

报告编号:E5A8D8DDFC1942B1B4A47BA8B5D4FD16

送检文档:基于Docker容器的农产品溯源管理系统

作者:李佳霖

送检单位:河南科技大学

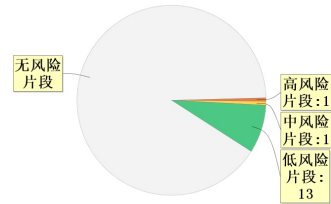
送检时间:2025-05-24 12:50:39

检测结果

疑似AIGC风险等级: **10.4% 低风险**

疑似AIGC片段: **15**

疑似AIGC片段 = 高风险片段+中风险片段+低风险片段



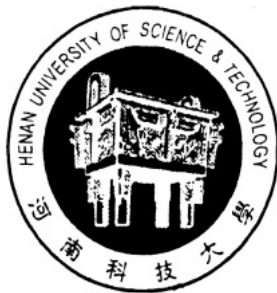
片段汇总列表

序号	送检片段	疑似AIGC概率
1	随着食品安全问题频频引发全社会的广泛关注，消费者对农产品来源的相关信息需求越来越强烈。为此，本系统构建了一套基于 Spring Boot框架 的农产品溯源管理系统，覆盖生产、加工、运输、销售、检测全流程。消费者只需用手机扫描商品的溯源码，即可轻松获取农产品的产地、加工、运输、检测、销售相关的详细信息，便于商品的来源追溯、责任落实，安全监管。系统使用Redis 缓存热点数据，对热点信息进行缓存，确保查询效率。同时使用 Spring AI 框架，集成通义千问-Max 大模型作为智能助手，帮助用户操作，降低用户学习成本提升满意度。用户可通过自然语言与 AI 助手交互，无需繁琐操作，助手可理解用户意图并代为完成各种操作。当用户表达不清或信息有误时，助手能主动引导用户明确需求，辅助用户做出准确决策。此外，系统结合 RAG技术，进一步提升 AI 回答的准确性，降低AI答非所问或失误的风险，满足农产品相关信息录入、查询的准确和便捷的需求。MSQL数据库使用Docker容器部署，在不同的环境上均可以运行，同时适合目前系统刚开始运行需要迭代的现实情况。	低
2	近年来，食品安全问题多次成为社会各界广泛关注的焦点，频频引起广大消费者的焦虑和不安。央视“3·15”晚会作为维护消费者权益的重要平台，在过去十年中曝光了数十起食品行业的热点事件。对晚会曝光的这些事件进行了深入的总结和分析，发现食品安全风险主要集中在非法添加、生产经营环境卫生不达标、掺假掺杂、虚假宣传以及篡改或超过保质期等方面。这些风险贯穿于食品生产经营的各个环节，从原材料采购到最终产品的宣传推广的所有环节，只要有一个环节出现纰漏都可能成为食品安全问题的源头。	低

序号	送检片段	疑似AIGC概率
3	对于普通消费者来说，这套系统的体验也很直观。扫一扫产品上的溯源码，种在哪、怎么种的、哪天检测过、谁运输的、什么时候上市的……这些关键节点一目了然，信息不再藏着掖着，整个流程透明得像一张白纸。可以说，这个系统不仅成了监管部门的“得力助手”，也实实在在守住了消费者的知情权 and 安全感，同时还顺势推动了农业数字化转型的脚步，一举多得，意义重大。	低
4	虽然我国在农产品溯源这块的起步比起一些发达国家晚了不少，但这几年，借着国家政策的东风和技术上的飞跃，我国已经逐渐追了上来，而且在多个方面都取得了不小的成果。	低
5	在本系统开发中，Pinia 被用来作为统一的状态管理工具，主要用于集中管理用户的登录信息、AI助手的对话记录、权限标识、页面主题设置等关键数据。相较于 Vuex，Pinia 的语法更加简洁，逻辑更直观，省去了很多模板式代码，让状态管理变得不再沉重。它的引入不仅提升了前端代码的整洁度，也让不同模块之间的数据交互变得更高效率，为整个系统的前后端协作打下了扎实基础。	低
6	Spring Boot 是 Pivotal 团队在 Spring 框架基础上打造的一款开源框架，说白了，它的最大特点就是“省事儿”。它通过“约定优于配置”的理念，帮我们省去了大量重复且繁琐的配置工作，让开发者可以把更多精力放在业务实现上，而不是各种集成细节里打转。这个框架自带一套自动化配置机制，只要添加上对应的依赖，像 Web 开发、数据库访问、安全控制这些常见功能就能直接用上，几乎是开箱即用。而且，它还集成了 Tomcat、Jetty 这类常用服务器，项目打成一个 JAR 包就能直接跑，部署和测试都变得特别方便。更棒的是，它和 MyBatis、Redis、Security、Swagger 等常见技术栈的配合也很顺滑，几乎不需要额外折腾。	低
7	MyBatis 是一个灵活度挺高的持久层框架，严格来说，它算是半自动化的 ORM（对象关系映射）工具。和 Hibernate 这类“全包式”的 ORM 框架不同，MyBatis 更偏向于让开发者手动编写 SQL，也就是说，开发者可以直接控制 SQL 的执行逻辑，这种方式在应对复杂业务需求时显得尤为实用，效率也往往更高。它主要通过 XML 配置或者注解，把 SQL 和 Java 方法一一对应起来，这种映射方式让数据库操作更直观，后续维护起来也比较顺手。MyBatis 的几个核心功能，比如动态 SQL、生动的 Mapper 接口、缓存机制和插件机制，都是为了解决实际项目中对数据层精细控制的需求，特别适合中大型项目使用。	高
8	Redis（Remote Dictionary Server）是一个基于内存、支持多种数据结构的高性能键值数据库，广泛应用于缓存、消息队列、会话管理等场景。它以极快的读写速度和丰富的数据类型（如字符串、哈希、列表、集合、有序集合等）著称，被誉为最受欢迎的 NoSQL 数据库之一。	低

序号	送检片段	疑似AIGC概率
9	在农产品溯源管理系统中，PageHelper 被用于对农产品分类、加工信息记录、运输信息记录、农产信息等数据进行分页展示。借助 PageHelper 提供的分页功能，系统能够高效处理大批量数据的查询请求，每次只加载一部分数据，同时使用Element-plus的分页展示组件，避免了数据一次性加载导致的页面卡顿与响应延迟问题，极大的提升了系统的稳定性，提高了用户满意度。	低
10	在农产品溯源管理系统的开发过程中，环境搭建方面也采用了更加现代化、灵活的方式。引入了 Docker 技术，将各类依赖组件打包成容器镜像，实现了开发环境的高度一致性和快速部署。通过 Docker，可以一键启动数据库、后端服务、前端界面等模块，极大地提高了系统的可移植性与运行效率。	低
11	可行性分析是一项复杂的综合研究，它不仅停留在表面判断，还需从时间成本、经济效益、操作流程、技术能力等方面进行全面调查和评估。通过这一过程，可以相对清晰地评估项目的可行性。为了找到一条最优的路线，尽可能地把各种信息资源安排得合理一些、整合得高效一些，这样项目在实施过程中才能把手头的资源发挥到最大，还能争取到该有的社会效益和经济效益。	低
12	该模块用于管理农产品的加工信息。在加工信息录入页面，用户可以通过下拉菜单选择相关信息，实现数据的快速录入。系统提供增加、删除、修改加工信息的功能，确保加工环节数据的完整性和可追溯性。	低
13	该模块用于管理农产品的运输信息，包括运输单位的基本信息、运输路线、运输时间等。系统提供增加、删除、修改运输信息的功能，确保运输环节的透明化和可追溯性。	低
14	这个模块主要是用来管理农产品的商品信息的。系统接入了阿里云的 OSS 对象存储服务，用来统一管理像图片、文档这些资源。这样一来，不管是上传还是访问这些文件，速度和效率都提升了不少。在数据校验方面，前端用了 Validation 框架，对用户提交的数据进行了一层严密的检查，目的是为了保证数据既完整又准确，同时也增强了整个系统的安全性。	低
15	通过这些功能模块的开发和整合，系统基本能覆盖农产品从种植到销售的整个过程，实现信息的全程追溯与管理。这样不仅能让食品安全监管更加透明、高效，也能更好地维护消费者的权益，同时也为农业信息化的推进提供了有力支持。	中

全文标红报告



题目 基于Docker容器的农产品溯源管理系统

姓 名 李佳霖

学 院 软件学院

专 业 软件工程

指导教师 谷志峰

2025年5月25日

学位论文写作声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立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，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

学位论文使用授权说明

本人完全了解河南科技大学关于收集、保存、使用学位论文的规定，即：按照学校要求提交学位论文的印刷本和电子版本；学校有权保留学位论文的印刷本和电子版，并提供目录检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在不以赢利为目的的前提下，学校可以将学位论文编入有关数据库,提供网上服务。（保密论文在解密后遵守此规定）

论文作者签名： 导师签名：

日期： 年 月 日

基于Docker容器的农产品溯源管理系统

摘 要

随着食品安全问题频频引发全社会的广泛关注，消费者对农产品来源的相关信息需求越来越强烈。为此，本系统构建了一套基于 Spring Boot框架 的农产品溯源管理系统，覆盖生产、加工、运输、销售、检测全流程。消费者只需用手机扫描商品的溯源码，即可轻松获取农产品的产地、加工、运输、检测、销售相关的详细信息，便于商品的来源追溯、责任落实，安全监管。系统使用Redis 缓存热点数据，对热点信息进行缓存，确保查询效率。同时使用 Spring AI 框架，集成通义千问-Max 大模型作为智能助手，帮助用户操作，降低用户学习成本提升满意度。用户可通过自然语言与 AI 助手交互，无需繁琐操作，助手可理解用户意图并代为完成各种操作。当用户表达不清或信息有误时，助手能主动引导用户明确需求，辅助用户做出准确决策。此外，系统结合 RAG技术，进一步提升 AI 回答的准确性，降低AI答非所问或失误的风险，满足农产品相关信息录入、查询的准确和便捷的需求。MSQL数据库使用Docker容器部署，在不同的环境上均可以运行，同时适合目前系统刚开始运行需要迭代的现实情况。

关键词：Spring AI；Redis；Spring Boot；通义千问-Max

A Traceability Management System for Agricultural Products Based on Docker Containers

ABSTRACT

With food safety issues frequently attracting widespread attention from society, consumers' demand for information regarding the sources of agricultural products is increasingly strong. To address this, the system has been built as an agricultural product traceability management system based on the Spring Boot framework, covering the entire process of production, processing, transportation, sales, and testing. Consumers only need to scan the traceability code of the product with their mobile phone to easily obtain detailed information related to the origin, processing, transportation, testing, and sales of agricultural products, which facilitates traceability

of the product's source, accountability, and safety supervision. The system uses Redis to cache hot data and ensure query efficiency. Additionally, the Spring AI framework is used to integrate the Tongyi Qianwen-Max large model as an intelligent assistant to help users operate, reduce learning costs, and enhance satisfaction. Users can interact with the AI assistant through natural language without the need for cumbersome operations; the assistant can understand user intentions and complete various tasks on their behalf. When users express themselves unclearly or provide incorrect information, the assistant can actively guide them to clarify their needs and assist them in making accurate decisions. Moreover, the system combines RAG technology to further improve the accuracy of AI responses and reduce the risk of AI misunderstanding the questions or making errors, meeting the demands for accurate and convenient information entry and querying regarding agricultural products. The MySQL database is deployed using Docker containers, allowing it to operate in different environments, and it is suitable for the current situation where the system is just beginning to run and needs iteration.

KEY WORDS: Spring AI ; Redis ; Spring Boot ; Tongyi Qianwen-Max

第1章 绪 论

§1.1 研究背景及意义

近年来，食品安全问题多次成为社会各界广泛关注的焦点，频频引起广大消费者的焦虑和不安。央视“3·15”晚会作为维护消费者权益的重要平台，在过去十年中曝光了数十起食品行业的热点事件。对晚会曝光的这些事件进行了深入的总结和分析，发现食品安全风险主要集中在非法添加、生产经营环境卫生不达标、掺假掺杂、虚假宣传以及篡改或超过保质期等方面。这些风险贯穿于食品生产经营的各个环节，从原材料采购到最终产品的宣传推广的所有环节，只要有一个环节出现纰漏都可能成为食品安全问题的源头。

虽然相关部门对食品行业的监管力度不断加大，但食品安全问题仍频繁发生，主要原因包括以下三点：

（1）企业主体责任落实不到位：部分食品企业缺乏社会责任感，为追求经济利益而忽视食品安全，面对监管也存在着侥幸心理，不能很好地遵守食品安全法律法规和标准规定，导致生产、加工、销售等环节出现问题。

（2）生产加工方面的监管仍十分薄弱：在食品行业中，上下游相关的加工企业非常多，涉及的环节也特别多，这就使得监管的难度也随之提高，不少地方面临人手不够、检测技术跟不上等问题，导致监管的作用难以落实到位。更关键的是，各个监管部门之间的信息流通也不够通畅，经常出现你管你的，我管我的等问题，很难做到信息资源的共享或者快速联动，所以在真正的执行层面，要实现食品安全的全流程的监管还有很长的路要走。

（3）消费者维权难的问题依然存在：虽然现在大家通过新媒体宣传对食品安全越来越关注，维权意识也比以前强了不少，但真碰上自己的权益被侵害，很多人还是会卡在“信息不对称”这一步——比如不知道责任在哪、也找不到对应的渠道维权，结果就是维权无门，权益得不到保障。

农产品溯源管理系统的设计初衷，其实就是为了破解当前农产品在流通环节中遇到的种种难题。它从消费者最关心的痛点出发，结合现代信息化技术，打出了一套“组合拳”——把农产品从种下那一刻开始，到采收、加工、运输，直到最终摆上货架、进入消费者手中的整个流程，全都详细记录在案。

这种全流程记录的做法，不仅让监管变得更高效，也让责任归属变得明确清晰。一旦哪一环出了问题，不会再互相推诿扯皮，直接对照数据就能找到源头。

对于普通消费者来说，这套系统的体验也很直观。扫一扫产品上的溯源码，种在哪、怎么种的、哪天检测过、谁运输的、什么时候上市的……这些关键节点一目了然，信息不再藏着掖着，整个流程透明得像一张白纸。可以说，这个系统不仅成了监管部门的“得力助手”，也实实在在守住了消费者的知情权 and 安全感，同时还顺势推动了农业数字化转型的脚步，一举多得，意义重大。

§1.2 国内外研究现状

农产品溯源系统通过追溯一件农产品从种下去到吃进去整个流程的信息，来保证食品的安全和质量。近几年，随着食品安全问题频频被曝光，无论是国内还是国外都对食品安全这个领域提高了重视程度，在不断探索和研究的過程中，已经取得了不少成果。下面就是对目前国内外在这方面研究和应用现状的一个简单汇总。

§1.2.1 国外研究现状

国外的农产品溯源管理系统的研究与推广起步较早，研究较深，值得借鉴他们的成功经验。

（1）欧盟：自从1997年爆发疯牛病之后，就建立起了一套完整的严格的牛肉追溯机制。例如牛耳标签、电子

数据库、动物护照、企业注册……一整套流程下来，牛从哪来、怎么养、卖到哪，消费者都能查得一清二楚。

(2) 日本：从2001年开始推广农产品和食品的追踪系统，到了2005年底，还建立起了粮农产品的认证制度，给每种产品制作了一张“身份证”。整个种植、加工过程中的各种信息都会被详细的记录下来，消费者只要一扫描或查一下，就能知道这东西从哪来的，放心吃。

(3) 美国则更多是从国家安全的角度看这个事。9·11事件之后，他们对食品追溯的重视程度达到了空前的高度。美国主要是靠行业协会和企业联合建立起来的，建立了家畜标识和可追溯体系。在关键时候，比如遇到疫病风险，他们能在48小时内迅速锁定所有相关企业。

总的来说，国外的做法大多是政府主导、企业参与，再加上各种高科技的加持，实现了对农产品从头到尾的追踪管理。整个体系成熟、高效，值得借鉴。

§1.2.2 国内研究现状

虽然我国在农产品溯源这块的起步比起一些发达国家晚了不少，但这几年，借着国家政策的东风和技术上的飞跃，我国已经逐渐追了上来，而且在多个方面都取得了不小的成果。

在研究层面，国内不少专家学者已经围绕物联网、区块链、大数据等热门技术展开了深度探索。这些研究的出发点就是提升系统的存储能力和查询效率，同时还得保证数据准确，和隐私信息的保护。而且这些研究并不是只停留在论文上，很多其实已经开始和现实场景结合起来了，比如在乡村振兴、电商平台等实际应用中，溯源系统都在发挥着作用。有些研究甚至深入到产业链内部，分析不同参与者在整个溯源体系中扮演什么角色。

从实践来看，地方上的尝试也越来越多。像辽宁省黑山县，在2022年4月上线了一个农产品质量安全溯源综合管理系统，到现在已经能覆盖二十多种农产品的溯源服务。这种落地应用对食品安全水平的提升起到了推动作用。

技术应用方面，物联网几乎成了最活跃的一块。各种传感器、智能设备再加上后台平台支持，逐步把“数字农场”、溯源码生成与打印、感知系统这些模块搭建了起来，让整个溯源流程变得更加智能、自动。

但是国内的溯源系统整体来看还是处在一个相对初级的阶段。虽然试点项目越来越多，但很多还只是局部性的尝试，覆盖面还比较有限。毕竟我们真正起步是在2002年左右，而像欧美国家那时候已经在实际应用层面走了不少年。

§1.3 研究方法 with 内容

本课题旨在设计并实现一个集成多方服务的农产品溯源管理系统，利用信息化平台，为消费者、监管部门、加工单位、运输部门和生产者提供高效、透明的服务。系统采用前后端分离的架构，前端使用 Vue 3 框架，结合 Element Plus UI 组件库和 Pinia 状态管理库，实现用户界面的动态交互和状态管理；后端基于 Spring Boot 框架，整合 MyBatis 持久层框架，提供稳定的业务逻辑处理和数据库访问支持。数据库方面，系统采用 MySQL 关系型数据库进行数据存储，Redis 内存数据库用于缓存处理，提升系统性能和响应速度。此外，系统还引入了 Spring AI 框架，接入阿里巴巴通义千问-Max大模型，提供智能问答服务；使用阿里云 OSS 对象存储服务，实现文件的统一存储与管理；采用 Google 布隆过滤器、RBAC 权限模型、ThreadLocal、检索增强生成 (RAG)、JWT 令牌、MD5 加密、Axios、Ajax、Validation 框架、Docker 容器和 PageHelper 等技术，提升系统的安全性、稳定性和可扩展性。

§1.4 人工智能助手模块

在农产品溯源管理系统的人工智能助手模块中，最初采用了 Spring AI 框架。然而，鉴于 Spring AI 主要支持国外的大模型（如 OpenAI、Azure OpenAI、Hugging Face 等），在集成国产大模型方面存在一定的限制，而且 Spring AI 现如今发布的仍然是快照版本，更新频繁，每当有更新就有可能需要修改底层代码逻辑或配置文件，影响系统稳定。为了更好地支持国产大模型并确保系统的稳定运行，最终更换成了阿里巴巴集团开发的 Spring AI Alibaba 框架。

Spring AI Alibaba 是在 Spring AI 基础上做的扩展，主要是为了适配阿里云的大模型服务。开发者用它接入像通义千问这样的阿里云百炼系列产品，会方便很多。这个框架提供了统一的 API 接口，可以覆盖多种生成式模型的应用场景，比如问答对话、文本生成图像、甚至生成语音这些功能。

除了基本功能外，它还内置了像 Prompt Template 和 OutputParser 这样的实用工具。只要在阿里云官网开通所需的服务，改一改配置文件，就能顺利接入阿里云的大模型，整个接入流程可以说是相当轻量化，开发体验也提升不少。

通过设置 application.yml 文件里的模型参数，系统可以比较灵活地在不同模型之间切换，比如 DeepSeek V3 alibaba、DeepSeek R1 alibaba、通义千问、通义千问-Max 等（这些模型目前都已经接入阿里云）。这种灵活的配置方式让系统能够快速适应各种业务场景，满足不同的需求。

通过使用 Spring AI Alibaba 框架，农产品溯源管理系统采用了通义千问Max大模型，实现了智能化的用户交互功能。用户可以通过自然语言指令与系统进行交互，大模型通过解析用户输入，判断信息的完整性，并引导用户补充必要的信息，最终自动完成相应的业务操作。例如，用户输入指令：“请插入一条商品信息，商品名称为苹果，未经过深加工。”系统接收到指令后，利用通义千问大模型进行语义解析，发现信息不完整（如缺少商品是否经过深加工的明确说明），系统将自动生成追问，例如：“请问该商品是否经过深加工？”用户回答系统的追问，例如：“是的，经过深加工。”系统在获取完整信息后，自动调用相应的业务接口，将商品信息插入数据库，并反馈操作结果给用户。AI 调用的接口与前端页面点击后信息传入后端调用的接口是一样的。只需要在接口上会添加一个注解，注解中写明这个接口的功能、条件和要求，然后将这个接口的方法名放入 AI的Controller 层中即可。

第2章 相关技术

§2.1 Vue 3 前端框架

Vue 3 是现在 Vue.js 的最新版本，由尤雨溪和他的团队打造，用来开发用户界面和单页应用（SPA）。简单来说，它就是 Vue 2 的升级版，不管是性能、写法，还是后期维护方面，都做了不少优化，是现代前端开发绕不开的一项技术。

说到 Vue 3 最大的亮点，非 Composition API（组合式 API）莫属。它换了一种更灵活的方式来组织组件逻辑，对于拆分功能和复用代码来说非常便捷，尤其在大型项目里优势特别明显。而且 Vue 3 抛弃了 Vue 2 里用 Object.defineProperty 实现响应式的方式，改用了 Proxy，性能更强，还能更好地支持类型系统。

除此之外，Vue 3 周边生态也很完善，比如 Vue Router 4 负责路由，Pinia 做状态管理，还有像 Vite 这样的构建工具，配合 Element Plus、ECharts 这些流行的组件库，基本能满足绝大多数的前端开发需求。

在农产品溯源管理系统的开发中，Vue 3负责搭建所有的交互界面，同时用 Axios 跟后端 API 对接，确保数据来回流畅，通过组件化的设计、响应式数据处理，实现了用户信息管理、AI助手、农产品分类、产品详情、图表看板等一系列功能模块。不仅让开发过程更高效，系统后期维护和迭代起来也省力不少。而且这样的架构也为系统后续扩展打下了扎实的基础，用户体验自然也随之提升。

§2.2 Element Plus 组件库

Element Plus 是一套为 Vue 3 设计的基于组件化开发思想的 UI 框架，由饿了么前端团队维护和开源。作为 Vue 2.x 时期广受欢迎的 Element UI 的继任者，Element Plus 完全重写以支持 Vue 3 的新特性，提供了更优秀的性能、更清晰的代码结构与更强的类型支持，广泛应用于后台管理系统、企业级中后台项目等场景。

Element Plus 提供了一整套非常丰富的组件，像表单、按钮、表格、弹窗、分页、通知这些常用的功能几乎都能找到，基本能覆盖前端开发里的大部分交互需求。而且这些组件的配置自由度很高，不仅支持多种主题风格，还有响应式设计，挺适合做现代化、专业级的前端界面。除了功能全面，它的文档也做得比较到位，还支持中英文切换，国外开发者用起来也很顺手。所有组件都遵循统一的交互和视觉规范，这样可以让项目整体看起来更一致，逻辑也更清晰，有助于提升用户体验。

在本系统的开发过程中，Element Plus 被大量用在管理后台的界面上，比如数据显示的表格、筛选条件的表单、分页组件、弹窗交互、提示消息、标签导航等等。这些组件的使用不仅让开发变得更轻松，也让系统界面更专业、更美观，整体的用户体验也得到了明显提升。

§2.3 Pinia 状态管理

Pinia 是 Vue 团队推出的一款官方推荐的状态管理库，也可以理解是 Vuex 的“接班人”。它是专门为 Vue 3 打造的，充分结合了 Composition API 的特性，用起来比之前轻便、清晰很多，而且还天然支持类型推导，特别适合现在这种类型安全越来越被重视的前端开发趋势。

Pinia 的理念很简单：用最少的学习成本，实现对状态的高效管理。它采用模块化的设计风格，每个 store 其实就是一个普通的 JavaScript 模块，里面可以定义状态（state）、计算属性（getters）和方法（actions），写法清晰直观，跟平时写组件代码没什么太大差别。更重要的是，它和 Vue 3 的配合非常紧密，几乎不用写多余的配置，状态共享就能轻松实现。同时，它支持像 SSR（服务端渲染）、插件扩展、状态持久化这些进阶功能，适用

范围也不只是小项目，大型企业级应用照样能应对。

在本系统开发中，Pinia 被用来作为统一的状态管理工具，主要用于集中管理用户的登录信息、AI助手的对话记录、权限标识、页面主题设置等关键数据。相较于 Vuex，Pinia 的语法更加简洁，逻辑更直观，省去了很多模板式代码，让状态管理变得不再沉重。它的引入不仅提升了前端代码的整洁度，也让不同模块之间的数据交互变得更高效率，为整个系统的前后端协作打下了坚实基础。

§2.4 Spring Boot 框架

Spring Boot 是 Pivotal 团队在 Spring 框架基础上打造的一款开源框架，说白了，它的最大特点就是“省事儿”。它通过“约定优于配置”的理念，帮我们省去了大量重复且繁琐的配置工作，让开发者可以把更多精力放在业务实现上，而不是各种集成细节里打转。这个框架自带一套自动化配置机制，只要添加上对应的依赖，像 Web 开发、数据库访问、安全控制这些常见功能就能直接用上，几乎是开箱即用。而且，它还集成了 Tomcat、Jetty 这类常用服务器，项目打成一个 JAR 包就能直接跑，部署和测试都变得特别方便。更棒的是，它和 MyBatis、Redis、Security、Swagger 等常见技术栈的配合也很顺滑，几乎不需要额外折腾。

在系统监控方面，Spring Boot 也提供了相当完善的支持。比如它的 Actuator 模块，不仅能实时查看应用状态、内存、线程等情况，还能对接像 Prometheus、Grafana 这样的专业监控平台。同时，它对多环境配置 (YAML 或 properties)、日志管理等也都有良好支持，这些都让它在现代微服务架构中有着天然优势。

在本系统的开发过程中，Spring Boot 构成了整个后端的骨架。从接口编写、权限控制，到 AI 模块的整合、数据库操作，基本都围绕着它来展开。它提供的开发效率、扩展能力以及背后庞大的社区支持，为系统的稳定运行和快速迭代提供了强有力的支撑，可以说是构建现代智能型 Web 系统不可或缺的核心技术之一。

§2.5 MyBatis 持久层框架

MyBatis 是一个灵活度挺高的持久层框架，严格来说，它算是半自动化的 ORM (对象关系映射) 工具。和 Hibernate 这类“全包式”的 ORM 框架不同，MyBatis 更偏向于让开发者手动编写 SQL，也就是说，开发者可以直接控制 SQL 的执行逻辑，这种方式在应对复杂业务需求时显得尤为实用，效率也往往更高。它主要通过 XML 配置或者注解，把 SQL 和 Java 方法一一对应起来，这种映射方式让数据库操作更直观，后续维护起来也比较顺手。MyBatis 的几个核心功能，比如动态 SQL、生动的 Mapper 接口、缓存机制和插件机制，都是为了解决实际项目中对数据层精细控制的需求，特别适合中大型项目使用。

在项目开发过程中，MyBatis 和 Spring Boot 的集成也非常自然。借助 Spring Boot Starter，可以很方便地引入并配置 MyBatis，再通过依赖注入管理 Mapper 接口，整个流程非常清晰省事。另外，它也支持多数据源和多表联查，在保证性能的同时，也给系统扩展预留了空间。

在本课题构建的系统中，MyBatis 主要负责后端与 MySQL 数据库之间的数据交互任务，简单的 SQL 语句就直接使用注解，复杂的 SQL，则使用 XML 配置。不管是用户信息的增删改查，还是一些复杂的查询、分页处理，它都能胜任。更重要的是，SQL 写得清清楚楚，这对后期维护很有帮助，也让系统在处理复杂数据结构时表现得足够稳定可靠。

§2.6 Redis 缓存技术

Redis (Remote Dictionary Server) 是一个基于内存、支持多种数据结构的高性能键值数据库，广泛应用于缓存、消息队列、会话管理等场景。它以极快的读写速度和丰富的数据类型 (如字符串、哈希、列表、集合、有序集合等) 著称，被誉为最受欢迎的 NoSQL 数据库之一。

在农产品溯源管理系统中，Redis 主要用于缓存热点数据 (如商品分类信息、可能频繁查询的热点数据)、存储临时状态信息 (如验证码、登录 Token 等) 以及减少对 MySQL 的频繁读取，防止大量的查询落到 DB 上导致数据库挂掉。用登录后的身份验证信息也存入 Redis 数据库，大幅提升了认证效率，减少了认证所需要的时间，AI 智能对话相关的数据缓存也借助 Redis 提供高并发支持。此外，利用 Redis 的键过期机制，确保了数据的时效性和安全性，为系统的高可用、高性能提供了有力支撑。

§2.7 Spring AI 人工智能集成框架

Spring AI 是由 Spring 官方推出的用于构建 AI 驱动应用的集成框架，打破了 Java 语言与 AI 大模型不能匹配的或匹配困难的问题，使得 Java 也可以像 Python，C++ 一样调用大模型来为用户服务，同时集成多个大模型的 Start，可以应对市面上比较常见的主流大模型，程序员不必再去考虑具体的模型参数，与调用一个 API 无异。

Spring AI 具备以下核心特性：

- (1)统一接口与组件封装：通过标准化 API 屏蔽不同 AI 平台间的差异，使切换模型或服务提供商变得更加简单；
- (2)Prompt 模板引擎：支持使用模板语言构建提示词（Prompt），提升与大模型交互的灵活性与可维护性；
- (3)上下文记忆与状态管理：配合 Redis、数据库等存储介质，实现对话历史和用户状态的持久化存储；
- (4)可扩展性强：可与 Spring Boot、Spring Security、Spring Cloud 等框架对接衔接。

在本系统中，Spring AI 的引入显著增强了用户交互体验。用户可通过自然语言与系统进行智能问答、查看农产品全流程信息、获取种植建议等，实现从传统查询到对话式智能的转变。Spring AI 的 Prompt 模板技术，使得 AI 回答更贴合业务语义，提升了响应的准确性与实用性。

§2.8 通义千问-Max 大语言模型

通义千问（Qwen）是由阿里云旗下的“达摩院”推出的一系列大语言模型，其 Max 版本是面向企业级场景优化的高性能模型，具备强大的自然语言处理、知识问答、文本生成、多轮对话等综合能力。作为国内领先的大模型之一，通义千问-Max 在中文语义理解、多任务执行能力、响应速度与稳定性等方面表现突出。

核心特点包括：

- （1）强大的中文理解与生成能力：针对中文语料进行深度训练，具备良好的语义识别能力与语言逻辑性，适配中文用户使用场景；
- （2）上下文记忆与多轮对话支持：支持多轮对话语境的自动保持，适合构建具备持续交互能力的智能应用；
- （3）任务通用性强：可广泛应用于智能问答、摘要生成、情感分析、文案撰写、代码生成等多个任务领域；
- （4）模型开放与API接入：提供 API 接口或通过 Spring AI 框架进行调用，降低集成门槛，提升系统智能化水平。

本系统使用阿里云旗下的通义千问-Max大模型作为AI助手功能的AI核心，凭借这阿里云针对中文设计的分词器，和其高效的理解能力，使用户可以通过自然语言方式完成对农产品信息的查询、农产品信息的插入、用户信息的修改等几乎所有系统功能，它还能辅助用户处理模糊性较强的语言输入，帮助用户发出正确的指令，同时使用 RAG 技术，大幅降低了市面上所有大模型都可能出现的，信息匹配错误，答非所问，调用错误的接口的问题。

通义千问-Max 还能够通过每日的业务操作，自动成长，越来越成熟，为整个农产品溯源平台注入了“懂业务、会交流、能成长”的智能化内核。

§2.9 PageHelper 分页插件

PageHelper 是一款 MyBatis 或 MyBatis-Plus 中常用的开源分页插件，用于在查询数据库时自动处理分页逻辑。它通过拦截 MyBatis 的 SQL 执行过程，在查询语句中自动添加 LIMIT 等分页参数，从而简化了开发者的分页实现流程。

主要特点如下：

- （1）零侵入式分页支持：不需要手动拼接 SQL，只需在查询语句前调用 PageHelper.startPage(pageNum, pageSize)，即可自动完成分页处理；
- （2）与 MyBatis 高度集成：支持各种 MyBatis 的查询方法，兼容 Mapper 接口、XML 配置等多种写法；
- （3）分页信息封装完善：PageHelper 提供了 PageInfo<T> 等分页结果封装类，包含当前页码、总页数、数据列表等常用分页信息，便于前端统一展示；
- （4）高性能：基于数据库原生分页（如 MySQL 的 LIMIT 子句），查询效率高，特别适合大数据量的表查询分页场景；
- （5）支持多种数据库：PageHelper 支持 MySQL、Oracle、SQL Server、PostgreSQL 等市面上常见的主流数据库。

在农产品溯源管理系统中，PageHelper 被用于对农产品分类、加工信息记录、运输信息记录、农产信息等数据进行分页展示。借助 PageHelper 提供的分页功能，系统能够高效处理大批量数据的查询请求，每次只加载一部分数据，同时使用Element-plus的分页展示组件，避免了数据一次性加载导致的页面卡顿与响应延迟问题，极大的提升了系统的稳定性，提高了用户满意度。

§2.10 Docker容器

在农产品溯源管理系统的开发过程中，环境搭建方面也采用了更加现代化、灵活的方式。引入了 Docker 技术，将各类依赖组件打包成容器镜像，实现了开发环境的高度一致性和快速部署。通过 Docker，可以一键启动数据

库、后端服务、前端界面等模块，极大地提高了系统的可移植性与运行效率。

开发者只需通过 Docker 配置相应镜像，即可快速开展工作，避免了繁琐的安装配置过程，也不需要担心版本兼容问题。

第3章 系统分析

§3.1 系统目标与背景

“本系统的目标是打造一个服务于农产品生产加工的上下游企业、检测单位和消费者的一个综合性系统。生产、加工、运输、销售的相关单位通过在本系统录入该产品在本单位负责的环节的详细信息进行备案，检验检测单位填写检验信息，消费者通过查询农产品的溯源码就可以查到任意一个生产加工环节中的详细信息，也可以看到相关检验信息，在发生食品安全问题时，相关单位也可以迅速通过商品的备案信息找出是哪一个环节出现了问题，便于监管和落实责任，同时也可以根据信息迅速召回问题产品，减少危害，保障公众健康与社会安全。

§3.2 可行性分析

可行性分析是一项复杂的综合研究，它不仅停留在表面判断，还需从时间成本、经济效益、操作流程、技术能力等方面进行全面调查和评估。通过这一过程，可以相对清晰地评估项目的可行性。为了找到一条最优的路线，尽可能地把各种信息资源安排得合理一些、整合得高效一些，这样项目在实施过程中才能把手头的资源发挥到最大，还能争取到该有的社会效益和经济效益。

有了这样一份扎实的分析，就可以在程序设计前期避开很多“坑”，比如一些一开始没考虑周全的小问题，很可能后面会演变成大麻烦。可行性分析正好就是在前面帮我们把这些潜在的问题找出来，为后续程序设计的顺利推进打好基础。

§3.2.1 时间可行性分析

在本系统开发过程中，采取了一套科学且务实的时间安排策略。像需求分析、功能架构设计、具体功能的开发，还有后期的系统测试这些关键阶段，都提前留出了足够的时间。整个项目推进得比较顺利，开发过程中，一边写代码，一边持续查找和整理各种技术资料，确保每个阶段都有稳妥的技术支撑。

期间，在老师的专业指导下，遇到的一些开发难题也都得到了比较有效的解决，项目整体进度基本都按计划走，没出现大的偏差。从目前的情况来看，不光有信心按时完成任务，甚至可能还能比预期提前收工。从时间这个维度来说，这次系统开发具有切实的可行性。

§3.2.2 经济可行性分析

在搭建农产品溯源管理系统的开发环境时，充分考虑了成本与效益之间的平衡。开发工具方面选用了 Cursor、Trae 和 IntelliJ IDEA 这几个主流平台，再结合 MySQL 数据库使用，整个系统也能很好地兼容常见的浏览器，基本不挑环境。这些工具几乎都是开源或免费的，开发者可以直接下载安装，完全不用担心软件授权的费用问题。

硬件方面只用一台普通的个人笔记本电脑就能完成全部开发工作，不需要额外购置高性能的专业设备，这一点极大地降低了成本开支。

§3.2.3 操作可行性分析

“农产品溯源管理系统使用了 Element Plus 的 UI 组件，采用极简风格界面，将核心功能模块集中在导航栏，以直观清晰的形式呈现，方便用户定位和操作。同时集成了 Spring AI 框架，可以直接通过对话来让 AI 去操作系统，几乎没有使用门槛。

§3.2.4 技术可行性分析

“Spring Boot，MyBatis，Vue 框架等知识都已认真学习，要使用的相关服务都有成熟的社区可供参考，Cursor，IDEA 编辑器也都熟练掌握。

§3.2.5 法律可行性分析

本论文所使用的代码属于社区开源项目无法律争议，阿里云服务等均已正常付费，所参考的资料如知识星球等内容已购买 VIP，代码的编写为本人独立完成，绝无任何抄袭、剽窃行为。

§3.3 功能性需求

§3.3.1 用户注册与登录模块

系统应支持用户通过前端表单完成注册与登录操作，并且前端和后端都要可以对输入的数据进行检验，如密码的长度和格式，邮箱的格式，用户名的格式等进行检验确保数据的合法性与完整性。对于热点数据和需要频繁查询的数据做好处理，降低数据库的压力。已经登陆成功的用户要有身份标识，不必重复登陆。前端要有用户状态

的管理能力。对于用户密码等敏感数据做好数据加密。

§3.3.2生产者信息管理模块

系统应支持添加、删除、修改生产者基本信息。在其它模块需要使用到该模块中的信息时应有可以调用已有信息的接口，如用户在录入农场信息时，通过下拉菜单选择对应负责人，实现农场与生产者的绑定关系。在多线程的环境要保持会话上下文的一致性，要确保数据一致。

§3.3.3加工单位管理模块

系统应支持添加、删除、修改农产品的加工单位信息。系统应支持通过下拉菜单快速选择相关信息，提升数据录入效率，同时也要保证保障加工数据的完整性与可追溯性，以支持后续查询和监管。

§3.3.4运输部门管理模块

系统应支持录入运输单位的基本信息，包括运输时间、路线、负责人等。并且可以对相关信息进行查询修改和删除。

§3.3.5 AI 助手模块

面对着众多的上下游企业和用户，系统应足够简单，要有帮助和引导客户进行操作的能力，利用AI技术简化和帮助用户的使用，同时要尽可能地避免AI出现答非所问、错误引导或错误操作的可能。

第4章 系统设计

§4.1 总体功能

在开发农产品溯源管理系统的过程中，尽可能地贴近实际业务需求，深入理解每一个流程背后的逻辑，然后一点点把这些需求拆解成具体功能，逐步构建成系统中的各个模块。功能规划上并没有一味追求“堆砌”，而是更注重实用性和整体的流畅体验。

技术选型这块，选择了 IntelliJ IDEA 和 Cursor 这两个比较顺手又专业的开发工具，后端的数据处理由 MySQL 来承担，搭建了一套高效又稳定的开发基础。整个系统采用了现在主流的 B/S 架构，也就是浏览器 / 服务器模式，好处特别明显——用户不需要安装任何客户端，只要打开 Chrome、Edge、必应这些常用浏览器就能直接用系统，简单方便，哪怕出差在外也能快速访问。

这种架构设计，不光是为现在服务的，也为以后留了很多余地。比如后续要加新功能、扩展业务线，系统本身就具备很好的可拓展性，不至于推倒重来。这一方面提升了使用体验，另一方面也让整个系统具备了长期演进的潜力。

§4.2 系统模块设计

§4.2.1 用户注册与登录模块

这个模块主要包括两个页面：用户注册和登录。前端这边用了 Element Plus 的表单校验功能，先对用户输入做一层基本的检查。后端则更严谨一些，使用了 Validation 框架来仔细验证传过来的数据，尽量保证数据既完整又精确。登录一旦成功，系统会生成一个 JWT 令牌。这个令牌前端是用 Pinia 状态管理来存着的，后端则把它放进 Redis 缓存，这样可以减少直接去查 MySQL 的次数，避免高并发时数据库压力太大，甚至挂掉。另外，像密码这样的敏感信息也做了加密处理，用的是 MD5 加密，多了一层保护，提升系统的安全性。

§4.2.2 生产者信息管理模块

该模块包括生产者信息和农场信息两个子模块。在农场信息页面，用户可以通过下拉菜单选择农场的负责人，实现农场与生产者的关联。系统提供增加、删除、修改农场信息的功能。为了确保多线程环境下各线程数据的一致性，系统利用 ThreadLocal 保存会话上下文信息。

§4.2.3 加工单位管理模块

该模块用于管理农产品的加工信息。在加工信息录入页面，用户可以通过下拉菜单选择相关信息，实现数据的快速录入。系统提供增加、删除、修改加工信息的功能，确保加工环节数据的完整性和可追溯性。

§4.2.4 运输部门管理模块

该模块用于管理农产品的运输信息，包括运输单位的基本信息、运输路线、运输时间等。系统提供增加、删除、修改运输信息的功能，确保运输环节的透明化和可追溯性。

§4.2.5 商品管理模块

这个模块主要是用来管理农产品的商品信息的。系统接入了阿里云的 OSS 对象存储服务，用来统一管理像图片、文档这些资源。这样一来，不管是上传还是访问这些文件，速度和效率都提升了不少。在数据校验方面，前

端用了 Validation 框架，对用户提交的数据进行了一层严密的检查，目的是为了保证数据既完整又准确，同时也增强了整个系统的安全性。

§4.2.6 AI 助手模块

这个模块集成了 Spring AI 框架，并接入了阿里巴巴的通义千问-Max大模型，用来提供智能问答服务。这样一来，用户在使用系统的时候，就不用花太多时间去摸索操作流程，整体体验也更友好。

系统采用了检索增强生成（RAG）技术，把内部的知识库和外部信息资源结合起来，进一步提高了大模型在回答问题时的准确率和实用性。

通过这些功能模块的开发和整合，系统基本能覆盖农产品从种植到销售的整个过程，实现信息的全程追溯与管理。这样不仅能让食品安全监管更加透明、高效，也能更好地维护消费者的权益，同时也为农业信息化的推进提供了有力支持。

§4.3数据库概念设计

在正式进入数据库建表环节之前，先从整体业务逻辑出发，进行了概念层面的数据库设计。说白了，这一步就像是画草图，把系统中所有可能涉及到的数据对象（比如农产品、生产商、运输节点、用户、溯源信息等）都罗列出来，再梳理清楚它们之间的关系。

（1）用户实体及属性如下图4-3-1所示：

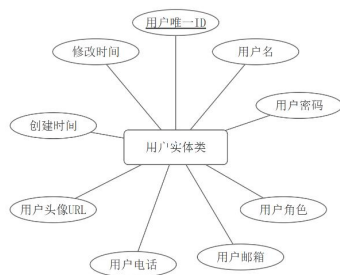


表4-3-1 用户实体及属性图

（2）商品分类实体类如下图4-3-2所示：

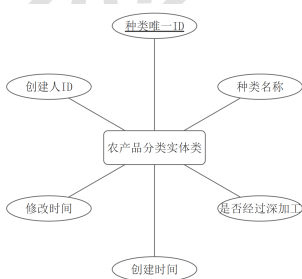


表4-3-2 农产品分类实体及属性图

（3）农场实体类如下图4-3-3所示：

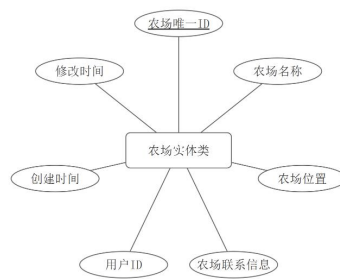


表4-3-3 农场实体及属性图

(4) 种植记录实体类如下图4-3-4所示：

表4-3-4 种植记录实体及属性图

（5）加工记录实体类如下图4-3-5所示：

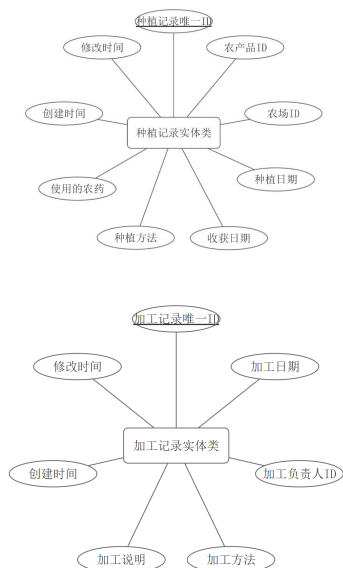


表4-3-5 加工记录实体及属性图

(6) 农产品实体类如下图4-3-6所示：

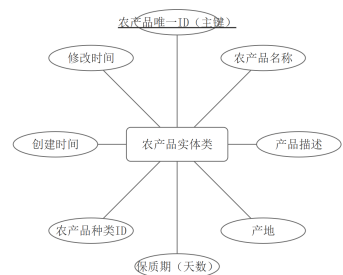


表4-3-6 农产品实体及属性图

(7) 农产品销售记录实体类如下图4-3-7所示：

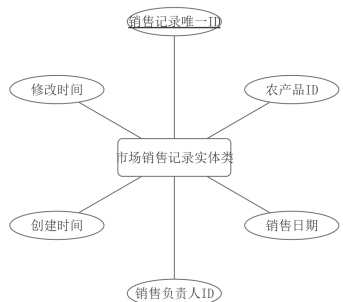


表4-3-7 销售记录实体及属性图

(8) 质量检测记录实体类如下图4-3-8所示：

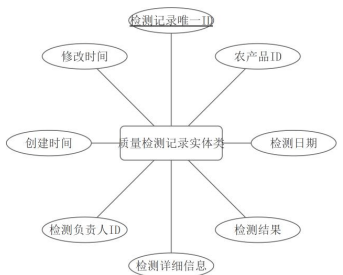


表4-3-8 质量检测实体及属性图

(9) 运输记录实体类如下图4-3-9所示：

表4-3-9 运输记录实体及属性图



(10) 总ER图如下图4-3-10所示：

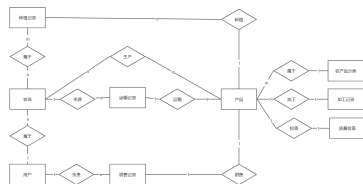


表4-3-10 总ER图

§4.3数据库表结构设计

在进行系统开发的过程中，数据库表结构设计这一环节可以说是整个系统的“地基”。从业务出发，理清各个模块之间的关系，比如产品信息、检测信息、用户信息、供应链节点等，再逐步拆解出对应的数据表结构。

每一张表的设计都尽量做到了结构清晰、字段合理。像主键这些基础设置自然是必不可少的，保证数据之间能正确地关联和调用。此外，还为部分关键字段添加了索引，加快了查询效率，避免后期系统访问量上来后出现卡顿的情况。

为了提高系统的灵活性和扩展性，在建表时也考虑到了以后可能会使用Hive来进行对数据进行数据加工和清理,在设计时就按字段信息分类好，字段命名上，则统一采用了具有明确业务含义的英文命名方式，确保可读性，方便后续维护。

整个表结构的搭建围绕“稳定、高效、可维护”这三个关键词来展开，力求为系统的稳定运行打下扎实的基础。

(1)用户表，记录系统用户的信息，包括用户名、角色、联系方式、头像等，如表4-1所示

表4-1 用户表

字段名	字段类型	长度	允许空	主键	说明
user_id	BIGINT	-	是	是	用户唯一ID
username	VARCHAR	255	否	否	用户名
password	VARCHAR	255	否	否	用户密码
role	VARCHAR	20	是	否	用户角色
email	VARCHAR	255	是	否	用户邮箱
phone_number	VARCHAR	20	是	否	用户电话
avatar	VARCHAR	255	是	否	用户头像URL
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(2) 农产品种类表，记录农产品的种类，如水果、蔬菜等，以及是否经过深加工等,如表4-2所示

表4-2 农产品种类表

字段名	字段类型	长度	允许空	主键	说明
category_id	BIGINT	-	是	是	种类唯一ID
category_name	VARCHAR	255	否	否	种类名称
is_processed	BOOLEAN	-	否	否	是否经过深加工
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间
user_id	BIGINT	-	是	否	创建人ID

(3) 农场表，记录农场的基本信息，包括名称、位置、联系方式等,如表4-3所示

表4-3 农场表

字段名	字段类型	长度	允许空	主键	说明
farm_id	BIGINT	-	是	是	农场唯一ID
name	VARCHAR	255	否	否	农场名称
location	VARCHAR	255	否	否	农场位置
contact_info	VARCHAR	255	是	否	农场联系信息
user_id	BIGINT	-	是	否	用户ID
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(4) 种植记录表，记录每个农产品的种植信息，包括种植日期、收获日期、使用农药等，如表4-4所示

表4-4 种植记录表

字段名	字段类型	长度	允许空	主键	说明
record_id	BIGINT	-	是	是	种植记录唯一ID
product_id	BIGINT	-	是	否	农产品ID
farm_id	BIGINT	-	是	否	农场ID
planting_date	DATE	-	否	否	种植日期
harvest_date	DATE	-	是	否	收获日期
planting_method	VARCHAR	255	是	否	种植方法
pesticide_used	VARCHAR	255	是	否	使用的农药
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(5) 加工记录表，记录农产品的加工信息，包括加工日期、加工方法、加工负责人等，如表4-5所示

表4-5 加工记录表

字段名	字段类型	长度	允许空	主键	说明
processing_id	BIGINT	-	是	是	加工记录唯一ID
processing_date	DATE	-	否	否	加工日期
user_id	BIGINT	-	是	否	加工负责人ID
Processing_method	VARCHAR	255	否	否	加工方法
description	TEXT	-	是	否	加工说明
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(6) 农产品表，记录每种农产品的详细信息，包括名称、描述、产地、保质期等，如表4-6所示

表4-6 农产品表

字段名	字段类型	长度	允许空	主键	说明
product_id	BIGINT	-	是	是	农产品唯一ID
name	VARCHAR	255	否	否	农产品名称
description	TEXT	-	是	否	产品描述
farm_id	BIGINT	-	是	否	产地
shelf_life	BIGINT	-	是	否	保质期
category_id	BIGINT	-	是	否	农产品种类ID)
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(7) 质量检测表，记录农产品的质量检测信息，包括检测日期、检测结果、检测负责人等，如表4-7所示

表4-7 质量检测表

字段名	字段类型	长度	允许空	主键	说明
check_id	BIGINT	-	是	是	检测唯一ID
product_id	BIGINT	-	是	否	农产品ID
check_date	DATE	-	否	否	检测日期
result	ENUM	-	否	否	检测结果
check_details	TEXT	-	是	否	检测详细信息
user_id	BIGINT	-	是	否	检测负责人ID
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(8) 市场销售表，记录农产品的销售信息，包括销售日期、销售负责人等，如表4-8所示

表4-8 市场销售表

字段名	字段类型	长度	允许空	主键	说明
sale_id	BIGINT	-	是	是	销售记录唯一ID
product_id	BIGINT	-	是	否	农产品ID
sale_date	DATE	-	否	否	销售日期
user_id	BIGINT	-	是	否	销售负责人ID
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

(9) 运输记录表，记录农产品的运输信息，包括运输日期、起始农场、运输追踪码等，如表4-9所示

表4-9 运输记录表

字段名	字段类型	长度	允许空	主键	说明
shipment_id	BIGINT	-	是	是	运输唯一ID
product_id	BIGINT	-	是	否	农产品ID
origin_farm_id	BIGINT	-	是	否	起始农场ID
destination	VARCHAR	255	否	否	目的地
shipment_date	DATE	-	否	否	发货日期
user_id	BIGINT	-	是	否	承运商ID
tracking_code	VARCHAR	255	是	否	运输追踪码
create_time	DATETIME	-	否	否	创建时间
update_time	DATETIME	-	否	否	修改时间

第5章 系统实现

本系统采用典型的前后端分离架构，后端使用 Spring Boot 框架，前端采用 Vue 3 与 Element Plus 组件库实现页面交互。数据库层采用 MySQL 进行数据存储，同时借助 Redis 进行缓存优化，使用 OSS对象存储服务（如阿里云OSS）管理用户头像与文件上传。

§5.1用户模块

§5.1.1 登录界面

用户可在此页面输入用户名和密码进行登录，密码会自动转换成*号防止密码泄露，如果还未注册，只需点击注册就可以进入注册页面。勾选记住我下次登录就无需再输入密码。

图5-1-1 登录界面



§5.1.2 注册界面

在此页面可以输入用户名密码进行注册，系统会自动校验电话号码和邮箱的格式，如果不符合，就会在上方以红色字体显示错误，并标明错误原因。



图5-1-1 登录界面

§5.2产品分类模块

§5.2.1 主页面

此页面为产品分类主页面，后面有修改和删除两个按钮，点击即可修改分类词条，点击添加分类就会弹出分类添加框。

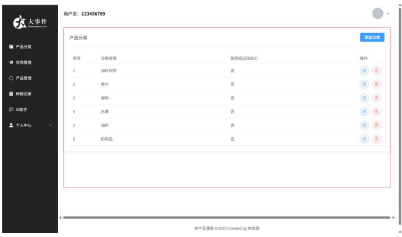


图5-1-1 登录界面

§5.2.2 添加分类页面

可以输入产品分类并且选择该分类是否经过深加工。成功添加后会有弹窗显示。

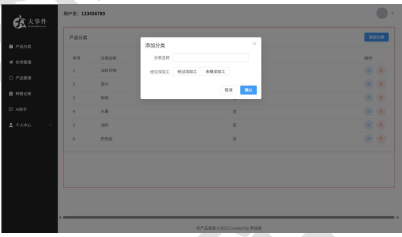


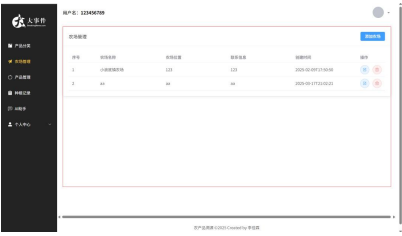
图5-1-1 登录界面

§5.3农场管理模块

§5.3.1 主页面

此页面为农产管理主页面，后面有修改和删除两个按钮，点击即可修改分类词条，点击添加分类就会弹出分类添加框。

图5-1-1 登录界面



§5.3.2 添加农场页面

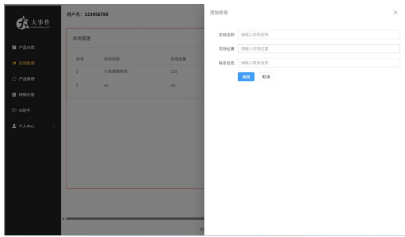
可以输入农场的详细信息。成功添加后会有弹窗显示。

图5-1-1 登录界面

§5.4产品管理模块

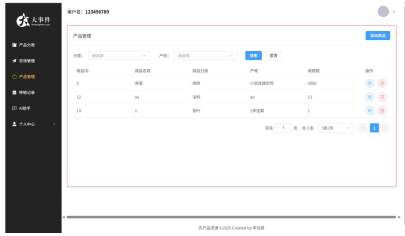
§5.4.1 主页面

此页面为产品管理主页面，后面有修改和删除两个按钮，点击即可修改分类词条，点击添加产品就会弹出分类



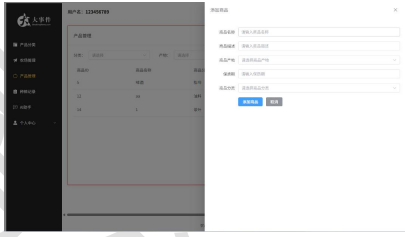
添加框，下方有跳转页码，可以选择一页上面有几条产品信息，也可以选择跳转到指定页面。

图5-1-1 登录界面



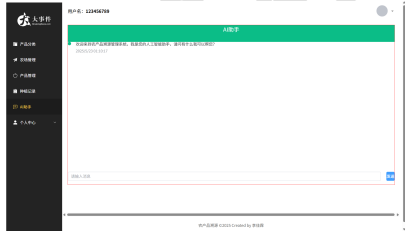
§5.4.2添加商品页面
此页面为添加商品主页面，其中商品产地输入框是一个下拉菜单，可以从以前录入的商品产地中选择，商品分类也是一个下拉菜单，可以从以前添加的商品分类中选择。

图5-1-1 登录界面



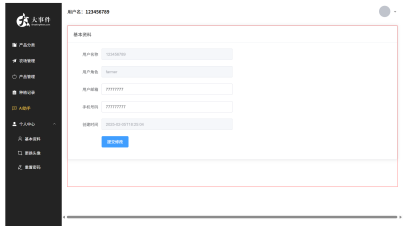
§5.5 AI助手模块
此页面是AI助手页面，用户只需要在此页面输入需求和信息，AI就可以自动调用后台接口，完成几乎本系统所有提供的可以的操作。

图5-1-1 登录界面



§5.6 用户管理模块
此页面是AI助手页面的第一个页面，可以修改注册时的用户信息，但用户id，用户密码和角色不可修改。

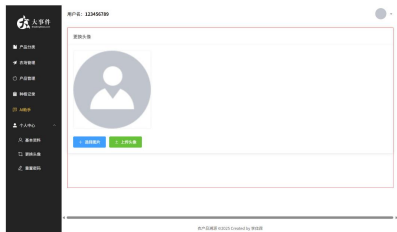
图5-1-1 登录界面



此页面是用户管理的第二个页面，可以在此页面上上传和修改用户头像。

图5-1-1 登录界面

第6章 系统测试
系统测试是系统开发的重要环节，旨在验证系统是否达到预期的功能需求、性能指标及稳定性要求。为保证农



产品溯源管理系统的稳定性与可靠性，本文在系统实现完成后，进行了详细的测试工作。

系统功能测试是对系统中各模块的核心业务逻辑进行验证，确保各功能模块按照设计要求实现。下列测试包括用户注册、登录、农产品管理、农场管理、种植记录、加工记录、质量检测、市场销售及运输记录等核心模块的测试内容，覆盖数据录入、信息查询、权限验证、数据更新与删除等多种操作。

§6.1 测试环境配置

测试环境如下：

操作系统：Windows 11

浏览器：Chrome / Edge

JDK版本：JDK 17

数据库：MySQL 8.0

服务器：Spring Boot 内置 Tomcat

前端环境：Vue 3 + Element Plus

测试工具：Postman、JMeter、Navicat、Chrome开发者工具

§6.2 用户模块测试

在本系统中，登录验证机制基于用户输入的登录名与密码信息展开。只有当登录名与密码完全匹配时，系统才会通过身份验证，允许用户获取访问权限。系统同时支持用户注册、信息修改、查看个人资料等功能，如表6-2

表 6-2

预期结果	实际结果	结论
注册成功，跳转至登录页面	成功	通过
登录成功，返回Token并跳转首页	成功	通过
登录失败，提示“账号或密码错误”	成功	通过
正确展示用户头像、邮箱、电话等信息	成功	通过
提交后用户信息变更成功	成功	通过

§6.3 农产品管理模块测试

农产品管理模块实现了产品的录入、修改、删除及展示等功能，支持管理员对农产品的基础信息进行维护，如表6-3

表 6-3

测试用例	输入信息	预期结果	实际结果	结论
添加农产品	输入产品名称、种类、产地等信息	新增成功，出现在产品列表中	成功	通过
修改农产品	修改保质期或产品描述等字段	修改成功，数据更新	成功	通过
删除农产品	点击删除按钮	该条农产品记录从列表中删除	成功	通过

§6.4 农场管理模块测试

农场管理模块负责维护农场的基本资料，如名称、地址、联系方式等。测试内容涵盖添加新农场、更新现有信息以及删除农场记录，确保数据操作逻辑正确，界面响应及时，信息准确保存，如表6-4

表 6-4

预期结果	实际结果	结论
新增成功，显示在农场列表中	成功	通过

提交后数据更新	成功	通过
成功删除该农场记录	成功	通过

§6.5 种植记录模块测试

种植记录模块是溯源系统的重要组成部分，主要记录产品的种植过程信息，包括种植时间、方法、使用农药等。测试内容涵盖记录添加、查看详情等操作如表6-5

表 6-5

实际结果	结论
成功	通过
成功	通过

§6.6 加工记录模块测试

该模块用于记录农产品的加工过程，支持用户录入加工方法、时间、人员等详细信息，系统应能正确保存和展示该记录。测试验证添加与查看功能的准确性和稳定性，确保加工信息真实有效，如表6-6

表 6-6

实际结果	结论
成功	通过
成功	通过

§6.7 质量检测模块测试

质量检测模块用于记录每批产品的检测信息，包括检测时间、检测机构、检测结论等，是保障农产品质量的重要环节。测试内容主要是添加检测记录及查询检测历史，确保系统能完整记录每项检测内容并正确展示，如表6-7

表 6-7

预期结果	实际结果	结论
成功添加，状态为待审核/合格/不合格	成功	通过
显示检测时间、检测人员、结果等	成功	通过

总结与展望

本次课题的目标是针对食品安全这个非常现实的问题，还有用户对农产品来源越来越透明的需求。围绕这个方向，搭建了一套功能全面、运行稳定的农产品溯源管理系统。

本系统前后端分离，后端用的是 Spring Boot框架，使用 MyBatis 和 Redis 等这些熟悉的技术栈，搭建起了系统稳定高效的业务逻辑部分；前端使用了Vue 3框架，再配合 Element Plus 和 Pinia组件，整体界面简洁、交互体验流畅，数据响应速度快。系统覆盖范围广，从种植、加工、运输、检测到最后的销售和售后，全流程覆盖。消费者只要扫一下商品上的溯源码，就能看到这个产品从田地到餐桌的全部旅程。这样既增强了消费者的信任感，也让监管机构工作更有依据。

同时还使用了 Spring AI 框架也用上了，接入了通义千问 Max 大模型，再结合 RAG 技术，搭建了一个AI智能助手。简单来说，用户现在可以直接“说话”来操作系统，比如查询产品信息、搜索种植记录这些，动动嘴就能搞定，大大提升了系统的智能体验和便利程度。

当然啦，现在系统虽然功能都基本没有什么问题可以运行起来，但要是真正推起来，还得继续优化，未来可以考虑几个方向继续打磨：

(1)数据安全和隐私保护这块，现在只做到了基本的信息安全，但本操作系统将存储和加工一些比较敏感的个人信息数据，后面还需进一步加强权限设置、加密机制，还有操作记录的审计功能，继续提升信息的安全性。

(2)多端适配的问题，现在就只有 Web 端，后期可以加入移动端和微信小程序，更加的方便客户随时随地的使用。

(3)AI 助手这块还能继续进化，一方面是让它学到更多农业相关知识，另一方面也得靠用户反馈来优化它的问答表现，最终希望它在农业知识图谱和数据分析上能更实用、更贴心。

(4)引入区块链技术，用来增强数据的可信度和不可篡改性，保证信息的准确性。

随着技术的持续进步与理念的不断更新，这套农产品溯源管理系统将会在保障食品安全、推动农业现代化方面

发挥越来越重要的作用。

参考文献

生产企业食品质量安全问题管理[J]. 赵敏.食品安全导刊,2024(16)

基于二维码防伪的农产品溯源优化对策——以金华市特色农产品为例[J]. 赵剑波;陆丹丹;姜霞;冯雨馨;张少源;陈乐怡.农业工程,2024(05)

基于Spring Boot的健身房管理系统的设计与实现[J]. 桑冉航;李晓明.电脑知识与技术,2023(22)

基于Springboot高并发Java Web开发模式[J]. 陈瑞.电脑编程技巧与维护,2019(04)

] 校园管理系统设计与实现. 李臻昊;杨永生.电脑编程技巧与维护,2024(01)

校园资助管理系统设计与关键技术研究. 吕雅婷.现代信息科技,2020(04)

人事工资信息管理系统设计. 刘峰民;刘洋;赵小娟.甘肃科技,2019(23)

试验测试数据管理系统设计与实现. 金燕;李静;雷鸣;张浩杰.国外电子测量技术,2020(02)

一种云计算支持下的智能煤矿管理系统设计. 李雷军;屈建国;姬鹏飞;黄天尘.中国科技信息,2024(23)

基于大数据的智慧消防管理系统设计探索. 陈天.信息与电脑,2024(23)

基于数据库技术的高校选修课管理系统设计与实现. 李曼.科技创新导报,2021(22)

基于微信小程序的被服清点管理系统设计与实现. 顾亚;龚林松;张长珍.无线互联科技,2023(19)

基于物联网技术的设备标识管理系统设计与开发. 章馨如;李哲威;陈国军;陆敏;江盛洋.通信世界,2024(01)

基于SpringBoot的分布式市政运维分析平台[J]. 韩佳彤;杜芳;甄必鑫;张宏娜;乔鹏程;敖登巴拉;王春涛.内蒙古大学学报(自然科学版),2022(03)

基于SpringBoot的云主机资源采集与可视化系统设计[J]. 黄安妮;符嘉成;潘俊冰;冯淞耀;陈柏龄.电脑编程技巧与维护,2023(03)

一种图标管理方法的设计与实现[J]. 张光明;张浩洋;陈肖勇.计算机时代,2023(04)

基于Vue的百度智能云图像识别模型技术的实现[J]. 孙艺宁.电脑知识与技术,2024(33)

中医信息采集系统服务器构建[J]. 刘子硕;刘国华.南开大学学报(自然科学版),2024(06)

面向实训基地管理系统的门锁控制网络终端设计. 梁亚峰;苏龙.电子技术与软件工程,2021(20)

基于Spring Boot与Vue的系统管理模块开发探究[J]. 杨妍.电声技术,2019(02)

基于MySQL的科研信息管理系统数据库设计[J]. 徐小卫;杨文超.信息与电脑(理论版),2022(11)

致 谢

岁月如流，冬去春来，疫情期间，随着本科学位论文的完成，四年的大学时光也接近尾声，即将进入新的人生阶段。本课题的顺利完成离不开许多老师、同学以及家人朋友的支持与帮助，在此谨向所有给予我关心、指导和鼓励的人表示最诚挚的感谢。

首先，衷心感谢我的指导老师谷志峰老师，在整个毕业设计过程中，老师始终给予我耐心指导与细致讲解。从选题确定、系统设计到论文撰写的每一个阶段，老师都提出了宝贵的意见，使我受益匪浅，深刻体会到严谨治学的精神。

其次，感谢各位专业课老师在本科期间的辛勤授课与悉心教导，使我在Java编程、数据库设计、Web开发等方面打下了坚实基础，为本课题的开展提供了理论支持和技术积累。

同时，感谢我的同学和朋友们在项目开发过程中给予的帮助和鼓励，大家的技术讨论和经验分享让我在遇到困难时不再孤单，也激励我不断前行。

最后，特别感谢我的家人对我始终如一的理解、包容与支持，是他们给予我坚持到底的勇气和动力，让我能专心完成本次毕业设计。

毕业设计不仅是对我四年学习成果的总结，更是我人生旅程中宝贵的经历。再次感谢所有帮助和陪伴我的人，愿在未来的学习和工作中继续努力，不负所学，砥砺前行。

说明

- 1、疑似AIGC概率越大，风险等级越高，代表AI生成可能性越大。
- 2、系统检测结果仅作为校方评价学生论文的参考依据之一，建议不作为最终评价依据。学生论文最终评价，以校方评价标准和评价结果为准。



关注微信公众号

官方网站: co.gocheck.cn

客服热线: 400-699-3389

客服QQ: 800113999

国家科学技术大学