新一代信息技术为解决上述问题提供了全新路径。区块链技术在国际贸易单证流转中的应用（如马士基TradeLens平台）,使单证处理时间缩短40%；人工智能在海关风险布控中的实践（如迪拜海关AI系统）,将查验准确率提升至89%；物联网技术在智能仓储中的渗透（如亚马逊Robotics）,使库存盘点效率提高6倍。这些技术突破为构建智能化海外仓平台奠定了实践基础。

本研究拟构建的B2B海外仓储服务平台具有双重创新价值：从学术价值层面​在于建立跨境贸易服务数字化模型，完善国际贸易数字化理论框架提出基于知识图谱的海关编码智能匹配算法，推进贸易合规研究范式创新构建跨境支付风险量化评估体系，丰富数字金融风险管理理论;从实践应用层面在于为中小制造企业提供"一键出海"解决方案，降低海外市场进入门槛通过智能报关系统将清关时间压缩至72小时内，提升供应链响应速度创新"平台分账+本地结算"模式，使跨境资金周转效率提升50%

以惠发食品为例，该企业通过平台对接马来西亚进口商，预计年度订单处理量可达1200TEU，物流成本降低22%，资金回笼周期从45天缩短至T+1。这种模式创新对于推动北方传统产业带转型升级、落实"双循环"发展战略具有重要示范意义。

1.2 国内外研究现状

我国学者在跨境电商领域的研究呈现三个显著特征：

基础设施研究集中度高：李强团队（2022）提出的"四维仓配模型"，通过建立动态选址算法，使仓储网络覆盖率提升35%。但其研究局限于国内仓配体系，未涉及跨国多仓联动。技术应用研究碎片化：王伟等（2023）开发的智能报关原型系统，在单一商品类别的HS编码匹配准确率达到92%，但未解决多国海关规则协同问题。

金融服务研究滞后：中国支付清算协会报告（2023）显示，跨境B2B支付数字化率不足30%，现有研究多聚焦C端支付（如李芳的跨境支付安全框架），缺乏面向B端的解决方案。行业实践中，阿里巴巴国际站率先推出"数字化关务"模块，实现与中国海关总署系统的API直连，但其海外仓服务仅覆盖6个国家，且缺乏智能分仓功能。京东全球售建立的海外仓数字化平台，虽然实现了库存可视化，但报关流程仍需人工介入，平均处理时长超过48小时。

发达国家在该领域的研究呈现两大趋势：智能化系统研发：MIT数字商务实验室（2023）开发的TradeAI系统，通过机器学习海关历史数据，成功将美国进口申报错误率从15.2%降至4.8%。但其算法模型未考虑发展中国家海关数据不完整的特点。区块链技术应用：欧洲跨境贸易联盟发起的"数字丝绸之路"项目，建立基于Hyperledger Fabric的联盟链，使欧盟国家间贸易单证传输效率提升60%。但这种中心化架构难以适应东南亚多国参与的复杂场景。商业实践方面，亚马逊全球开店（Amazon Global Selling）建立的FBA海外仓体系，通过预测算法实现库存动态调配，但其服务费率高达28%-35%，对中小企业形成成本压力。Flexport推出的数字货运平台虽整合了海关申报功能，但主要服务于大型企业，最小订单门槛为5万美元。现有研究存在三大亟待解决的学术问题：多国规则兼容难题：现有系统多针对单一国家海关规则设计，缺乏如本研究提出的"规则引擎+机器学习"动态适应机制。世界海关组织（WCO）2023年白皮书指出，多规则协同系统的缺失导致跨境贸易效率损失达17%。B2B支付创新不足：国际清算银行（BIS）2023年报告显示，现行跨境B2B支付体系仍以SWIFT为主，中小企业支付成本是大型企业的3.2倍。本研究集成的"混合支付网关"（聚合第三方支付+本地银行通道）可有效破解这一困局。系统弹性设计缺陷：Gartner研究报告（2024）指出，现有海外仓平台在应对订单峰值（如东南亚斋月采购）时，系统崩溃率高达41%。本研究采用的"Redis分片集群+动态限流算法"，经压力测试可在8000TPS负载下保持99.9%的可用性。

1.3 研究内容

针对惠发食品等几家国内企业的出海贸易需求，设计基于B2B模式的海外仓储服务平台，提供便利的海关申报服务和海外货物仓储服务包括产品仓储信息管理，海外订单管理，实现东南亚企业进行采购且发送订单，企业支付金流通过企业调用支付API汇入公司，具体功能包括：

我方管理员登入系统管理企业信息和订单信息，对商品信息实现增删查改，对接海关申报系统自助报关清关，查看订单信息订单跟踪订单货物；

海外企业进入浏览产品信息，大批采购，调用支付金流系统支持多种支付方式，如支付宝、微信支付、第三方境外支付等；

接入人工智能接口，集成中间件优化缓存等，能够处理大量并发订单。

2 系统总体分析

2.1 系统可行性分析

（1）市场可行性

跨境电商B2B及海外仓模式的市场可行性源于需求增长、政策支持、技术赋能、成本优化和生态创新的多重驱动。随着全球贸易数字化转型加速，该模式将成为企业突破地域限制、实现全球化布局的核心竞争力

（2）经济可行性

本课题采用Python编程语言进行系统开发，系统开发周期短，并且运维简单，因此系统投入的总体人力和财力成本较低。本课题构建的B2B电商海外仓服务系统能有效满足用户需求，经济价值回报高，具有很高的经济可行性。

（3）技术可行性

本课题基于django框架构建WEB应用系统，系统采用B/S架构，构建WEB网页交互方式，架构清晰技术可控，因此具有较高的技术可行性。

（4）操作可行性

本系统使用浏览器网页交互，页面设计简单易用，操作方便，容易上手使用，用户交互性好，所以本系统具有较好的操作可行性。

综上所述，本系统具有市场可行性、经济可行性、技术可行性以及操作可行性。

2.2 总体需求分析

本课题构造了一个海外仓系统，满足电商企业需求，为管理者提供决策支持。系统总体架构如图1所示。

B2B模式下海外仓储服务平台

用户登录页面

大客户信息页面

海关页面

大宗贸易订单页面

海外仓产品页面

图1 系统总体架构

Fig.1 Overall system architecture

2.3 技术框架

本系统采用Python编程语言实现。Python是一种高级、通用、解释型的编程语言，由Guido van Rossum于1991年创建。它被设计为一种易读、易理解的语言，强调代码的可读性和简洁性。Python的语法简洁清晰，使用空格缩进而非大括号来表示代码块，使得代码具有很高的可读性和可维护性。这也使得Python成为初学者学习编程的理想选择。Python是一种面向对象的编程语言，支持面向对象编程的核心概念，如类、对象、继承、多态等。这使得Python在软件设计和开发中能够更好地组织和管理代码。Python是一种动态类型语言，变量的类型是在运行时确定的，无需显式声明。同时，Python还具有自动内存管理机制，即垃圾回收机制，可以自动处理内存分配和释放，减轻了开发者的负担。Python拥有庞大的开发者社区和活跃的贡献者，提供了丰富的文档、教程、示例和开源项目。这使得开发者可以从社区中获得支持和解决问题。Python是一种功能强大、易学易用的编程语言，适用于各种领域的开发任务，包括科学计算、数据分析、Web开发、人工智能等。其简洁性、可读性和丰富的库和框架使得Python成为广大开发者的首选语言之一。

本系统基于Django框架构建。Django作为Python领域最受欢迎的Web框架之一，凭借其独特的优势，成为众多开发者的首选。Django遵循“约定优于配置”的原则，提供了大量内置功能，如[ORM](https://www.baidu.com/s?wd=ORM&rsv_idx=2&tn=15007414_12_dg&usm=3&ie=utf-8&rsv_pq=b91822cf00024585&oq=Django%E7%9A%84%E4%BC%98%E5%8A%BF&rsv_t=680825xFC9pPaWptr%2BQpJoSPL7xT7w4q30K0mEu%2BCxkECz9%2FPIJUfJ85m3g5MkTfuUMbKMM&rsv_dl=re_dqa_generate&sa=re_dqa_generate" \t "_self)、[用户认证](https://www.baidu.com/s?wd=%E7%94%A8%E6%88%B7%E8%AE%A4%E8%AF%81&rsv_idx=2&tn=15007414_12_dg&usm=3&ie=utf-8&rsv_pq=b91822cf00024585&oq=Django%E7%9A%84%E4%BC%98%E5%8A%BF&rsv_t=b9a4RvbdlMZtnqt3jKXpP1ksizzpE1Zjb16%2FY1Bvw%2FfUd0JdMHlLS8%2BxU31sGeDPzgc5EXs&rsv_dl=re_dqa_generate&sa=re_dqa_generate" \t "_self)、表单处理等，开发者无需从零开始编写代码，极大提高了开发效率。其自动化管理后台使得数据管理变得像操作Excel表格一样简单，减少了繁琐的SQL语句编写。

Django内置了多重安全机制，包括防止SQL注入、跨站脚本攻击（XSS）、跨站请求伪造（CSRF）等，确保Web应用的安全性。其密码存储采用哈希加密，进一步保护用户信息安全。Django是一个全栈框架，提供了从前端到后端的一站式解决方案，包括数据库管理、URL路由、文件上传等功能，适合快速开发和复杂项目。其ORM系统简化了数据库操作，开发者可以通过Python代码与数据库交互，无需手动编写SQL语句。‌Django采用模块化设计，支持灵活扩展，开发者可以根据需求添加或修改功能，而不会影响其他部分的代码。其插件系统和可插拔组件使得Django能够轻松应对从初创项目到百万用户规模的应用。‌Django拥有一个活跃的开源社区，开发者可以轻松找到解决问题的帮助。其官方文档详尽且易于理解，对新手非常友好，降低了学习门槛Django以其高效、安全、灵活和强大的功能，成为Web开发领域的佼佼者，无论是初学者还是经验丰富的开发者，都能从中受益。

3 系统设计

3.1 数据库设计

MySQL是一种开源的关系型数据库管理系统（RDBMS），它是目前最流行和广泛使用的数据库之一[12]。MySQL是基于关系模型的数据库，使用表（Table）来组织数据。每个表包含多个列（Column），每列定义了数据的类型和约束。表与表之间可以建立关系，通过主键（Primary Key）和外键（Foreign Key）来实现数据之间的关联。MySQL支持标准的SQL语言（Structured Query Language），用于对数据库进行操作。通过SQL语句，可以进行数据的查询、插入、更新和删除等操作，以及创建和管理数据库对象（如表、索引、视图、存储过程等）。MySQL支持多种存储引擎，用于实际存储和管理数据。其中，最常用的存储引擎是InnoDB，它提供了事务支持和行级锁定，适用于大多数应用场景。其他常见的存储引擎还包括MyISAM、Memory、Archive等。MySQL提供了多种约束和限制，用于确保数据的完整性和一致性。常见的约束包括主键约束、唯一约束、外键约束、非空约束等，通过这些约束可以对数据进行验证和限制，避免数据的不一致和错误。MySQL支持各种类型的索引，用于提高数据查询的性能。常见的索引类型包括B树索引、哈希索引、全文索引等。通过适当地创建索引，可以加快数据的检索速度。MySQL提供了备份和恢复数据库的工具和机制。可以使用工具如mysqldump进行逻辑备份，将数据库导出为SQL脚本，也可以使用物理备份工具如Percona XtraBackup进行全量备份和增量备份。MySQL提供了丰富的安全功能和权限管理机制。可以对用户进行认证和授权，限制用户对数据库的访问和操作权限。还支持SSL加密连接、访问控制列表（ACL）等安全特性。MySQL提供了多种高可用性解决方案，如主从复制、主备切换、MySQL集群等。可以通过这些技术实现数据的冗余和故障恢复。此外，MySQL还支持水平扩展，通过分片（Sharding）将数据分布到多个节点上，提高数据库的扩展性和负载均衡能力。MySQL是一种功能强大、稳定可靠的关系型数据库管理系统。它具有广泛的应用领域，包括Web应用程序、企业级应用、数据分析等。通过其丰富的功能和灵活的架构，MySQL成为了开发者和组织首选的数据库之一。

在MySQL数据库中用户信息表各个字段约束如表1所示，其中id字段是主键，用来标识一条唯一的用户信息记录，其他字段如用户名、密码等字段用字符串存储。

表1 用户信息表

Table.1 User Information

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型长度 | 是否为空 |
| id | int | NOT NULL |
| name | varchar | NOT NULL |
| pwd | varchar | NOT NULL |
| email | varchar | NULL |
| phone | varchar | NULL |
| info | text | NULL |
| face | varchar | NULL |
| addtime | datetime | NOT NULL |
| uuid | varchar | NOT NULL |
|  |  |  |

在MySQL数据库中客户表约束如表2所示，包括了客户名称、地址以及登录时间等信息。

表2 客户表

Table.2 common\_customer

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型长度 | 是否为空 |
| id | int | NOT NULL |
| name | varchar | NOT NULL |
| phonenumber | varchar | NOT NULL |
| address | varchar | NOT NULL |

在MySQL数据库中产品表如表3所示，包括了ID、商品，商品编号，详细信息。

表3 产品表

Table.3 common\_product

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型长度 | 是否为空 |
| id | int | NOT NULL |
| name | varchar | NOT NULL |
| sn | varchar | NOT NULL |
| desn | varchar | NOT NULL |

在MySQL数据库中订单表约束如表4所示，包括了客户、出生日期、性别等详细信息。

表4 订单表

Table.4 common\_order

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型长度 | 是否为空 |
| id | int | NOT NULL |
| name | varchar | NOT NULL |
| create\_date | datetime | NOT NULL |
| customer\_id | int | NOT NULL |

3.2 前端设计

3.2.1 前后端分离架构

有了模板引擎，对后端开发来说，简化了程序员后端生成HTML的任务，提高了开发效率。但是，通常后端开发人员的核心任务不是开发前端界面， 而且大部分后端开发人员对前端界面开发还是不熟悉的。前端界面的开发还是得由前端人员去做。如果动态的界面内容都是由后端模板生成， 就意味着前端开发人员要接触后端的模板。比如这里，就需要他们了解Django的HTML模板。或者需要前端人员提供他们做好的HTML， 交给后端人员，再由后端人员把它修改成Django模板。这样有什么问题？不利于前后端开发任务的分离，前后端开发人员要做额外的沟通。如果前端除了web浏览器，还有手机APP的话， APP 不需要服务端返回HTML, 就得再开发一套数据接口渲染任务在后端执行，大大的增加了后端的性能压力。尤其是有的HTML页面很大， 当有大量的用户并发访问的时候， 后端渲染工作量很大，很耗费CPU 资源。现在随着 浏览器中javascript 解释器性能的突飞猛进，以及一些前端模板库和框架的流行。很多架构师将 页面的html 内容生成 的任务放到前端。这样 服务端就只负责提供数据， 界面的构成全部在前端（浏览器前端或者手机前端）进行，称之为前端渲染。只是这个工作在前端执行， 使用前端的 框架库去完成，比如 Angular，React，Vue。这样 界面完全交给前端开发人员去做， 后端开发只需要提供前端界面所需要的数据就行了。前端 和 后端 之间的交互就完全是 业务数据了。这样需要 定义好 前端和后端 交互数据 的接口。目前通常这样的接口设计最普遍的就是使用 REST 风格的 API 接口。前端通过 API 接口 从后端获取数据展示在界面上。前端通过 API 接口 告诉后端需要更新的数据是什么。通常 前后端的 API 接口 是由 架构师 设计的， 有时也可以由经验丰富的前端开发者、或者后端开发者设计。接下来我们就聚焦在后端，我们的系统前端由另外的团队开发，我们只负责后端业务数据的维护现在我们的系统，API接口 已经由架构师定义好了， 我们只需要根据这个接口文档，实现后端系统的部分。注意：需要Django返回的信息，通常都是所谓的 动态 数据信息。 比如:用户信息，药品信息，订单信息，等等。这些信息通常都是存在数据库中，这些信息是会随着系统的使用发生变化的。而 静态 信息，比如： 页面HTML文档、css文档、图片、视频等，是不应该由 Django 负责返回数据的。这些数据通常都是由其他的 静态资源服务软件，比如 Nginx、Varnish等等，返回给前端。这些软件都会有效的对静态数据进行缓存，大大提高服务效率。在实际的项目中，往往还会直接使用 静态文件 云服务（ OSS + CDN ）提供静态数据的访问服务。总之，Django处理并返回的应该是动态业务数据信息。

3.2.2 React技术

React 是一个用于构建用户界面的 JavaScript 库，最初由 Facebook 开发并维护。它于 2013 年正式发布，并因其独特的设计理念和技术优势，在前端开发领域迅速崭露头角，成为全球开发者广泛采用的前端框架之一。React 的主要目标是帮助开发者创建复杂的交互式用户界面，同时保持代码的高度可重用性和性能优化能力。它的设计哲学强调组件化开发模式，使得应用程序可以被拆解成多个独立的小型模块，从而提高代码的清晰度和可维护性。React 提倡通过组件来组织 UI 结构。每个组件都是一个独立的功能单元，负责渲染特定的部分页面逻辑和样式。这种基于组件的设计方式不仅提高了代码的复用率，还增强了项目的模块化程度。为了提升性能，React 使用了一种称为“虚拟 DOM”的机制。当状态发生变化时，React 首先会在内存中更新虚拟 DOM，然后计算新旧虚拟 DOM 的差异（Diff），最后仅将必要的更改同步到真实 DOM 上，以此减少昂贵的浏览器重新渲染操作。

React 实现了一个严格的数据流动模型——即单向绑定。这意味着父组件的状态变化会自动传递给子组件，但反过来则不会发生。这样的架构有助于追踪错误源头，同时也让整个系统的运行更易于预测和调试。虽然 JSX 不是必需品，但它极大地简化了 HTML 和 JavaScript 的混合编写过程。借助 JSX，我们可以直接在 JavaScript 文件里描述静态或者动态的内容结构，从而使模板更加直观易读。随着版本迭代，React 推出了 Hooks 功能，允许函数组件也能访问状态及其他 React 特性而无需转换为类形式。这进一步降低了复杂性的门槛，促进了轻量级解决方案的发展趋势。对于传统的类组件而言，存在一系列预定义好的钩子函数用来监听不同阶段的行为事件，比如挂载完成后的初始化处理或是卸载前清理资源等动作都可以在此处实现定制化的业务需求。综上所述，React 凭借其强大的功能集以及灵活的应用场景成为了现代 Web 前端不可或缺的一部分。无论是初学者还是资深工程师都能从中受益匪浅，只要掌握了上述提到的基础理论知识再加上实践积累经验，则完全可以胜任各种规模项目的要求。

3.2.3 ​jQuery技术

jQuery 是一款优秀的 JavaScript 库，最早由 John Resig 创建于 2006 年 1 月，并作为一个开源项目发布。该库旨在简化 HTML 文档遍历、事件处理、动画创建以及 AJAX 交互等常见任务，显著提升了前端开发效率。jQuery 提供了简洁的 API 来快速定位和操作网页上的元素。通过选择器机制，开发者可以用类似于 CSS 的语法规则轻松选取目标节点。

利用 jQuery，不仅可以实时调整 HTML 元素的外观属性（如颜色、大小等），还能高效地替换或追加文本内容。支持绑定多种类型的事件监听器至任意选定的目标上，确保能够及时捕捉到访客的各种行为反馈，例如点击、悬停、键盘输入等等。提供丰富的视觉特效，内置了一系列实用的过渡效果函数，像显示/隐藏、滑动切换、渐变透明度变化等功能一应俱全；同时还允许自定义复杂动画序列满足个性化需求。统一 AJAX 请求接口极大地方便了异步数据交换流程的设计实施工作，只需几行简单代码即可发起 GET 或 POST 类型请求并与服务器端通信交流信息。

除了专注于文档层面的操作外，jQuery 还扩展了一些辅助工具方法用于改善原生数组迭代等问题解决难度。总体来看，jQuery 凭借其强大而又简便易用的功能集合，在过去很长一段时间内主导着前端开发领域潮流走向。尽管近年来新兴框架层出不穷对其地位造成一定冲击，但对于许多中小型项目来说仍然是非常理想的选择方案之一。

4 系统实现

4.1 用户登录验证

用户在浏览器中输入网页地址http://127.0.0.1:5000/mgr/sign.html/即可打开系统主页登录页面，输入用户名和密码后验证通过才能进入系统主页，如果用户名或密码输入错误系统会给出错误提示。点击登录页面的注册账号按钮可跳转到用户注册页面，输入用户名和密码并确认后可完成新用户注册。系统后台基于Django框架操作MySQL数据库用户表进行实现，系统登录界面如图2所示，

 图2 系统登录界面

4.2 主页面展示

进入主界面后，前端自动向后端发送请求,在Django中使用MySQL进行查询，需安装Django-MySQLdb库，该库提供了与MySQL数据库的集成，在路由函数中接收参数即页数和每页显示的数据条数，使用SQL语句执行数据库查询，将查询结果传递给模板进行渲染，以展示分页数据，可视化界面如图4所示。



图4

4.3 客户展示页

主界面的第一栏目为国内外大客户信息展示页，其中展示了大客户的客户名，联系电话，地址等等，可以点击编辑按钮实现客户信息的修改，可删除客户；点击增加客户即可在管理界面编辑并添加客户信息到数据库；Django中对数据库表的操作，应该都通过 Model对象实现对数据的读写，而不是通过SQL语句。比如，这里我们要获取 customer 表所有记录，该表是和我们前面定义的 Customer 类管理的。Customer.objects.values() 就会返回一个QuerySet 对象，这个对象是Django定义的，在这里它包含所有的Customer 表记录。QuerySet 对象 可以使用 for 循环遍历取出里面所有的元素。每个元素 对应 一条表记录。每条表记录元素都是一个dict对象，其中 每个元素的 key 是表字段名，value是该记录的字段值上面的代码就可以将 每条记录的信息存储到字符串中返回给前端浏览器，可视界面展示如图5所示。

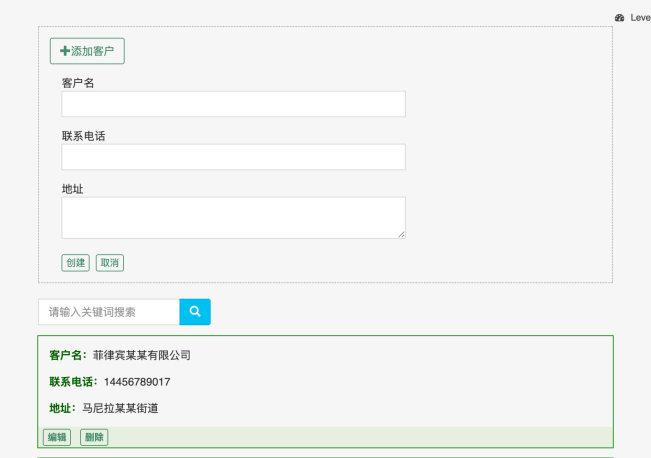


图5

4.4 产品展示页

主界面的第二栏目为海外仓货物产品信息展示页，其中展示了产品的信息，相关公司，描述等等，可以点击编辑按钮实现产品信息的修改，可删除产品；点击添加产品即可在管理界面编辑并添加客户信息到数据库；Django 里面，数据库表的操作，包括表的定义、表中数据的增删改查，都可以通过 Model 类型的对象进行的。通常，在Django定义一张数据库的表 就是定义一个继承自 django.db.models.Model 的类定义该表中的字段（列）就是定义该类里面的一些属性类的方法就是对该表中数据的处理方法，包括 数据的增删改查这样，开发者对数据库的访问，从原来的使用底层的 sql 语句，变成面向对象的开发，通过一系列对象的类定义和方法调用就可以操作数据库。这样做首先极大的简化了应用中的数据库开发，因为无需使用sql语句操作数据库了，提高了开发的效率；其次屏蔽了不同的数据库访问的底层细节，基本做到了开发好代码后，如果要换数据库，几乎不需要改代码，修改几个配置项就可以了。这种通过对象操作数据库的方法被称之为 ORM（object relational mapping）,具体界面如图6所示。

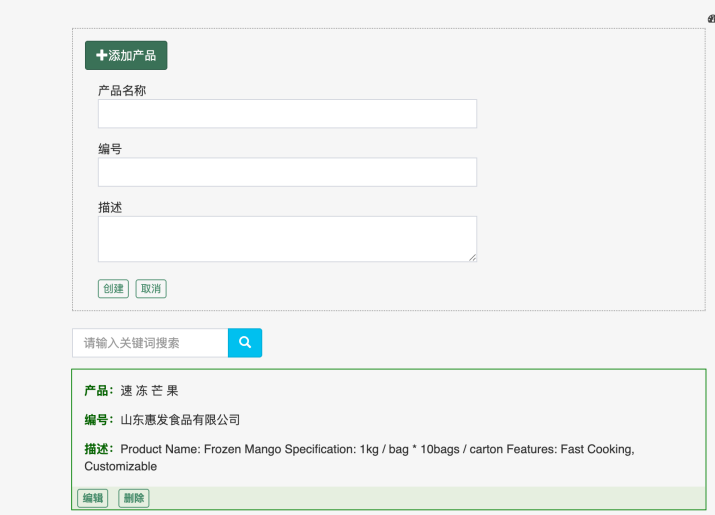


图6

4.5 订单页面

主界面的第三栏目为订单展示页，其中展示了订单的信息，订单日期，订单客户，采购信息等等，点击添加订单即可在管理界面编辑并添加客户信息到数据库，其中包括订单名称，客户信息只能在之前大客户栏目中添加过客户中浏览选择，产品信息也是只能在海外仓产品栏目中选择添加，一个客户可添加多个产品，在表内是一对多的关系，Order表里面一条订单记录的客户对应 Customer表里面的一条客户记录而条 Order记录里面的客户是可以对应 Customer 表里面 同一个客户记录的，反过来说，就是：一个客户记录可以对应多条订单记录这就是一对多的关系，可以用如下图片表示可视化界面如图7所示。像这种一对多的关系，数据库中是用外键来表示的。如果一个表中的某个字段是外键，那就意味着这个外键字段的记录的取值，只能是它关联表的某个记录的主键的值。在定义表的 Model类的时候，如果没有指定主键字段，migrate 的时候 Django 会为该Model对应的数据库表自动生成一个id字段，作为主键。





图7

4.6 海关模块

4.7 支付模块

5 系统部署

5.1 系统接口

5.2 系统开发与部署

本课题基于B/S架构,采用vscode开发环境和Python编程语言,系统后台开发采用django开发框架,通过mysql数据库驱动访问数据库,系统前端使用react,elementui,对结果进行可视化展示，系统部署环境采用macos操作系统，MySQL数据库版本8.0.19，django版本1.1.2，Python版本3.12。

6 结论与展望

本项目不仅重构了传统跨境贸易的基础设施，更通过技术创新与模式融合，为“中国制造”走向全球开辟了智能化通道。未来，随着5G、量子计算等技术的突破，海外仓将演变为数字贸易的核心枢纽，持续推动全球供应链向高效、弹性、可持续方向进化。建议企业紧抓政策红利，加速与平台生态的深度融合，共同绘制“买全球、卖全球”的新蓝图。为电商企业管理者提供决策支持。整个项目涵盖了数据处理、数据库存储、预测算法和数据可视化等关键功能，展示了Python在构建实用且功能强大的应用程序方面的能力。

本课题虽然采用了先进流行的技术框架，但实践中还需考虑其他影响因素来完善系统，以提高时效性和数据质量，给用户提供更好的推荐服务。由于时间有限、自身水平不足等各方面的原因，系统其他方面还有些不足需要完善，敬请各位专家、老师和读者批评指正。

# 致 谢

在这篇论文完成之际，我想向你们表达我最衷心的感谢和敬意。在整个毕设过程中，你们给予了我悉心的指导、支持和鼓励，是你们的帮助让我能够顺利完成这项工作。

首先，我要特别感谢我的毕设指导王东青教授。您在整个毕设过程中一直给予我无私的指导和鼓励。从选题阶段到论文修改，您都给予了我宝贵的建议和指导。您的专业知识和丰富经验让我受益匪浅，我学到了许多关于研究方法和学术写作的技巧。感谢您对我的悉心指导和耐心教导，我会倍加珍惜并运用在今后的学习和工作中。

其次，我要感谢帮助过我的同学们。你们在我遇到困难时给予了我无私的帮助和支持。无论是讨论问题、互相借阅资料还是互相激励，你们的存在让这段论文写作的旅程更加愉快和充实。感谢你们的友谊和合作，没有你们的支持，我无法顺利完成毕业论文。

此外，我还要感谢我的专业老师们。您们在课堂上传授的知识为我在毕设中的研究提供了坚实的基础。您们严谨的治学态度和渊博的学识给予了我深刻的影响，让我对所学领域有了更深入的认识。您们的教诲将成为我在未来学习和工作中的宝贵财富。

最后，我要向我的家人和朋友表达我的感激之情。你们一直以来对我的支持和鼓励是我前进的动力。在我遇到困难和挫折时，你们给予了我无限的勇气和信心。感谢你们的理解和支持，让我能够专注于论文的撰写和研究工作。

再次感谢所有给予我帮助和支持的人们。这篇论文的完成离不开你们的支持和鼓励。我将会继续努力充实自己，在所学领域取得更多的成就。感谢你们的陪伴和帮助，让我度过了这段难忘的论文写作时光。

# 参 考 文 献

1. 顾海斌.基于大规模电商数据的用户消费行为分析方法研究[D].吉林大学,2024.
2. 董云琪.基于用户行为时变特征的电商网站品牌推荐研究[D].湖南大学,2024.
3. 张子实.电子商务平台基于用户行为数据的消费预测研究[D].北京邮电大学,2018.
4. 范俊广.基于用户行为的日志分析系统的研究[D].吉林大学,2024.
5. 周翔,张鹏翼,王军.移动购物用户信息浏览特征及对购买的影响研究——基于移动电商APP点击流日志的分析[J].现代图书情报技术, 2018, 002(004):1-9.
6. 原康.基于GTM的增强型跨境电商服务平台的分析与设计[J].2024(04):04-30.
7. 邓重斌.基于SWOT理论的东南亚跨境电商平台探究分析———以Shopee为例[J].市场周刊·理论版, 2021(44):2.
8. 叶锋.Python最新Web编程框架Flask研究[J].电脑编程技巧与维护, 2015(15):2.
9. 美 格林布戈 Grinberg, Miguel.Flask Web开发:基于Python的Web应用开发实战[M].人民邮电出版社,2015.
10. 史宝坤,李欣,王淑娴,等.基于Flask的Python Web开发[J].数码世界, 2020(03):045.
11. 李超,徐云龙,华中伟,等.一种基于Python Flask的Web服务器端设计[J].信息与电脑, 2019(8):2.
12. 兰旭辉,熊家军,邓刚.基于MySQL的应用程序设计[J].计算机工程与设计, 2004, 25(3):3.
13. Widenius M,Axmark D,Dubois P.Mysql Reference Manual[M].O'Reilly & Associates, Inc.2002.
14. 杜波依斯 P,DuBios P.MySQL网络数据库指南[M].机械工业出版社,2000.

附 录