基于springboot+vue的智慧农田管理平台设计与实现

Smart Farmland Management Platform Based on Springboot+Vue

**一．选题意义及背景**

智慧农田建设的意义在于通过科技手段提升农业生产效率和粮食安全水平，实现精准监测、智能管理、智能决策，优化农业产品标准输出，对现代农业园区建设起到示范带动作用。它有助于降低生产风险，提高农业生产的可预测性和可控性，同时也关注生态环境的保护和可持续发展。

智慧农田目前的现状，中国正在大力推广智慧农田建设，并取得了一定的成效。然而，由于资金投入不足、技术水平不高、缺乏专业人才等原因，智慧农田建设还面临许多困难和挑战。因此，需要进一步加大政策扶持力度，加强技术研发和人才培养，推动智慧农田建设的发展，以保障粮食安全和农业可持续发展。智慧农田建设的背景是全球人口的不断增长和粮食需求的不断增加，传统的农田管理方式已经无法满足现代农业发展的需要。因此，智慧农田建设成为了解决这些问题的有效途径之一。智慧农田建设不仅可以提高农作物的产量和质量，还可以降低生产成本、节约资源、保护环境，具有很大的经济效益和社会效益。

**二．主要内容：**

智慧农田管理平台：主要分为农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块

农场主页公告：农场系统主页公告。

农场统计分析：

农场环境监测：环境数据、设备统计、视频监控

农田信息：查看农田的信息，例如农田的温度、空气湿度、土壤ph值、光照情况、水泵。

农田管理：负责各块农田的管理，新增农田和编辑农田。

农场物质管理：

物质采购：农场种植农田所需要的物质采购记录，例如化肥、除草剂等物资

物资库存：农场剩余物资的库存记录

农作物出售：农作物产品的出售记录

系统管理：系统公告管理、农场员工管理、系统角色管理、系统菜单管理。系统管理属于管理员模块，只有管理员才能进系统管理。可以设置公告和对职工进行管理，职工的角色进行管理以及分配系统权限给职。

**计划**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |

摘 要

利用网络和数据库技术，本论文的主要目标是通过互联网技术来解决传统农田方案，对农田进行实时监测、管理和决策支持，以提高农业生产效率、资源利用率和农产品质量安全。进行在智慧农田管理平台功能需求分析：通过调研和分析当前市场上拥有的智慧农田系统，确定智慧农田管理平台的功能模块，农田管理模块、农资管理模块、系统管理模块、环境监测分析模块、系统首页模块等；设计在智慧农田管理平台架构和模块设计：根据功能需求分析的结果，设计在智慧农田管理平台的系统架构和各个模块的功能和交互方式，确保系统能够满足用户的需求，并具备良好的用户体验；实施智慧农田管理平台的方案探讨：探讨智能化的实施方案，包括系统的部署和配置、用户培训和支持等，以确保系统能够顺利地被农户流畅使用。

**关键词：**智慧农田管理平台 农田管理模块 农资管理模块 系统管理模块 环境监控分析模块

**ABSTRACT**

Using network and database technology, the main goal of this paper is to solve the traditional farmland scheme through Internet technology, and carry out real-time monitoring, management and decision support for farmland, so as to improve agricultural production efficiency, resource utilization and agricultural product quality safety. Conduct functional requirement analysis on the smart farmland management platform: Through research and analysis of the smart farmland systems currently available in the market, determine the functional modules of the smart farmland management platform, including farmland management module, agricultural material management module, system management module, environmental monitoring and analysis module, and system homepage module; Design the architecture and module design of the smart farmland management platform: Based on the results of functional requirements analysis, design the system architecture and the functions and interaction modes of each module on the smart farmland management platform to ensure that the system can meet the needs of users and have a good user experience; Discussion on the Implementation Plan of Smart Farmland Management Platform: Explore the implementation plan of intelligence, including system deployment and configuration, user training and support, to ensure that the system can be smoothly used by farmers..

**Key words:** Smart Farmland Management Platform, Farmland Management Module, Agricultural Material Management Module, System Management Module, Environmental Monitoring and Analysis Module

**目 录**

目录

[第一章 绪论 - 3 -](" \l "_Toc172020653)

[1.1 研究背景 - 3 -](" \l "_Toc172020654)

[1.2 研究目的和意义 - 3 -](" \l "_Toc172020655)

**[1.2.1 研究目的](" \l "_Toc172020656)** [- 3 -](" \l "_Toc172020656)

**[1.2.2 研究意义](" \l "_Toc172020657)** [- 3 -](" \l "_Toc172020657)

[1.3 论文结构 - 4 -](" \l "_Toc172020658)

[第二章 相关研究和综述 - 5 -](" \l "_Toc172020659)

**[2.1](" \l "_Toc172020660)** [传统农场管理的问题 - 5 -](" \l "_Toc172020660)

[2.2智慧农田管理平台系统的发展历程 - 5 -](" \l "_Toc172020661)

[2.3智慧农田管理平台的功能和特点 - 6 -](" \l "_Toc172020662)

[2.4系统相关技术 - 6 -](" \l "_Toc172020663)

[第三章 智慧农田管理平台系统需求分析 - 8 -](" \l "_Toc172020664)

[3.1系统需求分析 - 8 -](" \l "_Toc172020665)

**[3.1.1系统的可行性分析](" \l "_Toc172020666)** [- 8 -](" \l "_Toc172020666)

**[3.1.2功能描述](" \l "_Toc172020667)** [- 9 -](" \l "_Toc172020667)

**[3.1.3功能需求](" \l "_Toc172020668)** [- 9 -](" \l "_Toc172020668)

[3.2系统总体设计 - 9 -](" \l "_Toc172020669)

**[3.2.1 系统的体系结构](" \l "_Toc172020670)** [- 9 -](" \l "_Toc172020670)

**[3.2.2 用例建模](" \l "_Toc172020671)** [- 10 -](" \l "_Toc172020671)

[3.3系统详细设计 - 12 -](" \l "_Toc172020672)

[第四章 智慧农田管理平台设计 - 13 -](" \l "_Toc172020673)

[4.1系统架构设计 - 14 -](" \l "_Toc172020674)

[4.2数据库设计 - 14 -](" \l "_Toc172020675)

**[4.2.1创建数据库](" \l "_Toc172020676)** [- 14 -](" \l "_Toc172020676)

**[4.2.2数据表结构](" \l "_Toc172020677)** [- 15 -](" \l "_Toc172020677)

[4.3界面设计 - 20 -](" \l "_Toc172020678)

**[4.3.1普通员工界面设计](" \l "_Toc172020679)** [- 20 -](" \l "_Toc172020679)

**[4.3.2管理员界面设计](" \l "_Toc172020680)** [- 24 -](" \l "_Toc172020680)

[4.4系统流程设计 - 27 -](" \l "_Toc172020681)

[4.5功能模块设计 - 28 -](" \l "_Toc172020682)

**[4.5.1系统公告](" \l "_Toc172020683)** [- 28 -](" \l "_Toc172020683)

**[4.5.2农场环境监测](" \l "_Toc172020684)** [- 29 -](" \l "_Toc172020684)

**[4.5.3农田信息](" \l "_Toc172020685)** [- 30 -](" \l "_Toc172020685)

**[4.5.4 农田管理](" \l "_Toc172020686)** [- 32 -](" \l "_Toc172020686)

**[4.5.5物资采购](" \l "_Toc172020687)** [- 32 -](" \l "_Toc172020687)

**[4.5.6物资库存](" \l "_Toc172020688)** [- 34 -](" \l "_Toc172020688)

**[4.5.7农作物](" \l "_Toc172020689)** [- 34 -](" \l "_Toc172020689)

[4.5.8试题管理 - 35 -](" \l "_Toc172020690)

[第五章 总结和展望 - 37 -](" \l "_Toc172020691)

[参考文献 - 38 -](" \l "_Toc172020692)

[致 谢 - 39 -](" \l "_Toc172020693)

第一章 绪论

随着PC互联网的快速发展，对基于Web的万物智联发展起到了极大的促进作用。本文使用springboot+vue框架技术开发了一个基于Web的智慧农田管理平台。该系统有效解决了传统考试带来的弊端。

* 1. 研究背景

智慧农田建设的意义在于通过科技手段提升农业生产效率，实现实时监测、智能管理，提升优化农产品的质量，对现代农业园区建设起到示范带动作用。它有助于降低生产风险，提高农业生产的可预测性和可控性，同时也关注生态环境的保护和可持续发展。

智慧农田目前的现状，中国正在大力推广智慧农田建设，并取得了一定的成效。然而，由于资金投入不足、技术水平不高、缺乏专业人才等原因，智慧农田建设还面临许多困难和挑战。因此，需要进一步加大政策扶持力度，加强技术研发和人才培养，推动智慧农田建设的发展，以保障粮食安全和农业可持续发展。智慧农田建设的背景是全球人口的不断增长和粮食需求的不断增加，传统的农田管理方式已经无法满足现代农业发展的需要。因此，智慧农田建设成为了解决这些问题的有效途径之一。智慧农田建设不仅可以提高农作物的产量和质量，还可以降低生产成本、节约资源、保护环境，具有很大的经济效益和社会效益。

* 1. 研究目的和意义
     1. **研究目的**

本论文的研究目的是设计并开发在线管理农田管理平台解决方案和系统，以推动智慧农田建设的发展。通过对智慧农田管理平台的系统功能需求分析、系统设计和实施方案的探讨，本论文旨在为各大农场提供一个可行的解决方案，以满足未来农场可以通过在线系统区管理各个区域的农田。

* + 1. **研究意义**

智慧农田管理平台利用物联网、云计算、大数据等现代信息技术，对农业生产全过程进行实时监控和精细管理，其优势主要体现在以下几个方面：

提高生产效率：通过精准的数据分析和预测，帮助农民科学合理地制定生产计划，减少生产成本，提高农业生产效率。实现农业生产的数字化、信息化和智能化，提高农业生产效率、降低生产成本，并确保农产品质量安全。促进农业现代化发展，推动农业向中高端迈进，同时提高生态效益，减少环境污染，并保护生态环境。提供多项功能，如自动信息检测与控制、自动灌溉、自动降温等，以及气象数据和土壤数据监测。建立完善、规范、精确、协同的信息化管理平台，提升运营效率，使管理模式具有可扩展性和可复制性。实现精准科学的成本和利润核算体系，以及全产业链的“三层”质量追溯体系。实现销售预测、生产计划和库存管理的高效循环，加快周转率降低损耗，保证高投入产出比，降低经营风险。

智慧农田管理平台通过这些优势，为农业生产提供了高效、经济、环保的解决方案，有助于推动农业的可持续发展。

* 1. 论文结构

智慧农田管理平台，使用基于springboot+vue+MySQL等主流的web开发技术。本文各章节的内容如下：

1. 绪论。本章主要介绍了智慧农田管理平台的研究背景，阐述了研究目的和意义，并简要介绍了论文的结构。
2. 相关研究和综述。本章将对与智慧农业管理平台相关的技术和理论进行综述，包括计算java编程语言、数据库管理MySQL技术、前端vue技术。通过市场调研，发现传统农田管理中存在的问题，并指出智慧农田管理平台的功能和优点。
3. 对智慧农田管理平台需求分析。本章将对在线考试系统的功能需求进行详细分析，包括：农场统计分析、农田管理、农场农资管理、系统管理等方面。
4. 智慧农田管理平台的设计和实现。本章将对在线考试系统的系统架构和各个模块的设计进行详细描述，并介绍系统的实施方案和部署流程。
5. 总结和展望。本章将对全文进行总结，并展望智慧农田管理平台的未来发展方向和应用前景。
6. 相关研究和综述

**2.1** 传统农场管理的问题

传统农场经营管理人才缺失：随着土地流转和规模化经营的趋势，新型农业经营主体需要与产业链上下游进行有效沟通和整合，但许多经营主体由个体农民转变而来，缺乏相应的管理和整合经验。

生产管理粗放：农场普遍存在缺乏统一的生产管理和种植计划，以及主观式、粗放式种植管理的问题，导致农事操作不规范、排产与采收管理混乱。

种植技术落后：农场仍在使用传统的种植技术，依赖大量化肥和农药，这些做法导致土壤质量下降，影响作物产量和品质。

农产品品质不一：由于缺乏标准化生产和有效统筹，农产品品质不一，市场供应不稳定，影响品牌建设和销售。

产后流通损耗大：农产品在流通环节的损耗严重，影响农场收入，中国农产品在流通环节的损耗高达30%，远高于发达国家的5%以内。投入成本控制：不合理使用水肥资源造成成本过高，影响经济效益。

监管问题：随着企业规模的扩大，如何有效监管旗下农场成为一个难题，传统管理方式难以满足现代化农场的需求。

气候多变风险高：全球气候变化带来的极端天气事件增多，增加了农场运营的不确定性。

信息数字化问题：数据采集、处理和分析能力不足，导致决策滞后，难以实现精准农业。

这些问题表明传统农场管理在人才、技术、管理体系和市场适应性等方面存在明显的不足，制约了农业的现代化发展进程。

2.2智慧农田管理平台系统的发展历程

1. 初期阶段。最早的智慧农田管理系统主要是基于简单的网页或电子表格形式，用于收集各块农田的基本信息和每年的投入和产出情况。这些简单的功能，只能满足于以前的系统简单需要钱，并不满足于当下互联网时代的要求。
2. 中期阶段。随着技术的发展，智慧农田管理平台系统逐渐增加了更多的功能，例如农场统计分析：环境检测和农田信息列表，农场物质管理：物资采购、物资库存和农作物出售记录等功能。
3. 现代阶段。智慧农田管理平台具：农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块。能够满足绝大多数农场对农田管理的要求。

2.3智慧农田管理平台的功能和特点

智慧农田管理系统是一套集成了现代信息技术的综合管理平台，旨在提高农业生产的智能化水平和效率。以下是智慧农田管理系统的几个关键功能特点：

实时环境监测：系统通过安装在农田中的传感器实时监测土壤湿度、温度、光照等环境参数，确保作物获得最佳生长条件。

数据分析与决策支持：收集的数据通过云计算平台进行分析，为农民提供关于灌溉、施肥等农业活动的科学决策支持。

远程监控和管理：农民可以通过移动设备或电脑远程访问农田数据，为农民提供关于灌溉、施肥等农业活动的科学决策支持

自动化控制：系统可以远程控制灌溉电磁阀等设备，实现自动化灌溉，节约水资源和劳动成本。

视频监控：高清视频监控系统允许农民远程查看作物生长情况，及时发现并处理问题。

数据分析和市场预测：系统能够根据农田数据和市场需求提供生产优化方案和市场趋势预测，帮助农民做出更有利的市场决策。

支持多种通信协议：智慧农田管理系统通常支持多种通信协议，如LoRa/NB-IoT，确保数据的稳定传输。

这些功能共同构成了智慧农田管理系统的核心，使农业生产更加精准、高效和可持续。

2.4系统相关技术

1. 计算机网络技术。计算机网络技术是在智慧农田管理平台的基础，它提供了农田和农户之间的信息传输和交互平台。在智慧农田管理平台中，计算机网络技术需要具备高速、稳定和安全的特点，以确系统的顺利运行。常用的计算机网络技术包括局域网（LAN）、广域网（WAN）和互联网等。
2. 数据库管理技术。数据库技术是计算机科学技术的一个重要分支，涉及数据的组织、管理、存储和检索。它是信息系统的核心组成部分，支持数据的共享性，并实现对数据的统一控制。数据库技术的发展经历了层次模型、网状模型和关系模型等阶段，目前关系模型是最成熟和应用最广泛的数据模型。数据库的分类：数据库可以根据数据模型的不同分为两大类：关系型数据库和非关系型数据库（NoSQL）。常用数据库：MySQL：开源、多平台的关系型数据库，广泛应用于Web应用和中小型项目。PostgreSQL：开源、功能强大的关系型数据库，支持高级特性，如GIS和JSON。Oracle Database：商业数据库，以其高性能和稳定性著称，适用于大型企业级应用。SQL Server：微软开发的关系型数据库管理系统，适用于Windows环境和企业级应用。DB2：IBM出品的数据库，具有高度的可扩展性和兼容性。
3. Springboot技术：Spring Boot 是一个开源的Java-based框架，用于创建独立运行的。它旨在简化新Spring应用的初始搭建以及开发过程，并使现有Spring应用程序现代化。Spring Boot的核心特性包括自动配置、起步依赖（starter POMs）、嵌入式服务器支持以及Spring Boot Actuator等。核心概念和优势。自动配置：Spring Boot能够根据项目依赖和类路径中的bean来自动配置Spring应用。起步依赖：Spring Boot提供了一系列起步依赖，这些依赖包含了开发特定类型Spring应用所需的所有基本库。嵌入式服务器：Spring Boot支持内嵌Tomcat、Jetty或Undertow等服务器。Spring Boot Actuator：提供了生产就绪的特性，如度量信息收集等和健康检查。
4. 智慧农田管理平台系统需求分析

3.1系统需求分析

**3.1.1系统的可行性分析**

可行性分析是对一个项目或系统进行评估和分析，以确定其是否可行和值得实施。对于智慧农田管理平台的可行性分析，可以从以下几个方面进行论述：

1. 技术可行性。智慧农田管理系统是一个集成了先进信息技术和互联网技术的系统，它能够对农业生产过程进行全面监测和管理，实现农业生产的智能化、精准化和高效化。技术可行性分析通常涉及以下几个方面：智慧农田管理系统依赖于互联网技术、大数据分析、人工智能等现代信息技术。这些技术共同构成了系统的技术框架，使得系统能够收集农田环境数据、作物生长信息，并基于这些数据进行分析和决策支持。系统优势，智慧农田管理系统的优势在于它能够助力农业现代化，通过系统系统提供的农田环境报告，进而提高产量并减少资源浪费。此外，系统还能促进数据共享与交流，提高农业资源利用率，从而推动整个农业产业的协同发展。
2. 经济可行性。经济可行性分析需要考虑投资回报率和投资风险。智慧农业的实施虽然需要一定的初始投资，但通过提高生产效率和降低资源浪费，可以在中长期内实现成本节约和收益增加。此外，政府政策和市场发展潜力也是评估经济可行性的重要因素。智慧农田管理系统的实施可能会面临技术风险、市场风险和经济风险。为了应对这些风险，需要进行周密的市场调研，建立风险管理机制，并确保技术的稳定性和安全性。
3. 操作可行性。智慧农田管理平台系统，直观的用户界面：现代Web系统通常采用响应式设计，确保在各种设备上都有良好的显示效果。界面设计遵循用户体验原则，使用户能够直观地理解如何导航和执行任务。简化的操作流程：通过合理的流程设计，Web系统可以引导用户一步步完成操作，减少复杂性。例如，表单填写、注册流程等都被设计为多步骤向导，每一步都提供必要的指导和反馈。自动化和智能化功能：许多Web系统集成了自动化工具，如智能填充、建议搜索和自动校正等，这些功能可以减少用户的输入工作量，提高操作效率。清晰的导航结构：有效的导航菜单和面包屑路径帮助用户快速定位当前位置并回到先前的页面，减少迷路的可能性。

**3.1.2功能描述**

根据需要，智慧农田管理平台：主要分为农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块。整个过程参与有管理员和农场员工两种用户类型。按照功能和权限进行角色划分，可划分为：

管理员：管理系统中所有的功能，可以自行定义系统。

农场员工：农场员工，可以划分为为采购人员、销售人员、农田管理人员。

**3.1.3功能需求**

1. 用户管理。智慧农田管理平台：管理员和农场员工两种类型的用户，每个用户都有自己的账号和密码，可以通过登录系统来访问其功能。
2. 农田管理，管理员或者农场员工可以自行新增农田、编辑农田、删除农田和导出农田生成Excel表格。
3. 农田信息：可以查看农田的基本信息，例如农作物名称、数量、生长状态、文档、空间湿度、土壤湿度、农田负责人等基本信息
4. 农场物资管理：农场采购人员和销售人员负责的板块，分别对农场物资的采购和农作物的销售记录。
5. 系统管理：系统管理由管理独自拥有的权限，负责对系统公告管理、农场员工管理、系统角色管理、系统菜单管理。

3.2系统总体设计

项目树

## **前端项目分析**

前端项目基于 Vue.js 开发，文件结构如下：

复制

src/

├── App.vue # 根组件

├── main.js # 应用入口文件

├── assets/ # 静态资源

│ └── gloable.css # 全局样式文件

├── components/ # 组件文件夹

│ ├── Aside.vue # 侧边栏组件

│ ├── Header.vue # 顶部导航组件

│ └── Identify.vue # 验证组件

├── router/ # 路由配置

│ └── index.js # 路由配置文件

├── store/ # 状态管理

│ └── index.js # Vuex 店铺配置

├── utils/ # 工具函数

│ └── request.js # 请求工具函数

└── views/ # 页面视图

├── 404.vue # 404 页面

├── Dashbord.vue # 仪表盘页面

├── Farmland.vue # 农田管理页面

├── Home.vue # 首页

├── Im.vue # 即时通讯页面

├── Inventory.vue # 库存管理页面

├── Login.vue # 登录页面

├── Manage.vue # 管理页面

├── Menu.vue # 菜单页面

├── Notice.vue # 通知页面

├── Password.vue # 密码管理页面

├── Person.vue # 个人信息页面

├── Purchase.vue # 采购管理页面

├── Register.vue # 注册页面

├── Role.vue # 角色管理页面

├── Sales.vue # 销售管理页面

├── Statistic.vue # 统计页面

└── User.vue # 用户管理页面

## **后端项目分析**

后端基于 Spring Boot 开发，采用典型的分层架构，包括控制器、服务层、数据访问层等。

### **项目结构**

复制

src/

└── main/

├── java/

│ └── com/

│ └── farmland/

│ └── intel/

│ ├── SpringbootApplication.java # 应用启动类

│ ├── common/ # 公共工具类

│ │ ├── Constants.java # 常量类

│ │ ├── Result.java # 统一返回结果类

│ │ └── RoleEnum.java # 角色枚举类

│ ├── component/ # 组件

│ │ └── WebSocketServer.java # WebSocket 服务

│ ├── config/ # 配置类

│ │ ├── CorsConfig.java # 跨域配置

│ │ ├── InterceptorConfig.java # 拦截器配置

│ │ ├── MybatisPlusConfig.java # MyBatis Plus 配置

│ │ ├── SwaggerConfig.java # Swagger 配置

│ │ ├── WebSocketConfig.java # WebSocket 配置

│ │ └── interceptor/ # 自定义拦截器

│ │ ├── AuthAccess.java # 认证访问

│ │ └── JwtInterceptor.java # JWT 拦截器

│ ├── controller/ # 控制器

│ │ ├── FileController.java # 文件控制器

│ │ ├── InventoryController.java # 库存控制器

│ │ ├── MenuController.java # 菜单控制器

│ │ ├── NoticeController.java # 通知控制器

│ │ ├── PurchaseController.java # 采购控制器

│ │ ├── RoleController.java # 角色控制器

│ │ ├── SalesController.java # 销售控制器

│ │ ├── StatisticController.java # 统计控制器

│ │ └── UserController.java # 用户控制器

│ │ └── dto/ # 数据传输对象

│ │ ├── UserDTO.java # 用户数据传输对象

│ │ └── UserPasswordDTO.java # 用户密码数据传输对象

│ ├── entity/ # 实体类

│ │ ├── Dict.java # 字典实体

│ │ ├── Farmland.java # 农田实体

│ │ ├── Files.java # 文件实体

│ │ ├── Inventory.java # 库存实体

│ │ ├── Menu.java # 菜单实体

│ │ ├── Notice.java # 通知实体

│ │ ├── Purchase.java # 采购实体

│ │ ├── Role.java # 角色实体

│ │ ├── RoleMenu.java # 角色菜单关联实体

│ │ ├── Sales.java # 销售实体

│ │ ├── Statistic.java # 统计实体

│ │ └── User.java # 用户实体

│ ├── exception/ # 异常处理

│ │ ├── GlobalExceptionHandler.java # 全局异常处理器

│ │ └── ServiceException.java # 自定义异常

│ ├── mapper/ # 数据访问层接口

│ │ ├── DictMapper.java # 字典数据访问接口

│ │ ├── FileMapper.java # 文件数据访问接口

│ │ ├── InventoryMapper.java # 库存数据访问接口

│ │ ├── MenuMapper.java # 菜单数据访问接口

│ │ ├── NoticeMapper.java # 通知数据访问接口

│ │ ├── PurchaseMapper.java # 采购数据访问接口

│ │ ├── RoleMapper.java # 角色数据访问接口

│ │ ├── RoleMenuMapper.java # 角色菜单关联数据访问接口

│ │ ├── SalesMapper.java # 销售数据访问接口

│ │ ├── StatisticMapper.java # 统计数据访问接口

│ │ └── UserMapper.java # 用户数据访问接口

│ ├── service/ # 服务层接口

│ │ ├── IInventoryService.java # 库存服务接口

│ │ ├── IMenuService.java # 菜单服务接口

│ │ ├── INoticeService.java # 通知服务接口

│ │ ├── IPurchaseService.java # 采购服务接口

│ │ ├── IRoleService.java # 角色服务接口

│ │ ├── ISalesService.java # 销售服务接口

│ │ ├── IStatisticService.java # 统计服务接口

│ │ └── IUserService.java # 用户服务接口

│ └── utils/ # 工具类

│ ├── TableColumn.java # 表格列工具类

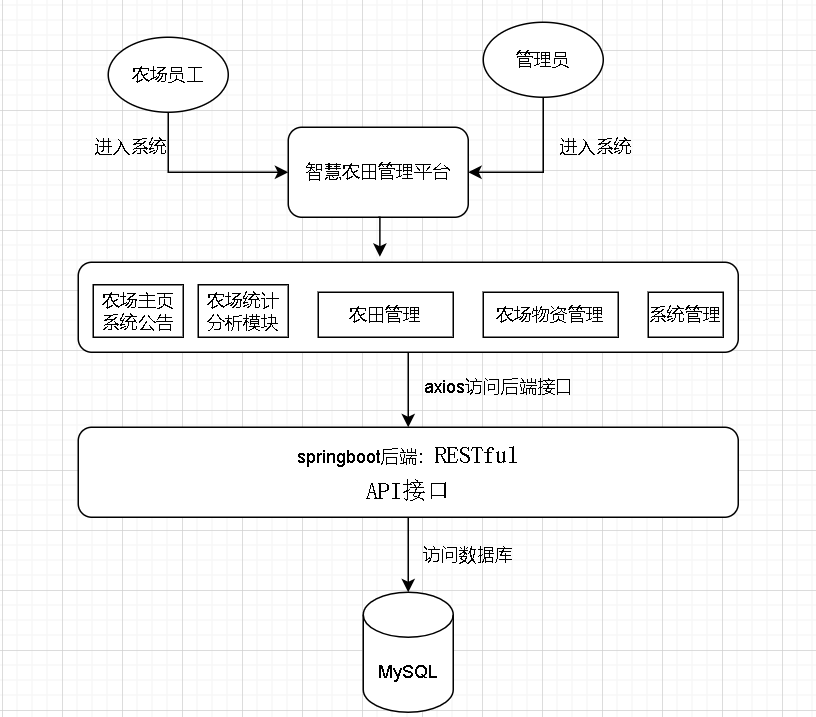
│ └── TokenUtils.java # Token 工具类

└── resources/ # 资源文件

└── mapper/ # MyBatis 映射文件

**3.2.1 系统的体系结构**

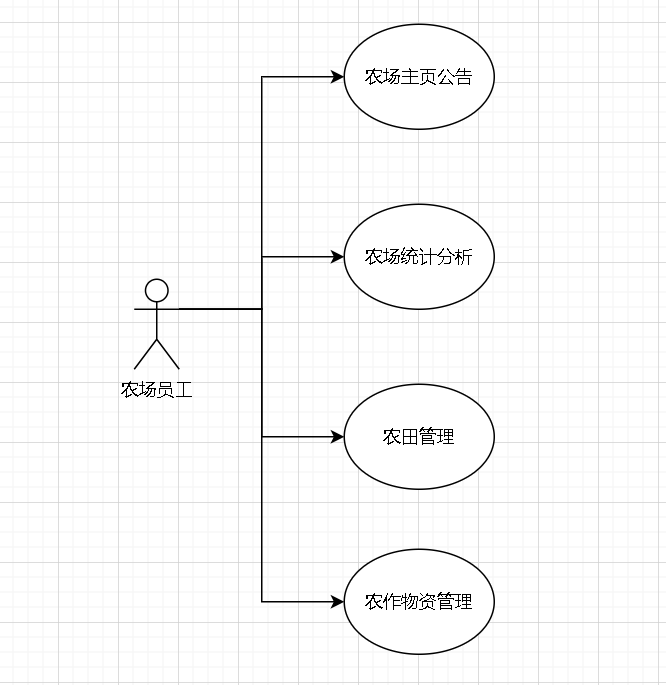
系统参用使用springboot+vue前后端分离架构，将系统分为两个层次，springboot的RESTful API 来处理来自 Vue 前端的 HTTP 请求。创建 Controller 来映射 URL 到相应的处理方法，以及 Service 和 Repository 层来封装业务逻辑和数据访问逻辑，同时配置跨源资源共享（CORS）策略，以便前端能够从不同的域进行请求。Vue 前端负责构建用户界面和处理用户交互，前端vue使用 Axios 或其他 HTTP 端库来发送请求到 Spring Boot 后端。实现了系统的高内聚、低耦合的设计目标。设计体系结构模型如图3-1所示。



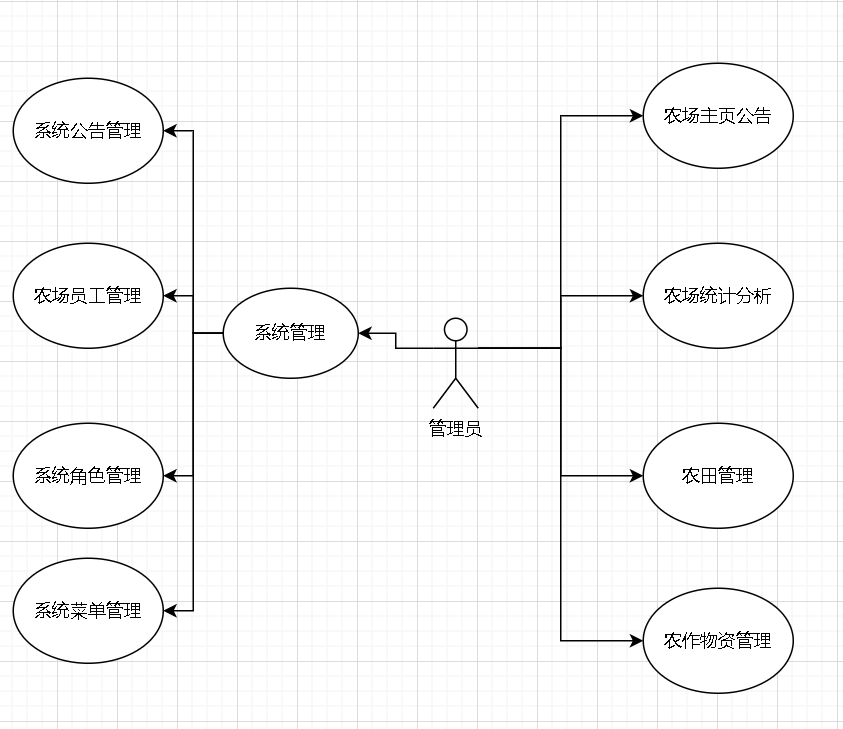
**图3-1 系统体系结构图**

**3.2.2 用例建模**

智慧农田管理平台的主要目标围绕在线可以实时管理农田。围绕这个主题，我们可以分别将动态划分出主要分为农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块。农场员工登录后，如图3-2所示。当管理员登录系统后，管理员进行的操作如图3-3所示。



**图3-2 学生用例图**



**图3-3 管理员用例图**

3.3系统详细设计

农场管理员工或者员工分别输入自己的账号和密码，通过身份的验证，分别进入各自的系统主界面，农场管理员可以：操作系统所以模块包括：农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理。农场员工登录后，可以对农场主页系统公告通知、农场统计分析、农田管理、农作物资管理等模块管理。

普通员工模块：

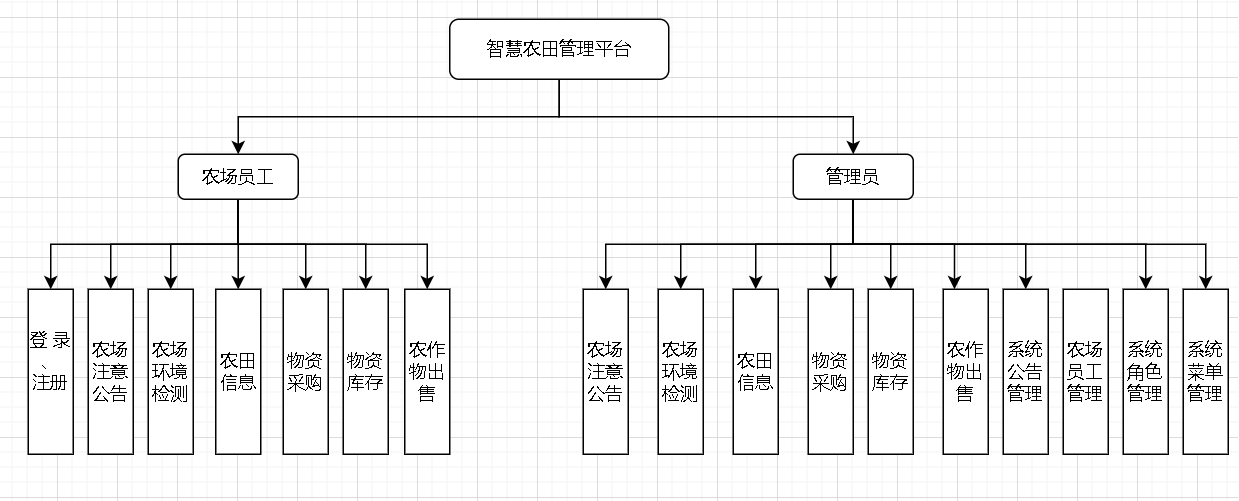
1. 登录：输入账号和密码进行登录操作。
2. 农场主页公告：查看农场系统公告
3. 农场统计分析：查看农场环境监测结果和农田基本信息
4. 农田管理：新增、编辑、删除农田，对农田进行管理
5. 农作物资管理：
6. 物质采购：农场种植农田所需要的物质采购记录，例如化肥、除草剂等物资；
7. 物资库存：农场剩余物资的库存记录，对物资库存的入库、出库进行记录登记。
8. 农作物出售：农作物产品的出售记录，新增、删除、编辑出售的农作物，同时可以导入excel和导出到excel表格。

管理员模块：

1. 登录：输入账号和密码进行登录操作。
2. 农场主页公告：查看农场系统公告
3. 农场统计分析：查看农场环境监测结果和农田基本信息
4. 农田管理：新增、编辑、删除农田，对农田进行管理
5. 农作物资管理：
6. 物质采购：农场种植农田所需要的物质采购记录，例如化肥、除草剂等物资；
7. 物资库存：农场剩余物资的库存记录，对物资库存的入库、出库进行记录登记。
8. 农作物出售：农作物产品的出售记录，新增、删除、编辑出售的农作物，同时可以导入excel和导出到excel表格。
9. 系统管理
10. 系统公告管理：可以新增、编辑、删除系统公告。
11. 农场员工管理：可以新增、编辑、删除农场员工，同时可以通过excel导出。
12. 系统角色管理：负责系统的角色（采购人员、销售人员、农田负责人、系统管理员）进行管理，可以新增角色、编辑角色、删除角色，以及给角色分配菜单功能
13. 系统菜单管理：负责对系统功能菜单进行管理，编辑菜单、新增菜单、删除菜单等操作
14. 智慧农田管理平台设计

4.1系统架构设计

在设计智慧农田管理平台的架构时，需要考虑系统的可扩展性、可靠性和性能。系统的层次图如图4-1所示。

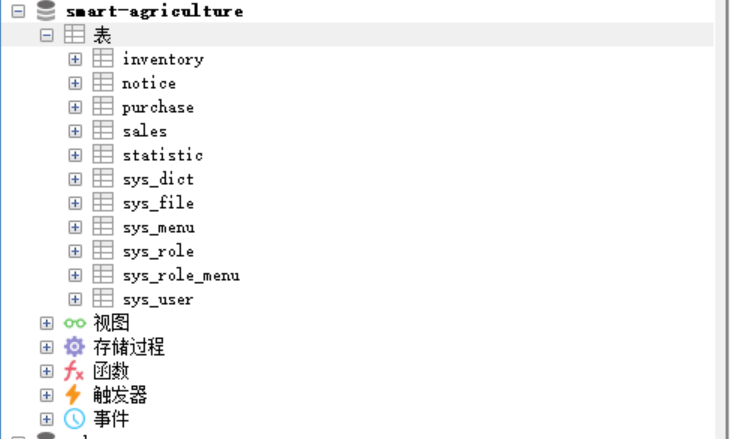


**图4-1 系统层次图**

4.2数据库设计

**4.2.1创建数据库**

智慧农田管理平台使用MySQL作为数据库管理系统，MySQL以其高性能、稳定性、支持多种操作系统和编程语言接口而受到青睐，适用于Web应用程序、企业应用程序、移动应用程序等多种场合。MySQL支持多种存储引擎，如InnoDB和MyISAM，其中InnoDB是默认的存储引擎。如图4-1结构所示。



**图4-1 数据表结构建立图**

**4.2.2数据表结构**

1. 用户表

用户表用于存储管理员和农场员工的用户信息和账户信息，这些信息包括但用户ID、用户名、密码、邮箱、手机号码、用户头像、用户角色、注册时间，该表结构如4-1所示。

**表4-1 表sys\_user**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| Username | Varchar | 255 |  |  |
| Password | Varchar | 255 |  |  |
| Nickname | Varchar | 255 |  |  |
| Email | Varchar | 255 |  |  |
| Phone | Varchar | 255 |  |  |
| Address | varchar | 255 |  |  |
| Create\_time | timestamp |  |  |  |
| Create\_url | varchar | 255 |  |  |
| role | varchar | 50 |  |  |

1. 角色表

角色表拥有存储系统角色的表，包括角色id、系统人员名称、角色描述、角色英文名称。该表结构如4-2所示。

**表4-2 表sys\_role**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| Name | Varchar | 255 |  |  |
| description | varchar | 255 |  |  |
| flag | varchar | 255 |  |  |

1. 角色和用户的中间表

角色和用户的中间表，主要用于存储角色和菜单的对应关系，包括角色id、菜单id。该表结构如4-3所示。

**表4-3表sys\_role\_menu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Role\_id | Int | 20 | 否 |  |
| Menu\_id | Int | 20 | 否 |  |

1. 菜单表

菜单表用于存储菜单信息，包括菜单id、菜单名称、菜单路径、菜单图标、菜单秒、pid、sort\_num。该表结构如4-4所示。

**表4-4 sys\_menu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为null | 键 |
| Id | Int | 20 | 否 | 主键 |
| name | Varchar | 255 |  |  |
| path | Varchar | 255 |  |  |
| icon | Varchar | 255 |  |  |
| description | Varchar | 255 |  |  |
| pid | int | 20 |  |  |
| Page\_path | varchar | 255 |  |  |
| sort\_num | int | 11 |  |  |

1. 文件表

文件表用于存储用户头像文件和系统文件，字段包括：文件id、文件名称、文件类型、微距大小、文件url、文件md5、是否可删除、是否禁用链接。如表结构如4-5所示。

**表4-5 sys\_file**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| name | Varchar | 255 |  |  |
| type | Varchar | 255 |  |  |
| size | bigint | 20 |  |  |
| url | Varchar | 255 |  |  |
| Md5 | varchar | 255 |  |  |
| Is\_delete | tinyint | 1 |  |  |
| enable | tinyint | 1 |  |  |

1. 系统图标表

系统图标表，用于存储系统菜单图标。字段包括：图标id、图标名称、图标value、图标类型。如表结构如4-6所示。

**表4-6 sys\_dict**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为null | 键 |
| Id | Int | 20 | 否 | 主键 |
| name | Varchar | 255 |  |  |
| value | varchar | 255 |  |  |
| type | varchar | 255 |  |  |

1. 农田表

农田表，用于管理和存储农田信息的表，字段包括：id、农田名称、面积、地址、农作物、数量、生长状态、温度、突然湿度等字段。其表结构如4-7所示。

**表4-7 statistic**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| farm | Varchar | 255 |  |  |
| area | Varchar | 255 |  |  |
| addresss | Varchar | 255 |  |  |
| crop | Varchar | 255 |  |  |
| number | int | 11 |  |  |
| state | varchar | 255 |  |  |
| temperature | decimal | 11,1 |  |  |
| airhumidity | int | 11 |  |  |
| soilhumidity | int | 11 |  |  |
| carbon | int | 11 |  |  |
| ph | decimal | 11,2 |  |  |
| light | int | 11 |  |  |
| filllight | varchar | 255 |  |  |
| monitor | varchar | 255 |  |  |
| pump | varchar | 255 |  |  |
| keeper | varchar | 255 |  |  |

1. 销售表

销售表用于记录农作物销售记录，字段包括：id、产品名称、价格、采购数量、采购商、地址、电话、出货人、备注等字段。其表结构如4-8所示。

**表4-8 sales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| prodeuct | Varchar | 255 |  |  |
| price | decimal | 10,2 |  |  |
| number | int | 11 |  |  |
| buyer | varchar | 255 |  |  |
| address | varchar | 255 |  |  |
| phone | varchar | 255 |  |  |
| shipper | varchar | 255 |  |  |
| remark | varchar | 255 |  |  |

1. 采购表

采购表负责农资采购记录，其字段包括：id、采购产品、价格、采购数量、供应商、采购人、备注等字段。其表结构如4-9所示。

**表4-9 purchase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| product | Varchar | 255 |  |  |
| price | decimal | 10,2 |  |  |
| number | int | 11 |  |  |
| provider | varchar | 255 |  |  |
| address | varchar | 255 |  |  |
| phone | varchar | 255 |  |  |
| purchaser | varchar | 255 |  |  |
| remark | varchar | 255 |  |  |

1. 系统公告表

系统公告表，存储农田管理平台的系统公告，其字段包括：id、标题、内容、发布时间、发布人、封面等字段。其表结构如4-10所示。

**表4-10 notice**

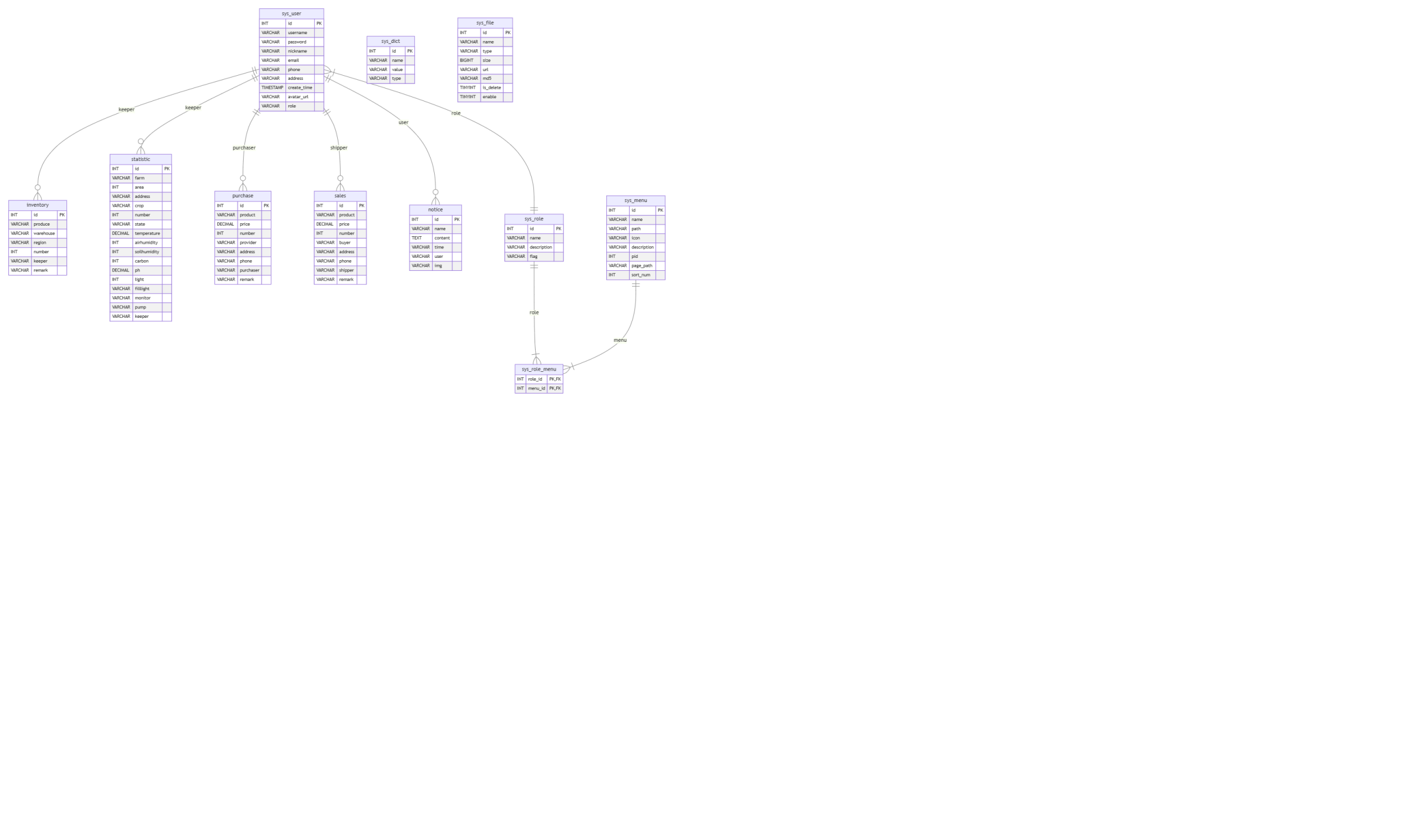
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| name | Varchar | 255 |  |  |
| content | text |  |  |  |
| time | varchar | 255 |  |  |
| user | varchar | 255 |  |  |
| img | varchar |  |  |  |

1. 仓库表

仓库表，用于存放仓库产品的表，其字段包括：id、产品、仓库、存储区、数量、仓库管理员、备注等字段。其表结构如4-11所示。

**表4-11 inventory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 是否为Null | 键 |
| Id | Int | 20 | 否 | 主键 |
| produce | Varchar | 255 |  |  |
| warehouse | Varchar | 255 |  |  |
| region | varchar | 255 |  |  |
| number | int | 11 |  |  |
| keeper | varchar | 255 |  |  |
| remark | varchar | 255 |  |  |

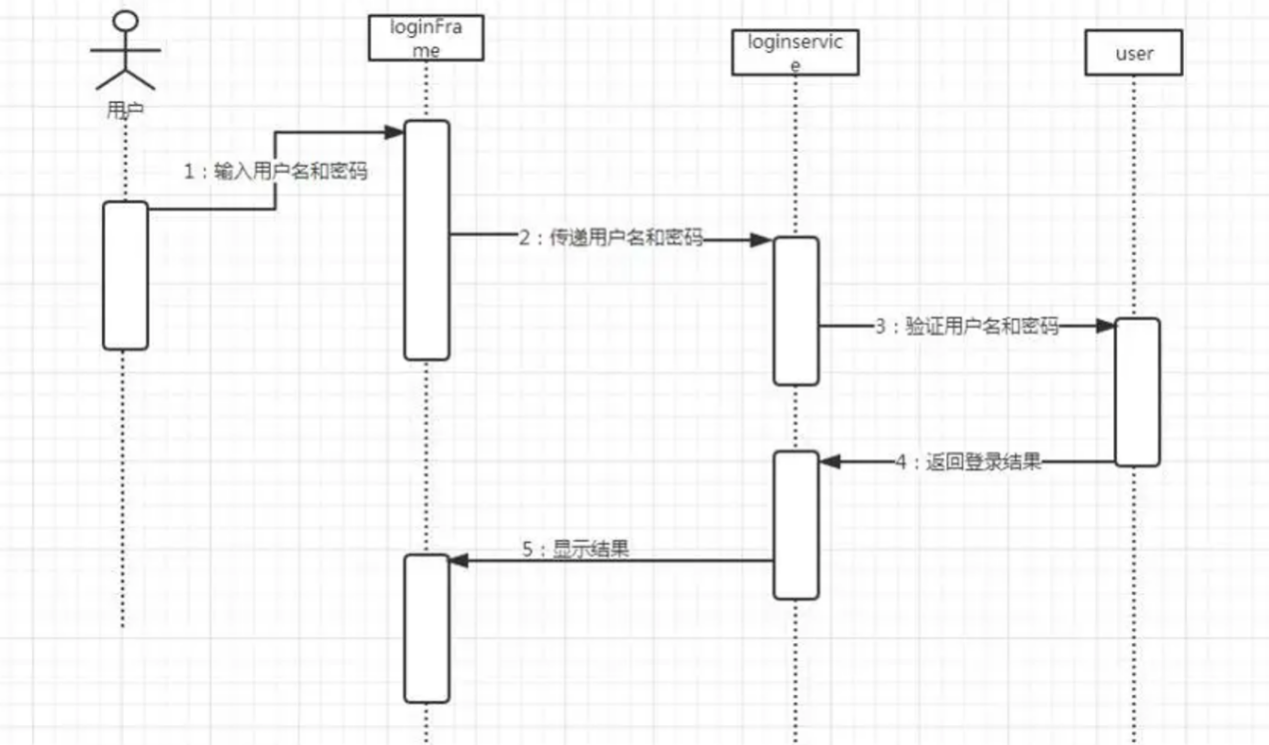


4.3界面设计

**4.3.1普通员工界面设计**

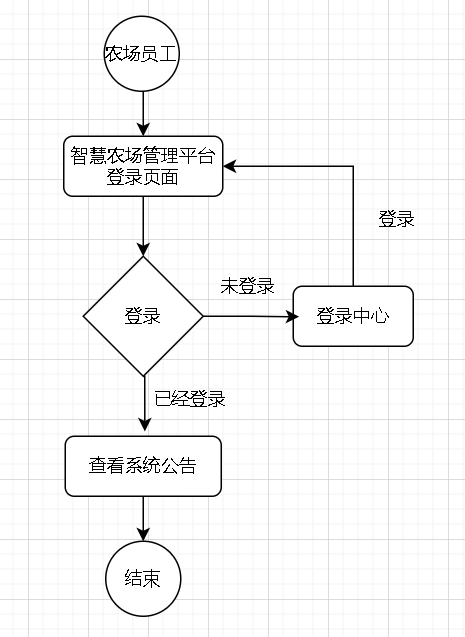
普通员工可以进行多功能的操作。农场员工登录后，可以对农场主页系统公告通知、农场统计分析、农田管理、农作物资管理等模块管理等操作。

登录活动图表示学生进行登录时的信息。如图4-2所示。



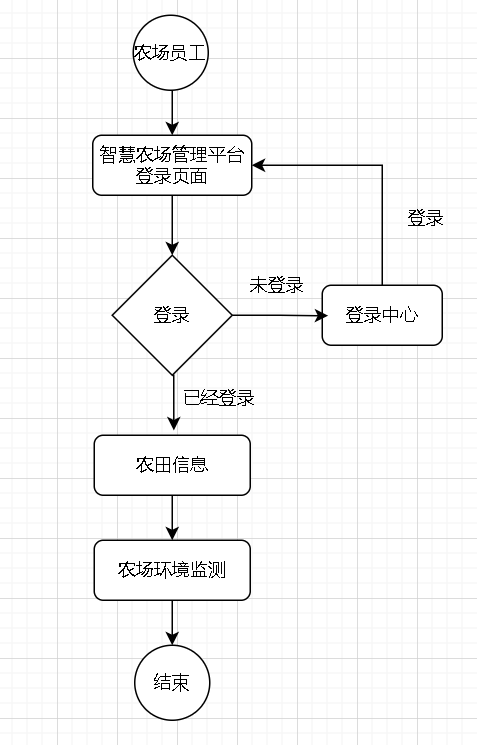
**图4-2 登录活动图**

农场系统公告活动图表示员工查看农场系统公告的操作。如图4-3所示。



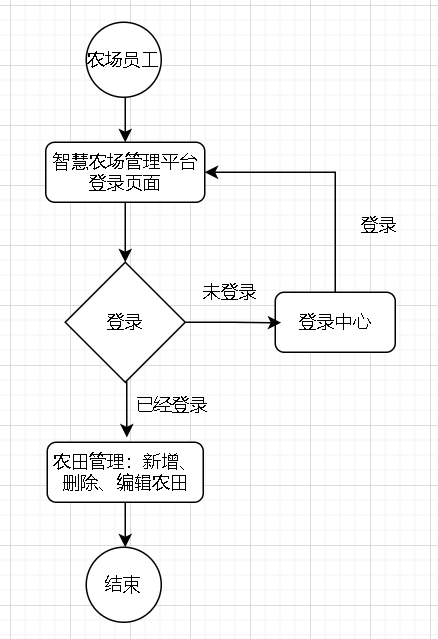
**图4-3 系统公告活动图**

农场统计分析活动图：包括农场环境监测和农田信息管理。如图4-4所示。

****

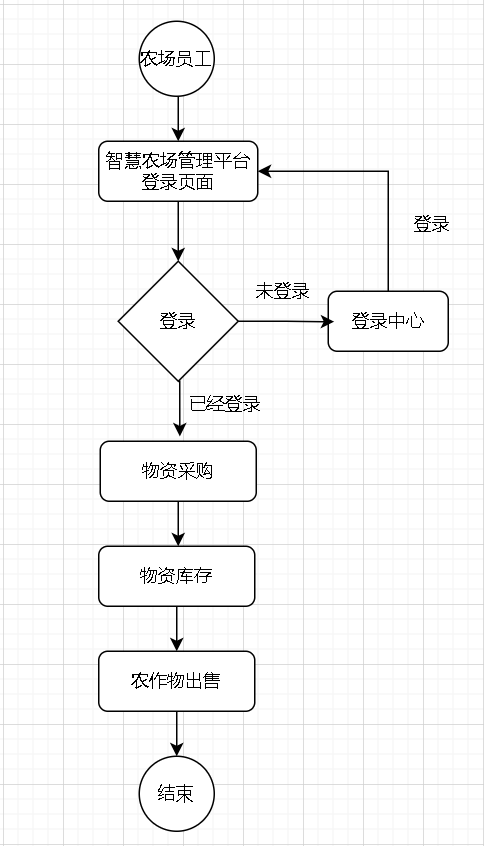
**图4-4农场统计分析活动图**

农田管理活动图：包括农场环境监测和农田信息管理。如图4-5所示。

****

**图4-5农田管理活动图**

农场物资管理活动图：包括物资采购、物资库存、农作物出售等操作。如图4-6所示。

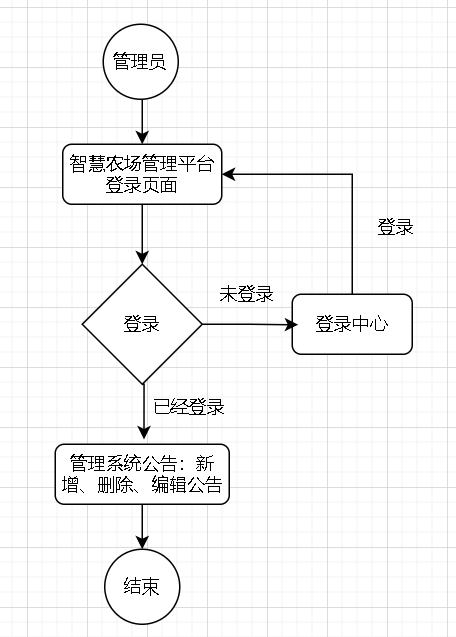
****

**图4-6农场物资管理活动图**

**4.3.2管理员界面设计**

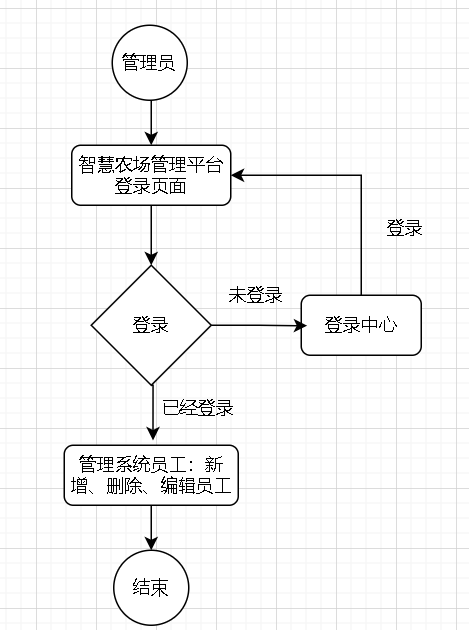
农场管理员可以：操作系统所以模块包括：农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理。管理员除了拥有普通员工拥有的功能，还有：系统公告管理、农场员工管理、系统角色管理、系统菜单管理等模块。

管理员管理系统公告管理活动图，如图4-7



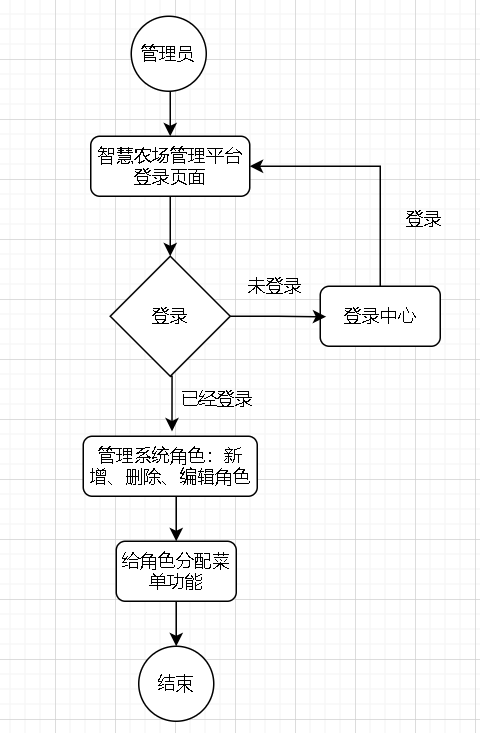
**图4-7管理系统公告活动图**

管理员管理农场员工管理活动图，如图4-8



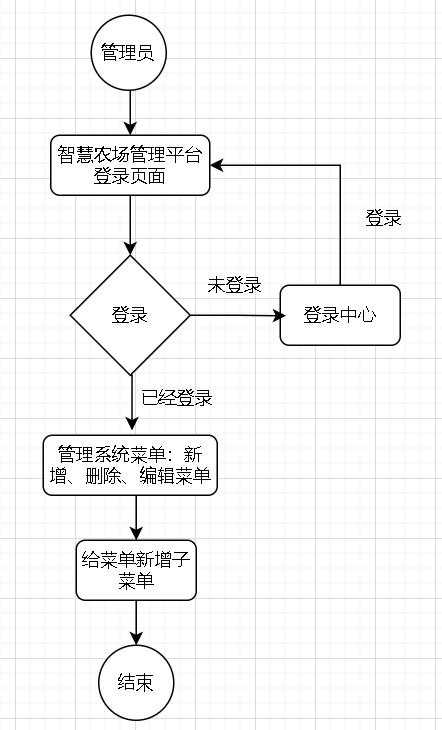
**图4-8管理农场员工活动图**

管理员管理系统角色管理活动图，如图4-9



**图4-9管理角色活动图**

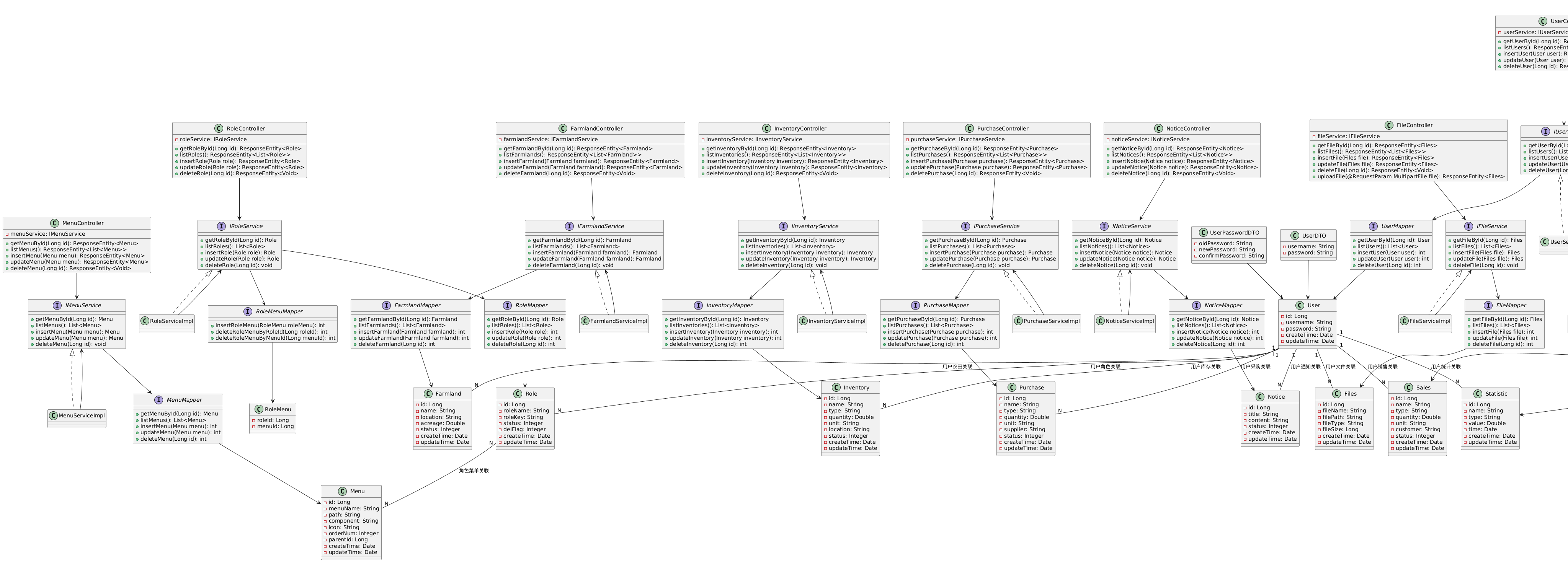
管理员管理系统菜单管理活动图，如图4-10

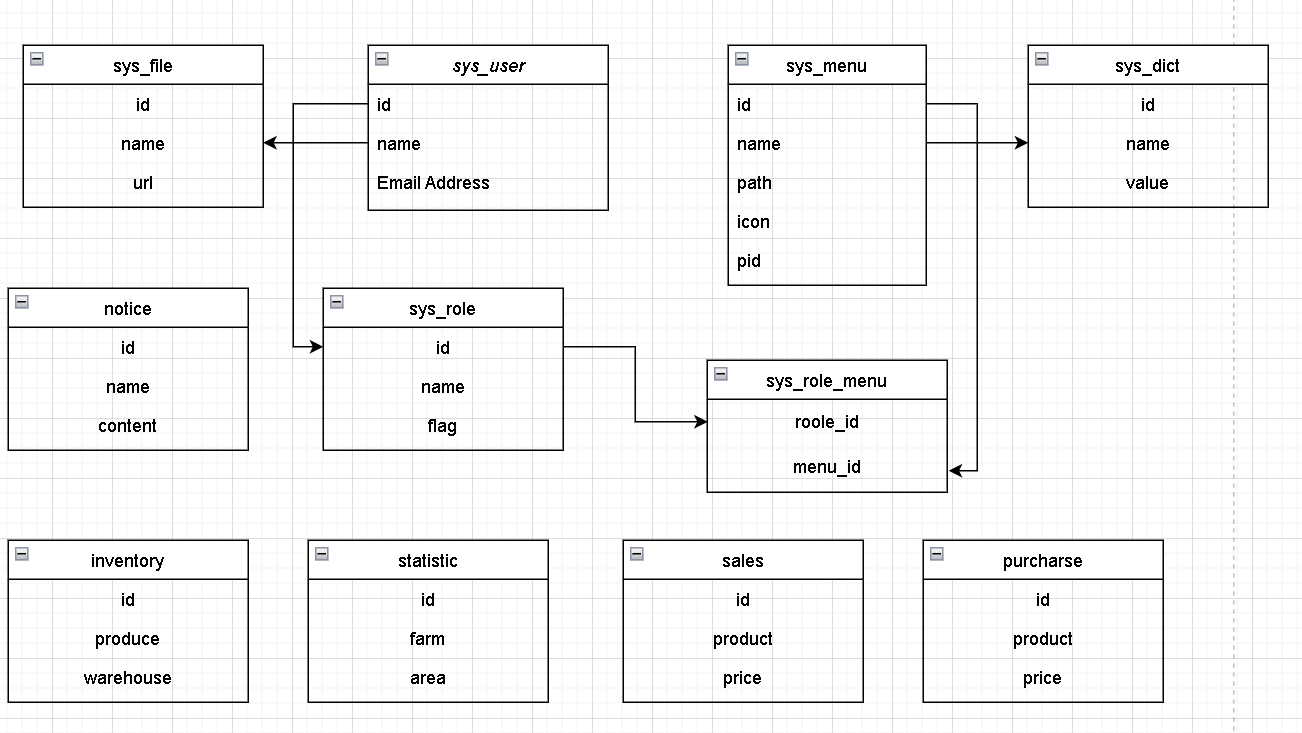


**图4-10管理菜单活动图**

4.4系统流程设计

智慧农田管理平台系统之间各模块有着各自对应的关系，其关系可能是一对一，一对多，多对一，也存在其他的关系，关系如图4-11所示。

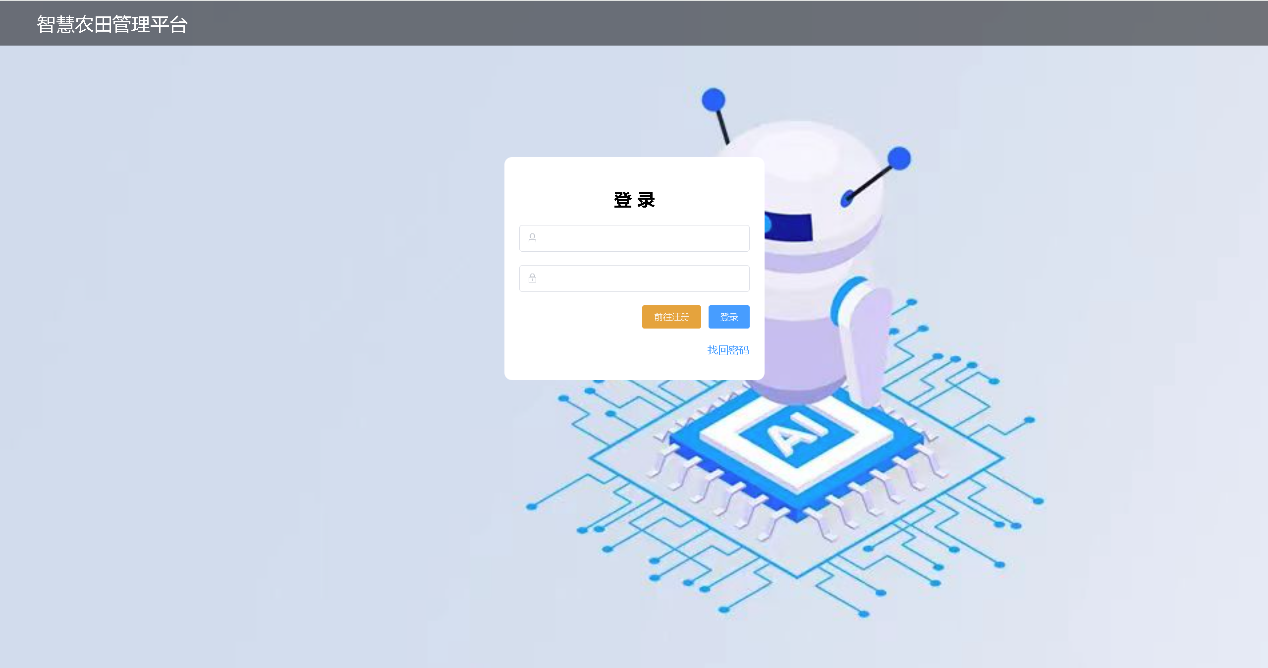




**图4-12 农田管理平台系统类图**

4.5功能模块设计

在使用本系统之前必须进行登录操作，登录之后，不同的角色可以查看不同的页面和权限。如图4-13所示。



**图4-13 登录页面图**

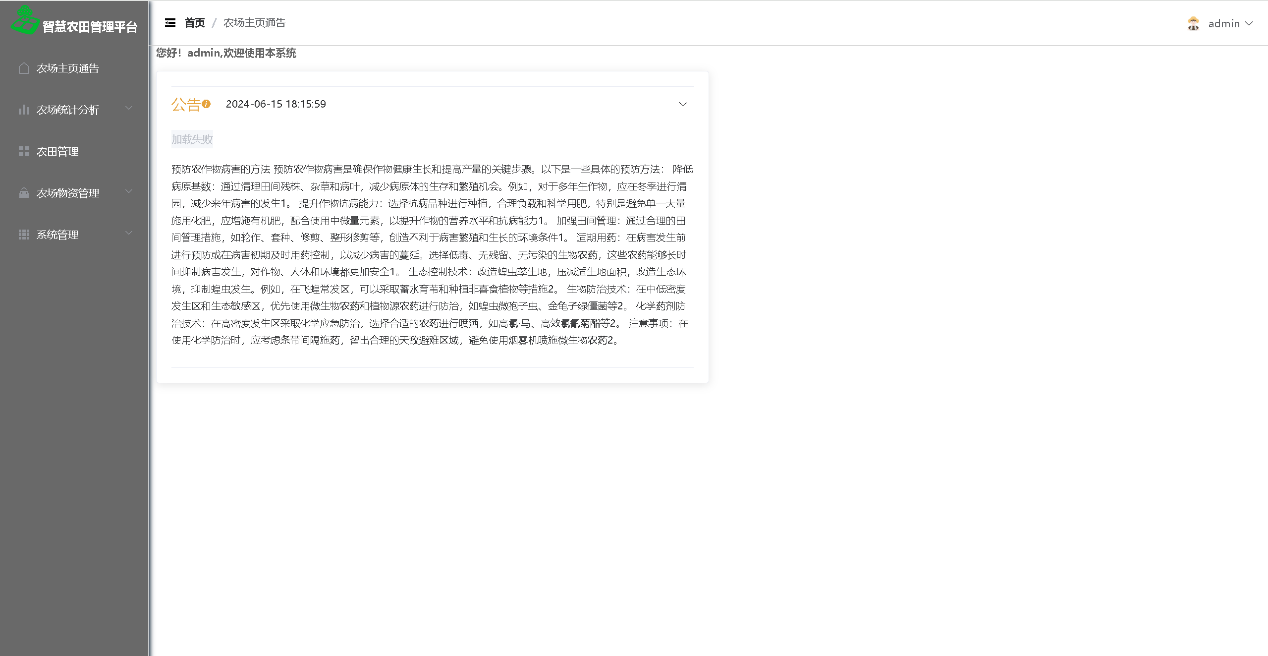
本系统的的主要功能模块分为：智慧农田管理平台：主要分为农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块。整个过程参与有管理员和农场员工两种用户类型。按照功能和权限进行角色划分，可划分为：

管理员：管理系统中所有的功能，可以自行定义系统。

农场员工：农场员工，可以划分为为采购人员、销售人员、农田管理人员。各模块的具体功能实现如下：

**4.5.1系统公告**

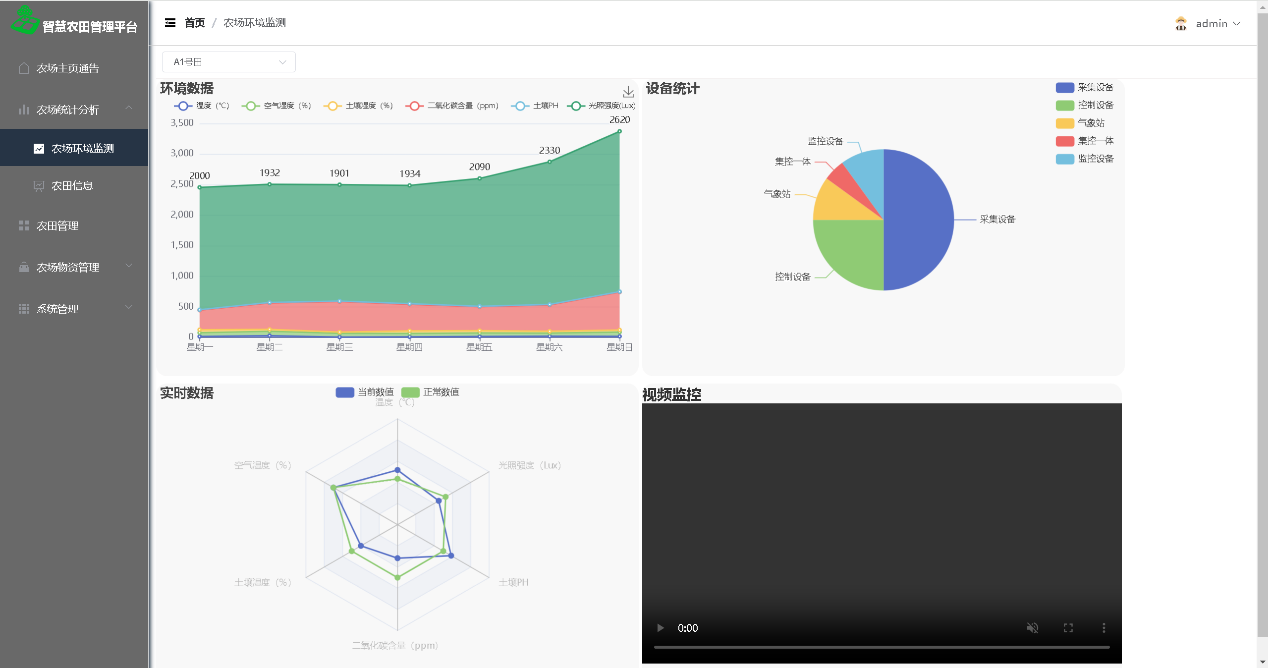
当管理员或者普通员工登录智慧农田管理平台后，首先进入的是系统公告页，系统公告主要作用是提醒农场管理员或者员工天气情况和害虫防御要求。如图4-3所示。



**图4-14系统公告页面**

**4.5.2农场环境监测**

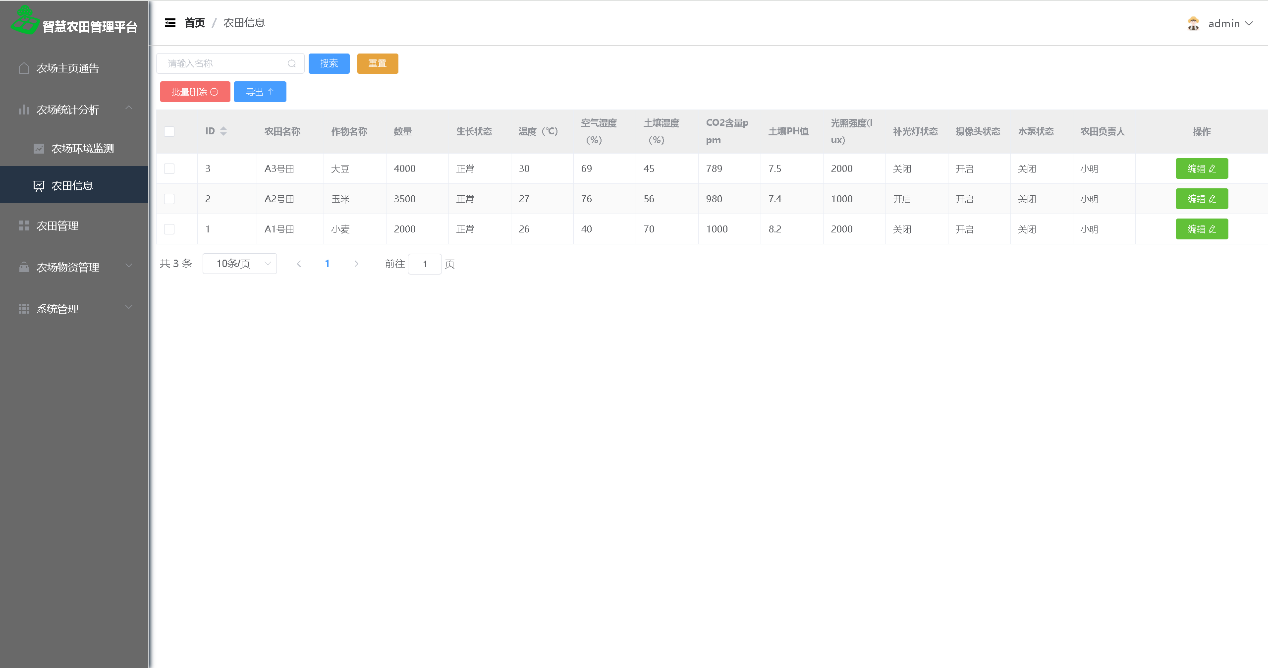
农场环境检测：可以分别查看A1、A2、A3等农田的环境信息和视频实时监控画面。如图4-15所示。



**图4-14 环境检测图**

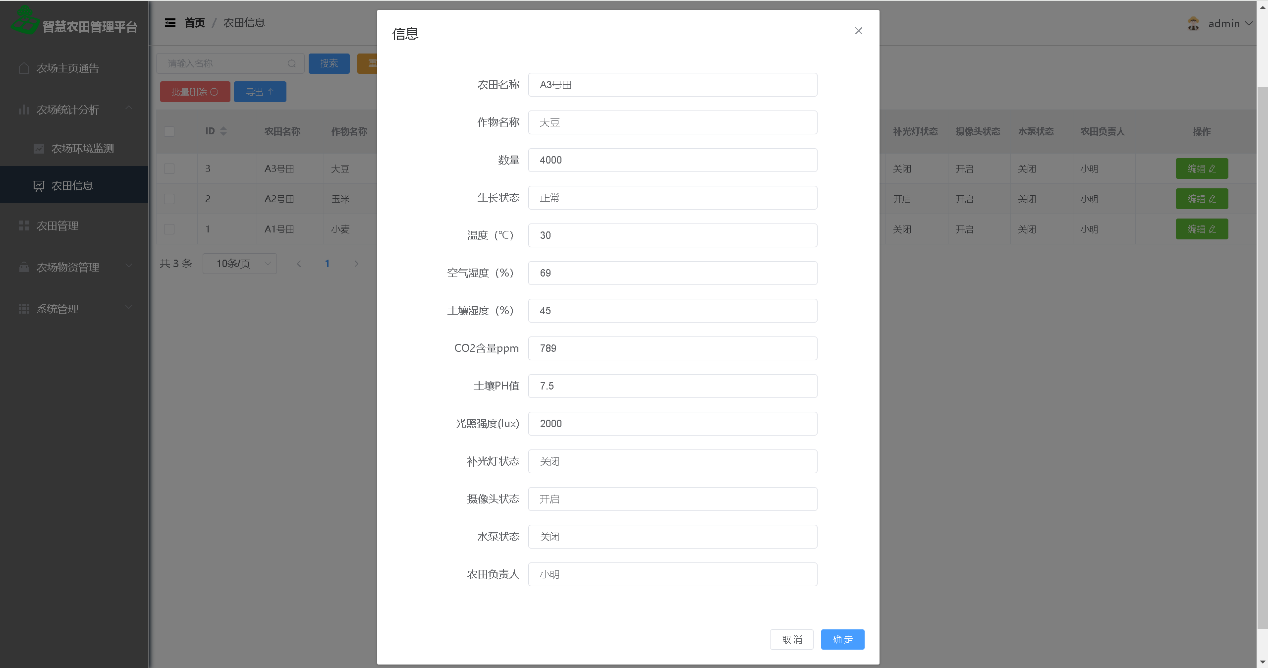
**4.5.3农田信息**

农田信息：可以编辑农田、批量删除农田，导出农田信息到excel表格中。如图4-15所示。



**图4-15农田信息图**

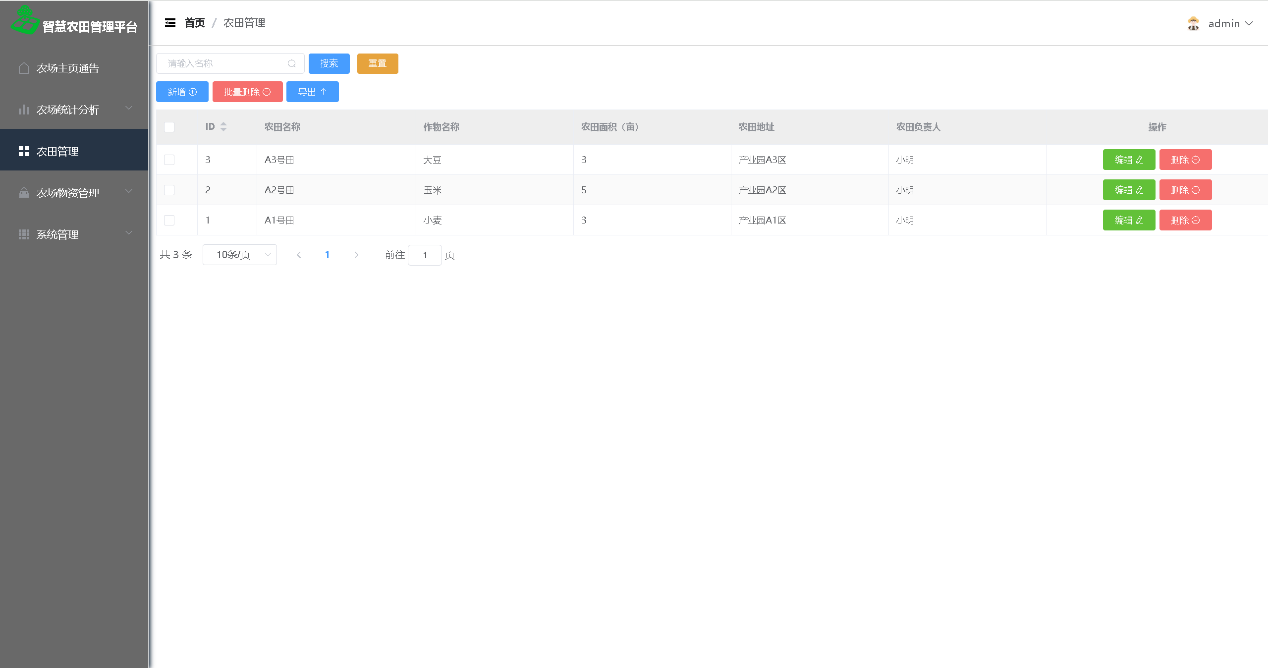
编辑农田，可以编辑农田的基本信息。如图4-16所示。



**图4-16编辑农田图**

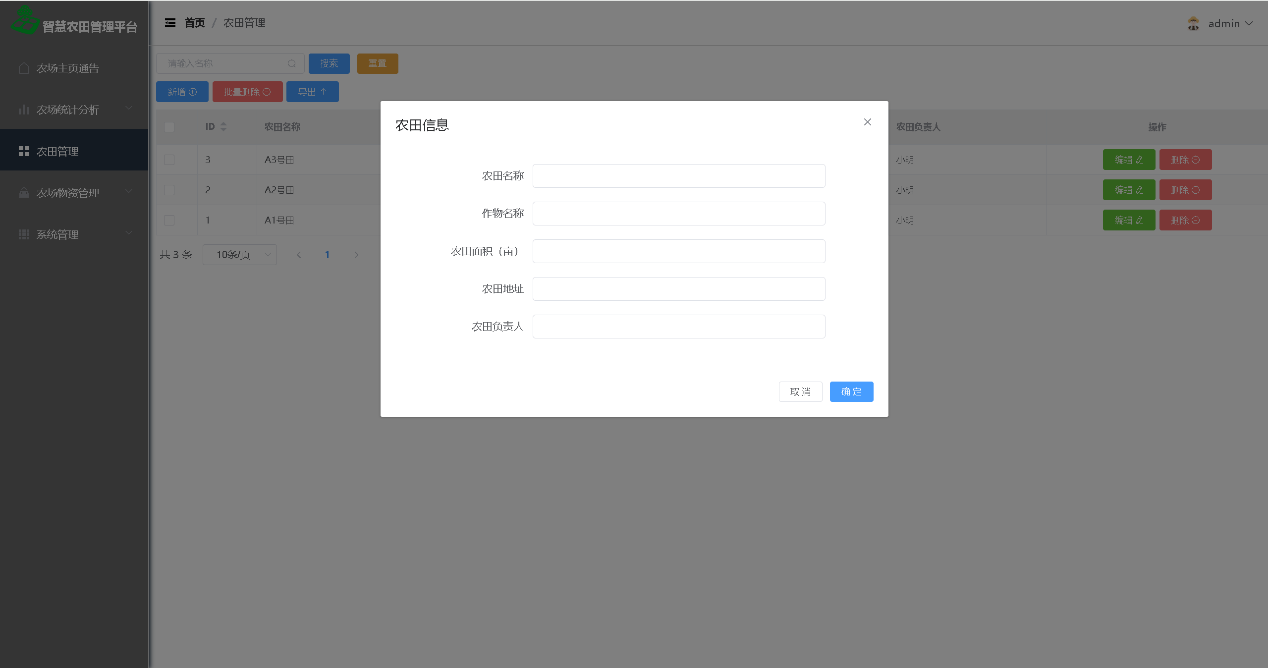
**4.5.4 农田管理**

农田管理：可以新增农田、导出农田、批量删除农田和编辑农田。如图4-17所示。

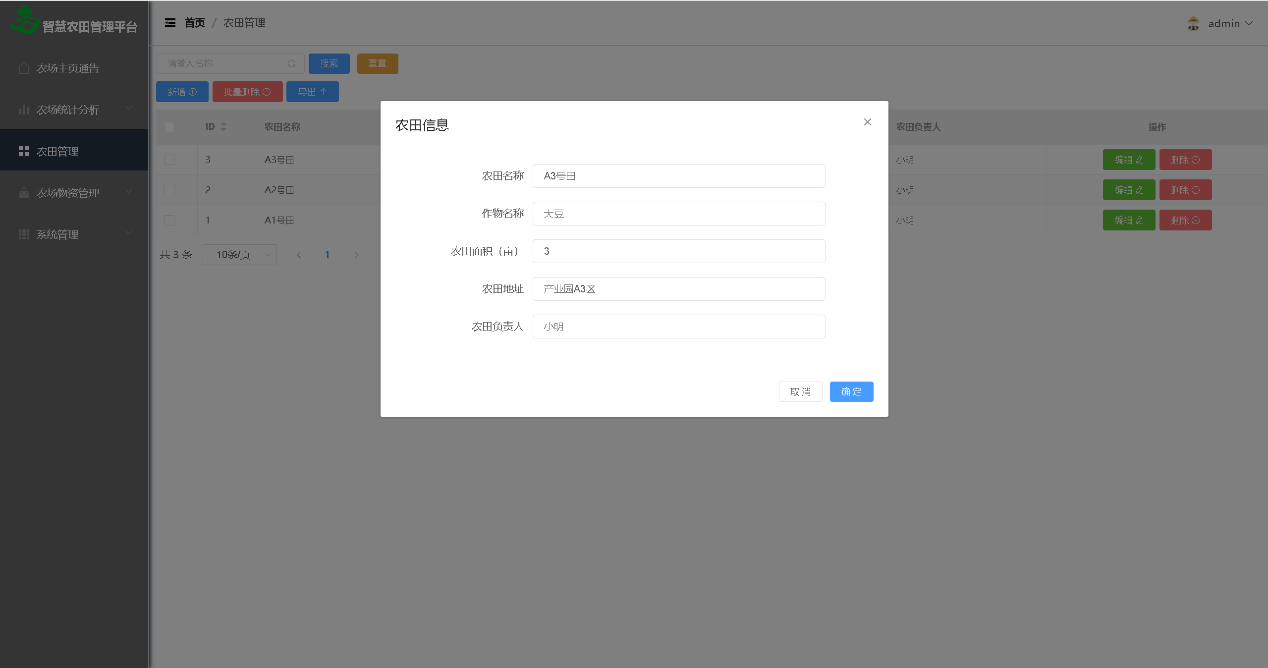


**图4-17农田管理图**

新增农田：添加农田的基本信息之后点击新增。如图4-18、4-19所示。



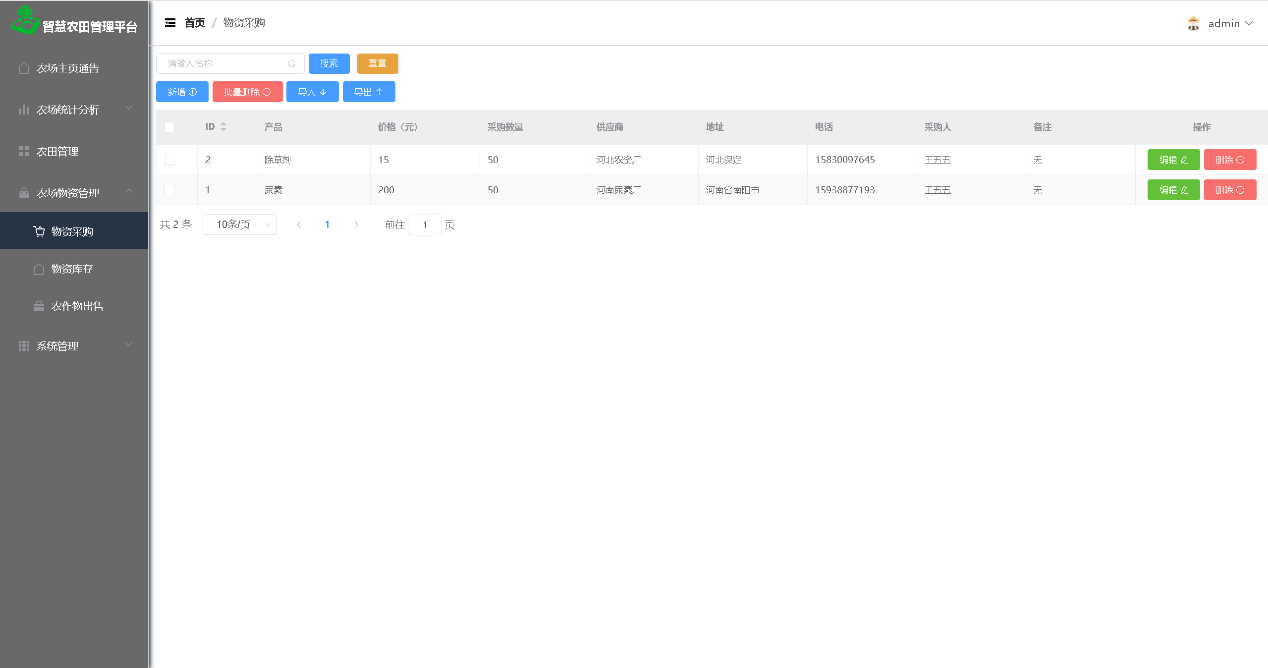
**图4-18 新增农田**

****

**图4-19编辑农田农田**

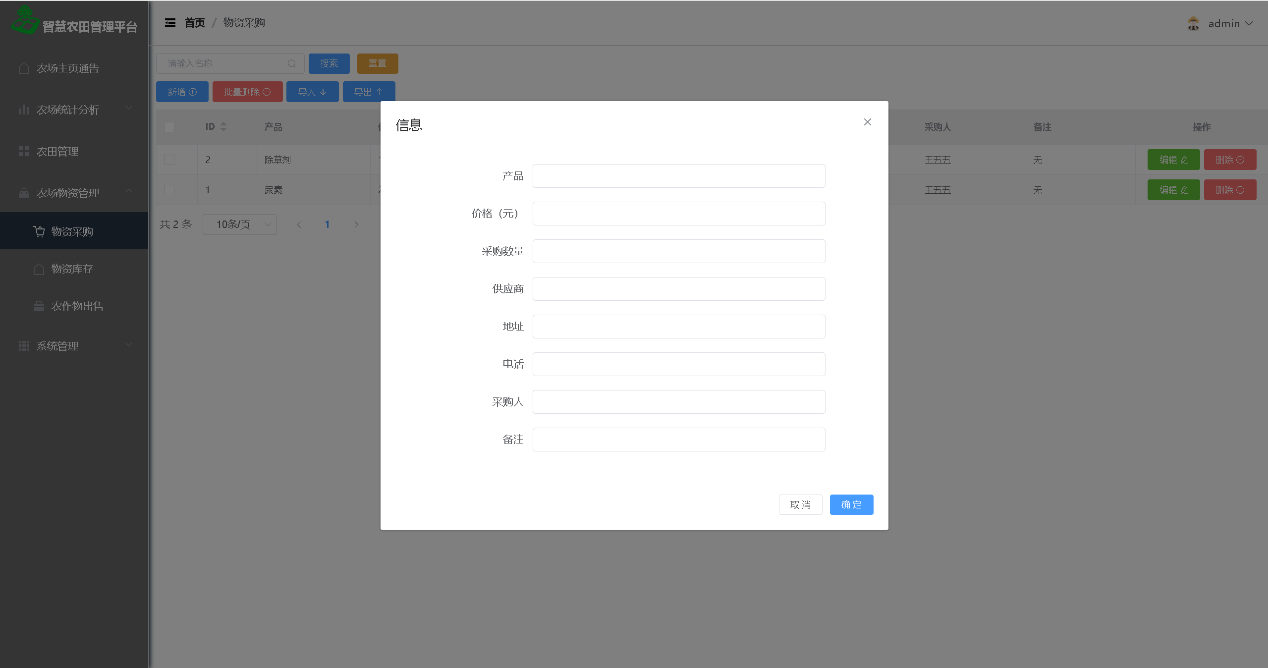
**4.5.5物资采购**

物资采购：负责农作物资的采购记录，可以新增采购物资、编辑采购物资、导出或导入采购物资和删除采购物资。如图4-20所示。



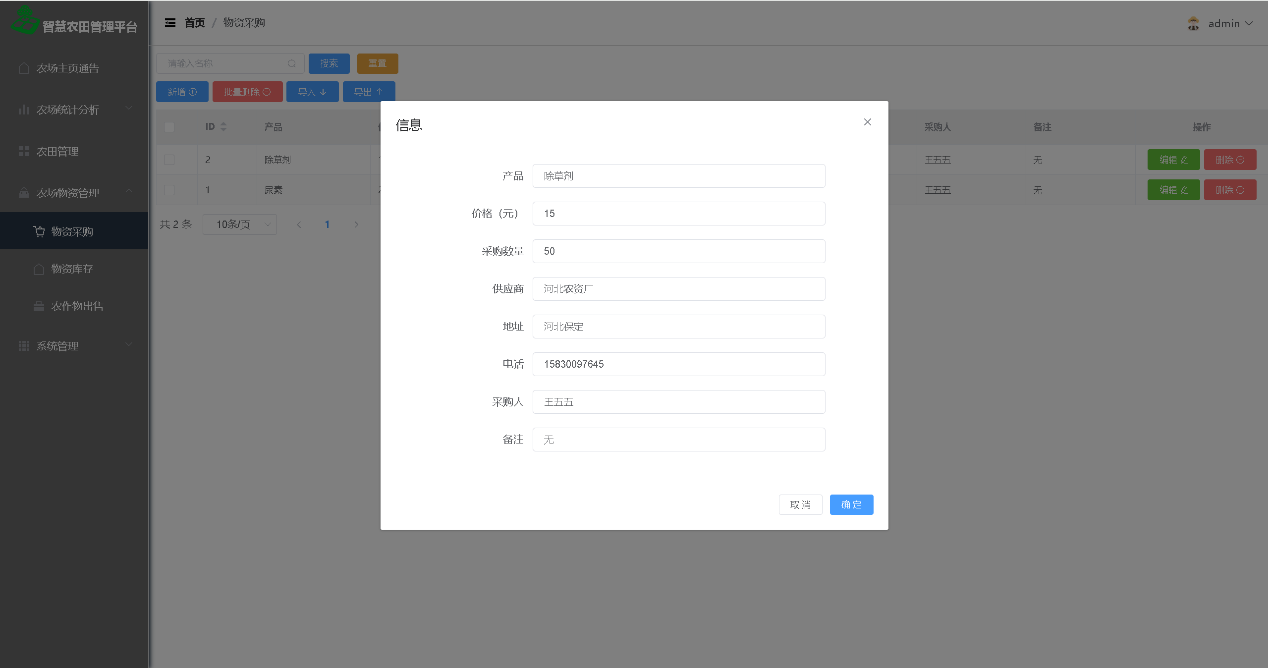
**图4-20 物资采购图**

新增采购物资操作。如图4-21所示。



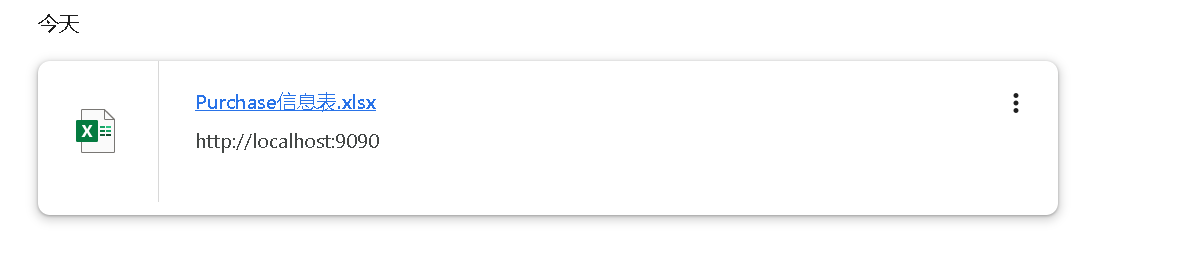
**图4-21 新增采购物资图**

编辑采购物资操作。如图4-22所示。



导出采购物资操作。如图4-22所示。

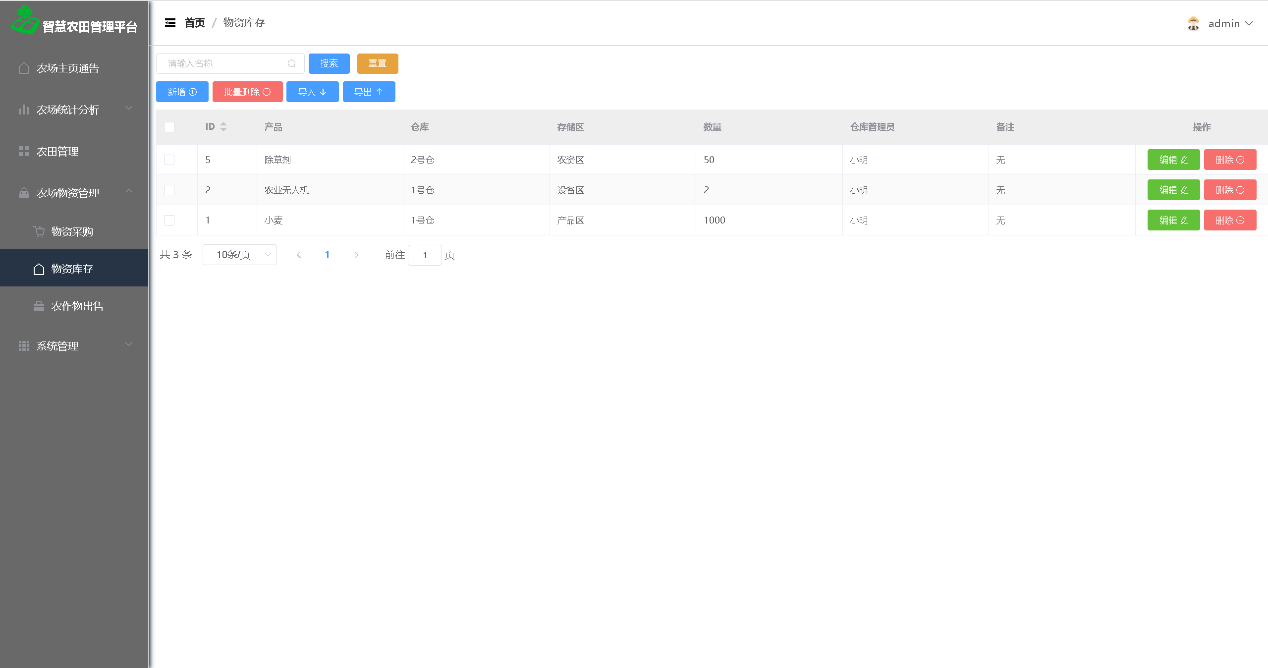
**图4-22 编辑采购物资图**

****

**图4-23 导出采购物资图**

**4.5.6物资库存**

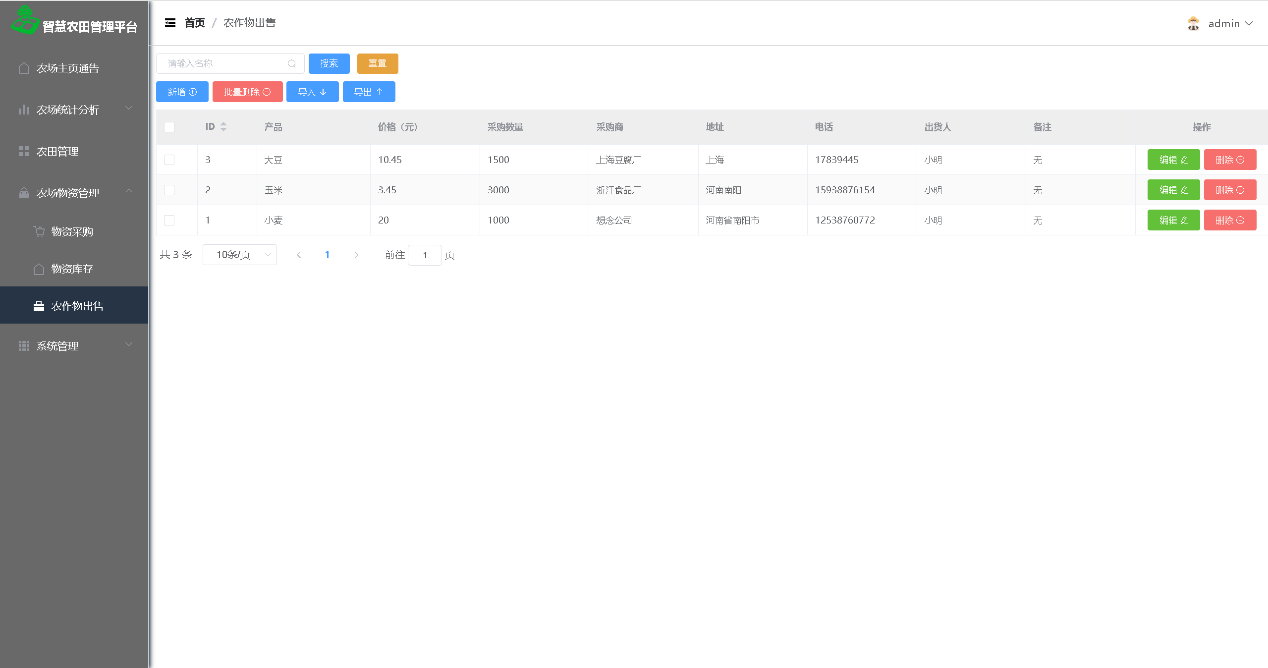
物资库存，用于记录农场目前所拥有的物资。如图4-24所示。



**图4-24 物资库存图**

**4.5.7农作物出售**

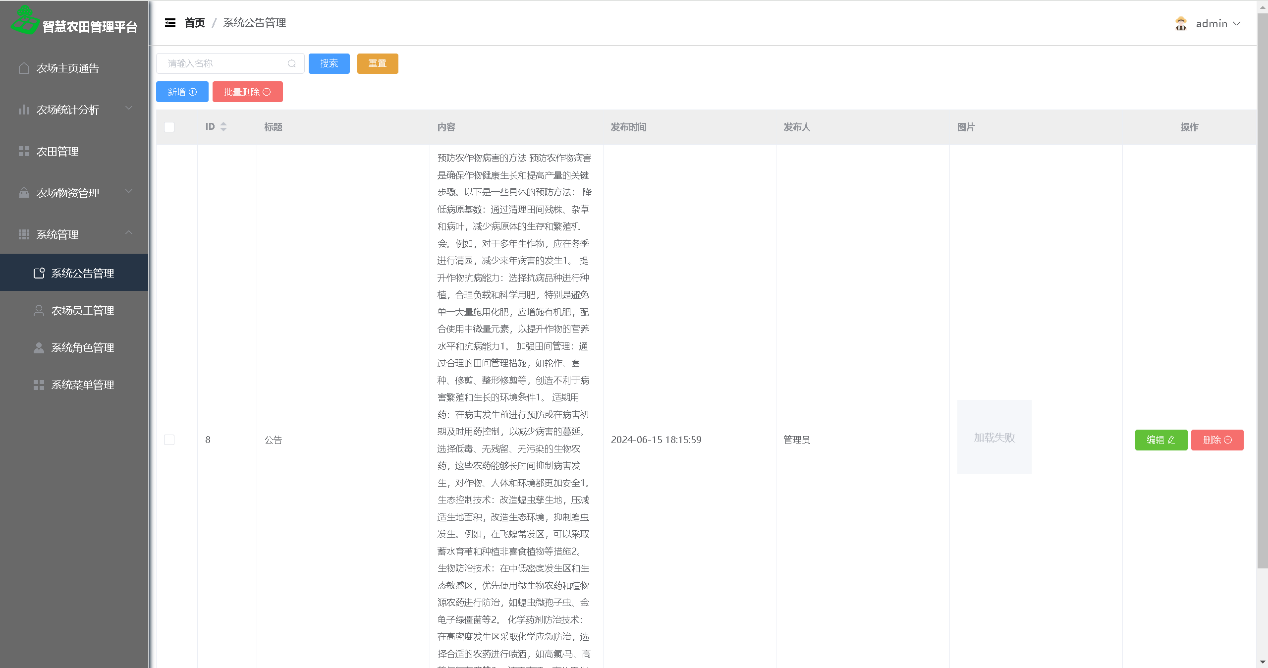
农作物出售，记录农场所出售的农作物。如图4-25所示。



**图4-25 农作物出售图**

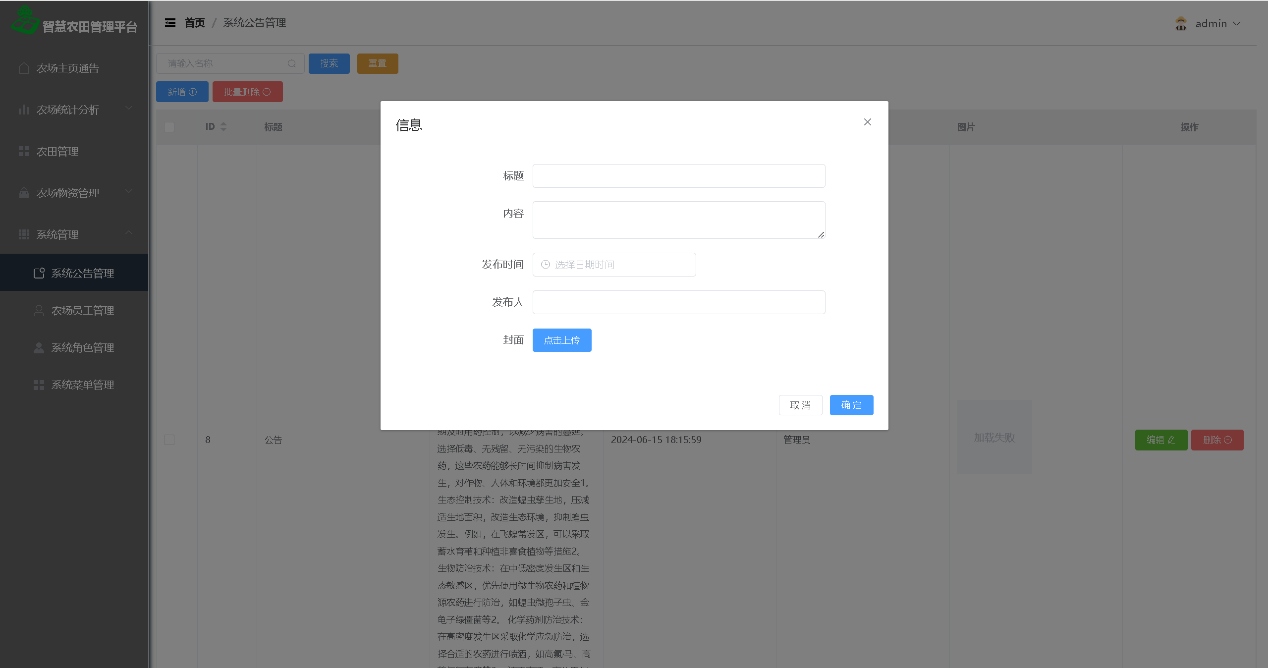
**4.5.8系统公告管理**

系统公告管理，负责编辑、新增、删除系统公告。如图4-26所示。



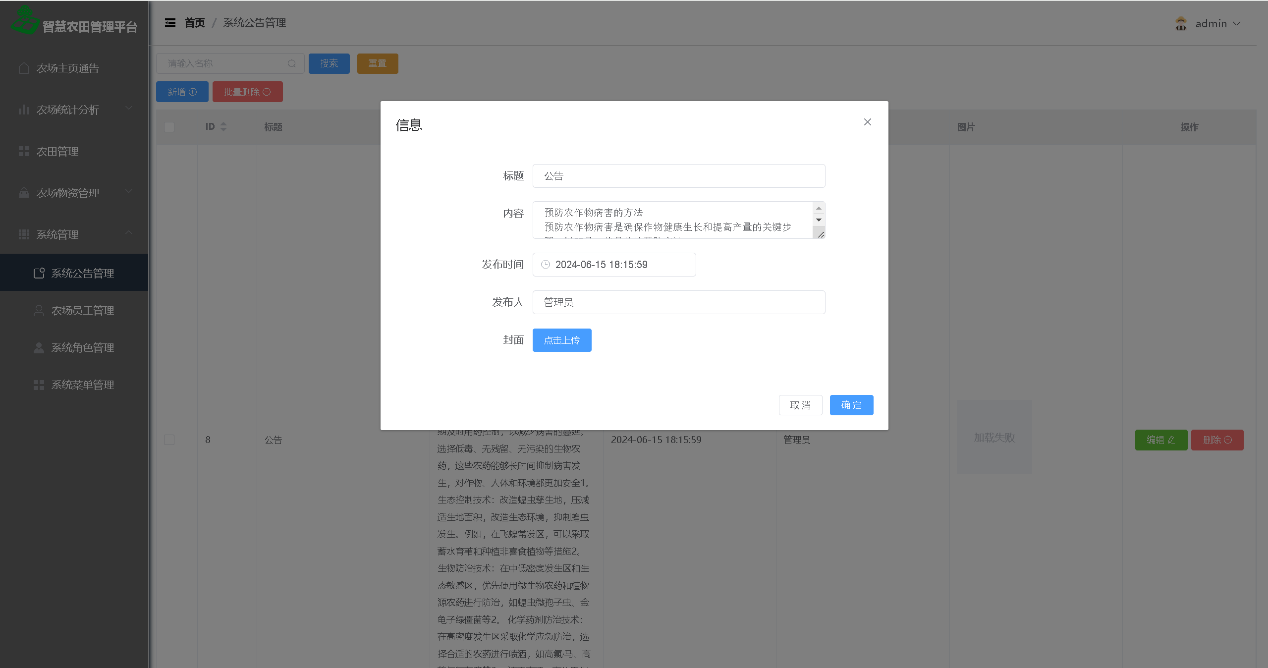
**图4-26 系统公告管理图**

新增系统公告。如图4-27所示。



**图4-27 新增系统公告图**

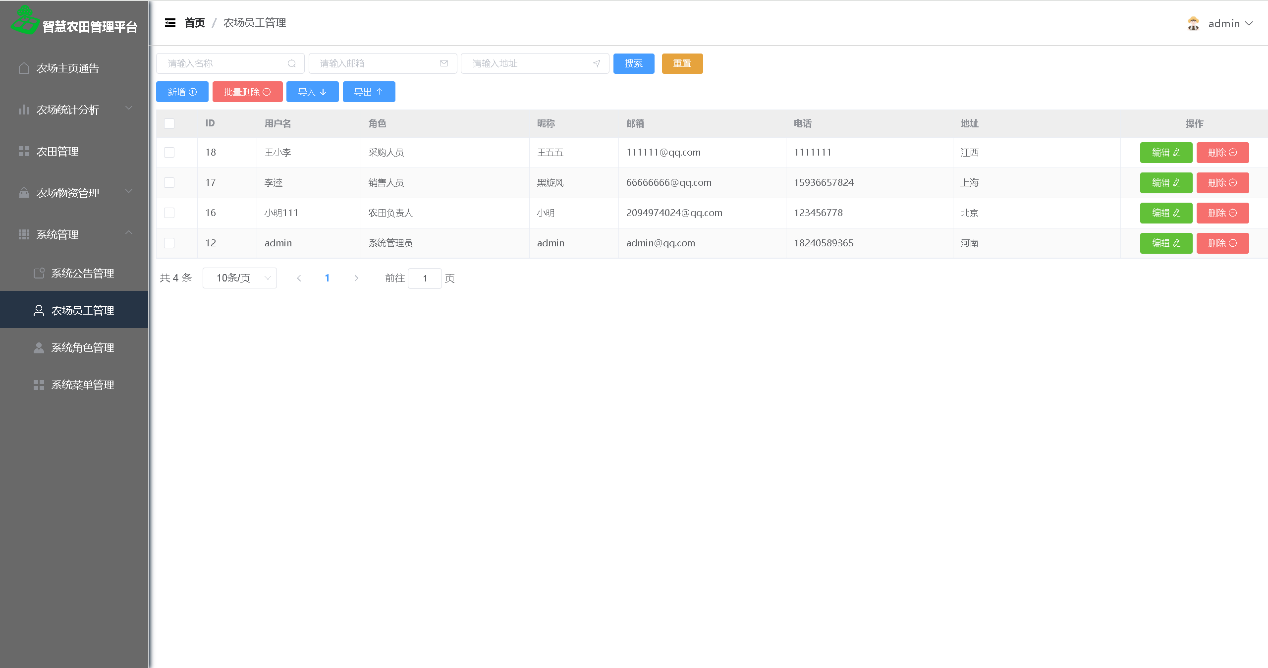
编辑系统公告。如图4-28所示。



**图4-28 编辑系统公告图**

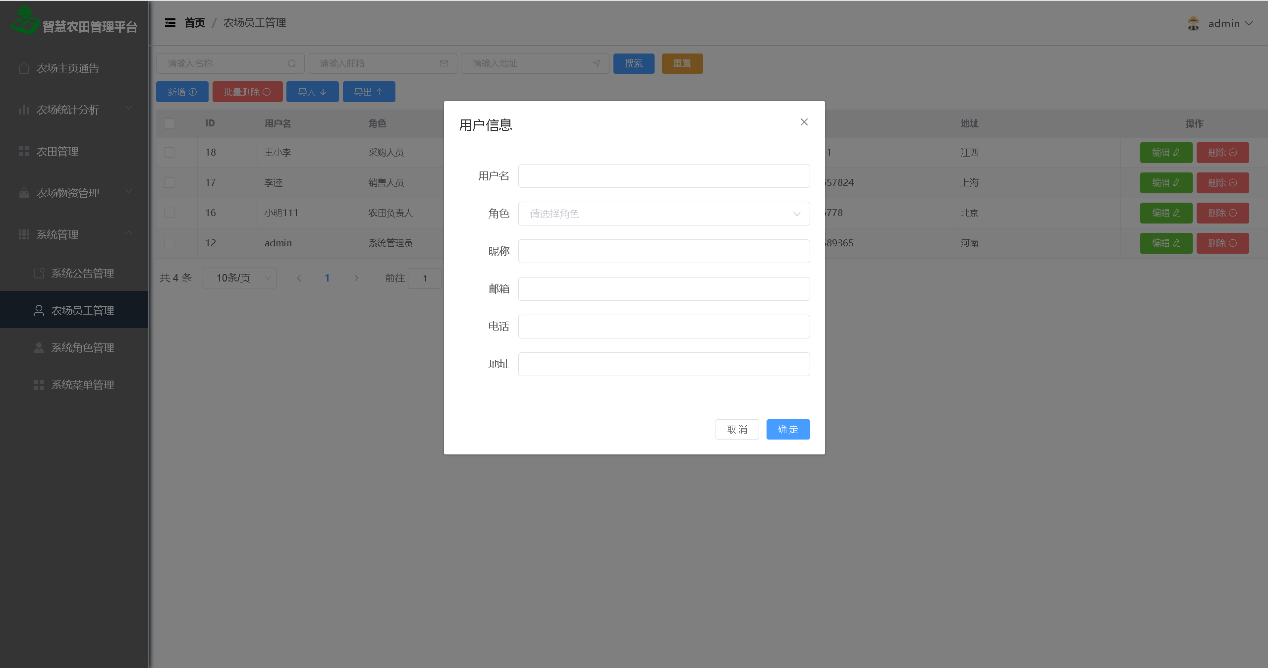
**4.5.9 农场员工管理**

农场员工管理：可以新增农场员工、导入或者导出员工、编辑和删除农场员工。如图4-29所示。



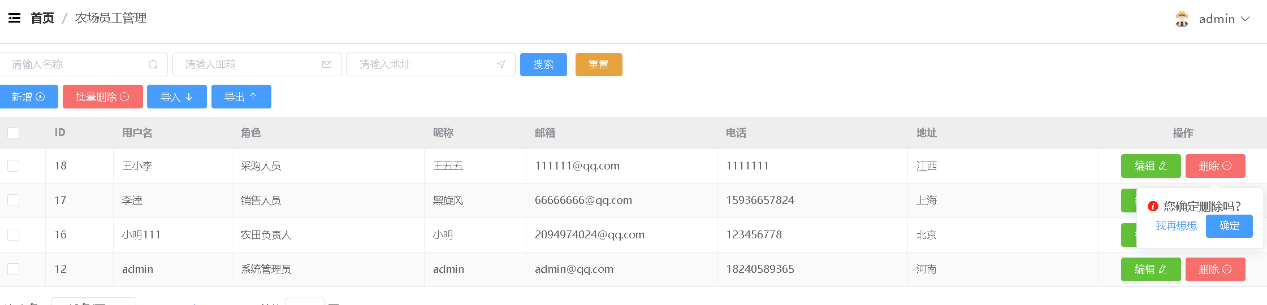
**图4-29 农场员工管理图**

管理员可以新增农场员工。如图4-30所示

****

**图4-30 新增农场员工图**

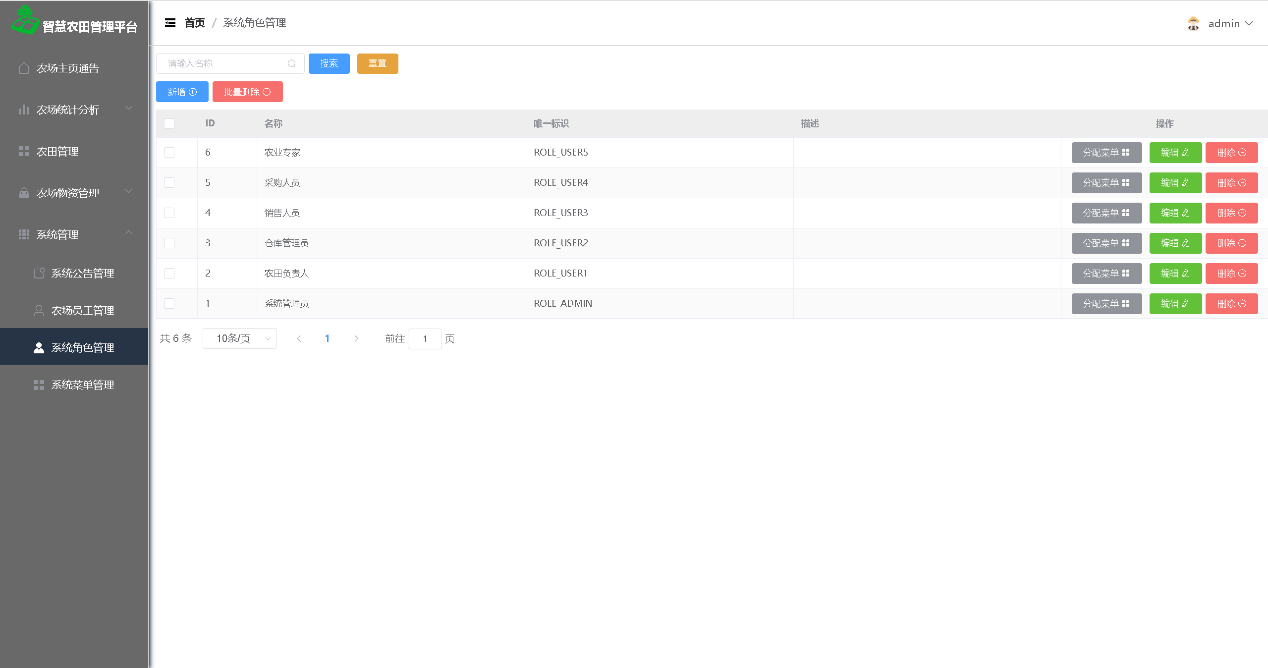
管理员可以自行删除员工。如图4-31所示

****

**图4-31 删除农场员工图**

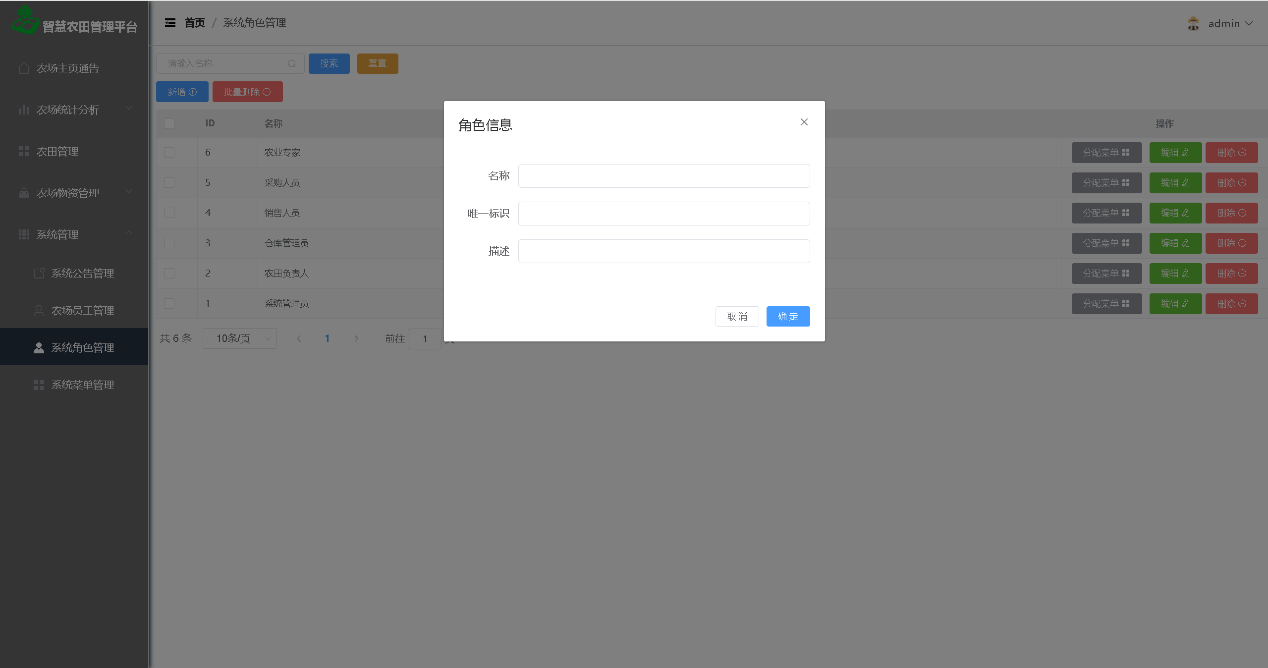
**4.5.10 系统角色管理**

系统角色管：由管理员分配系统角色，可以新增心态角色、编辑系统角色、删除系统角色和对新系统角色分配菜单功能。如图4-32所示



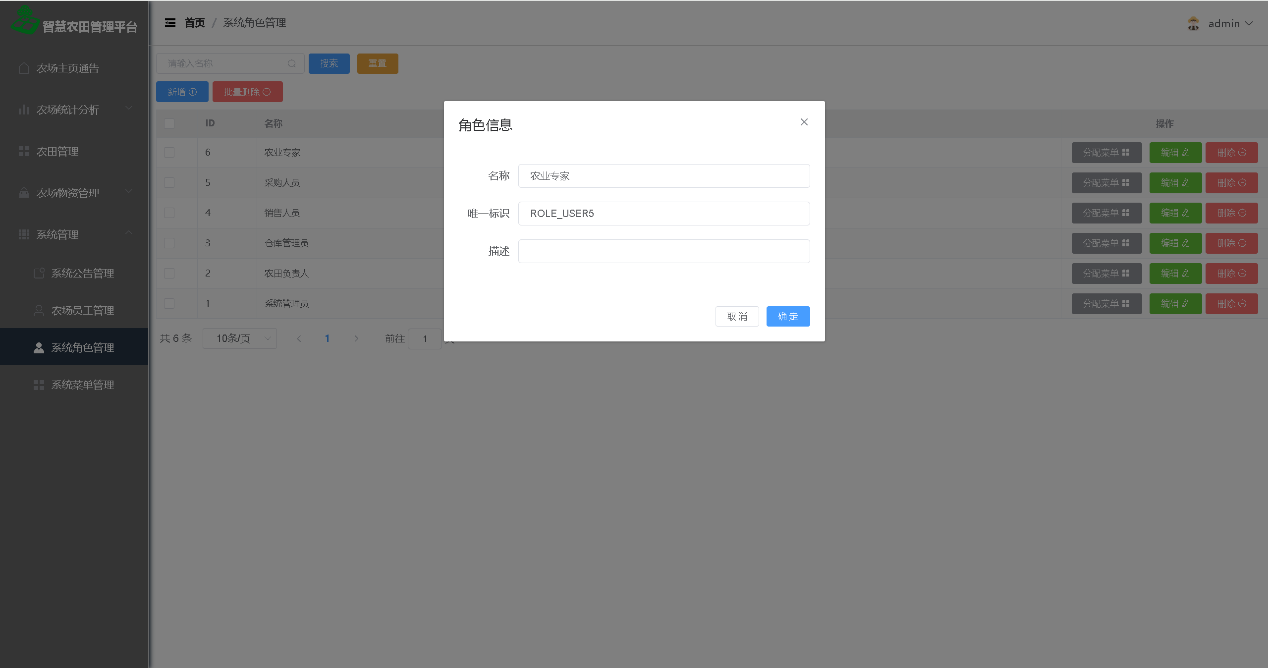
**图4-32 系统角色管理图**

新增系统角色。如图4-33所示



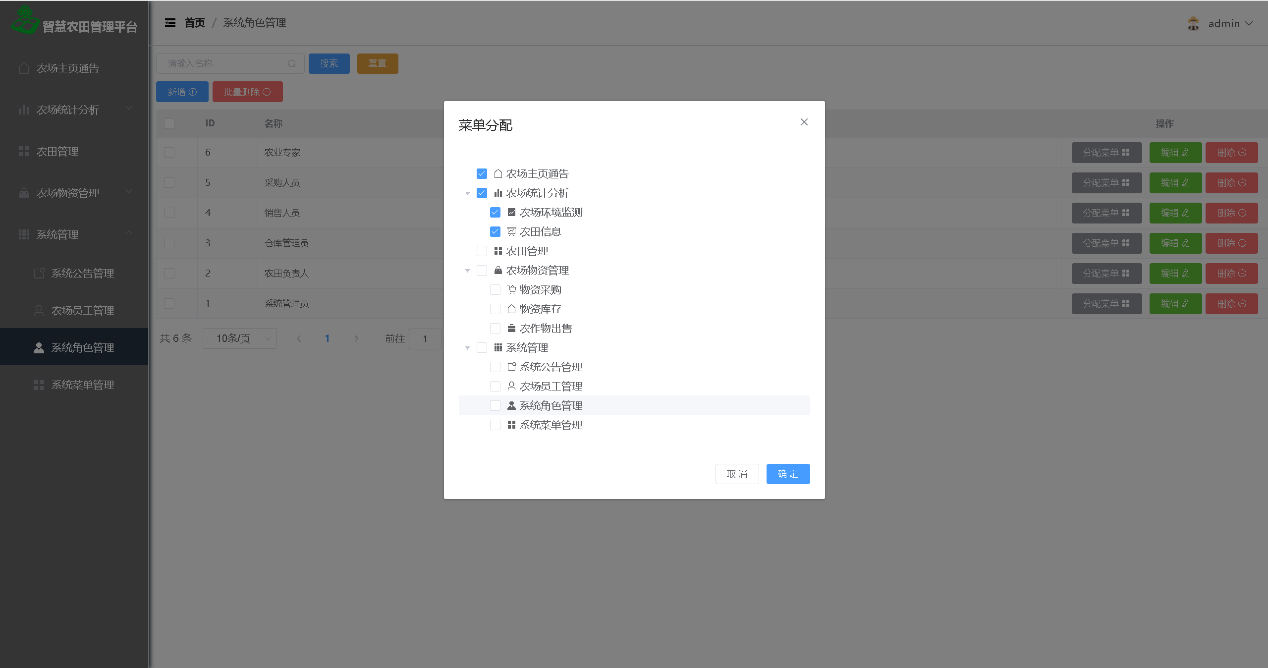
**图4-33 新增系统角色图**

编辑系统角色。如图4-34所示



**图4-34 系统角色管理图**

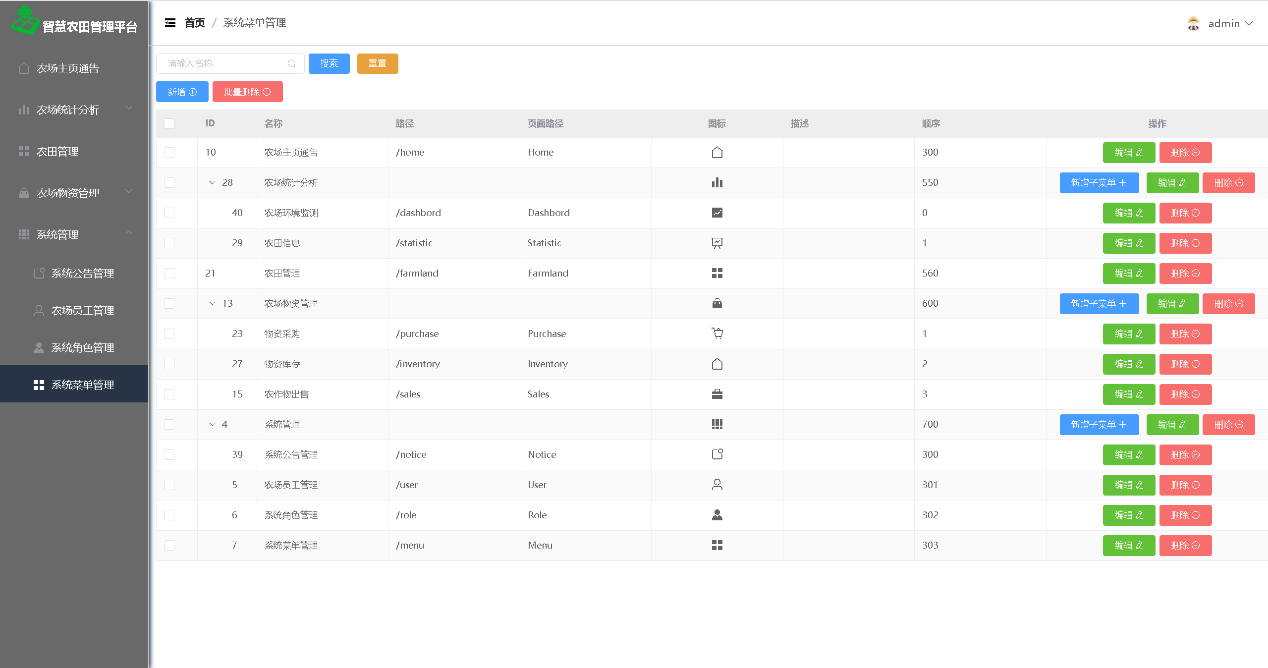
给系统角色分配菜单功能。如图4-35所示



**图4-35 系统角色分配菜单图**

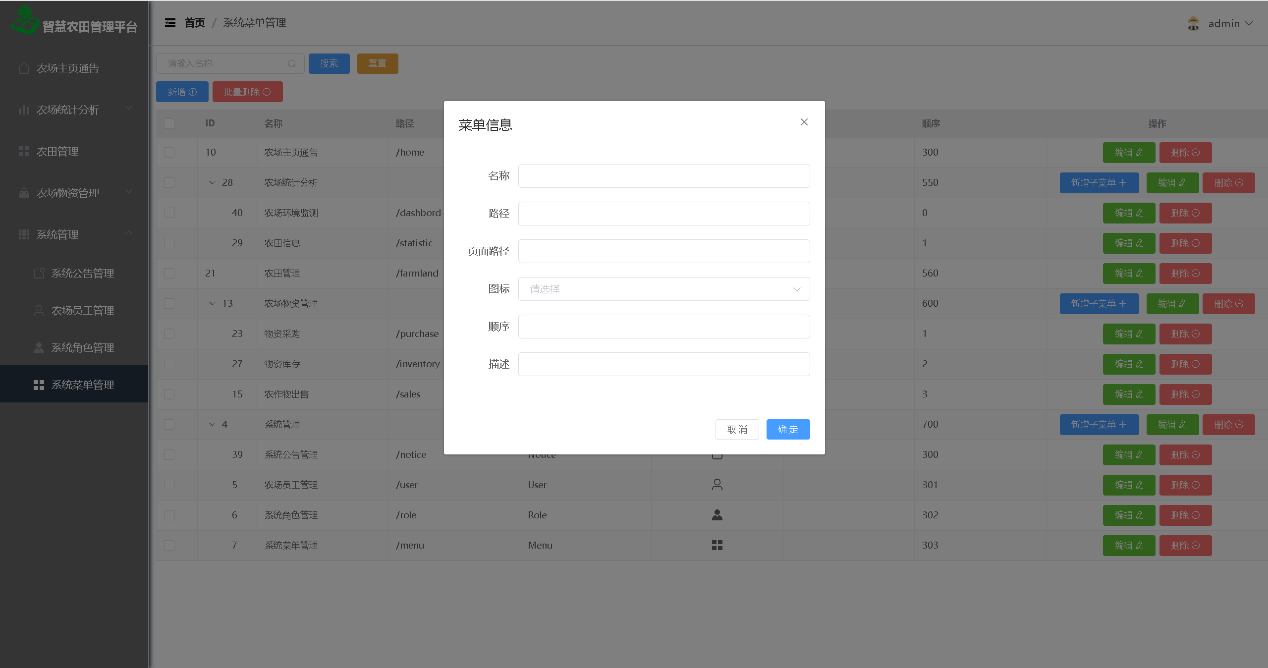
**4.5.11 系统菜单管理**

系统菜单管理：由管理员进行管理，可以新增系统菜单、编辑系统菜单、删除系统菜单、和新增系统子菜单等功能。如图4-36所示。



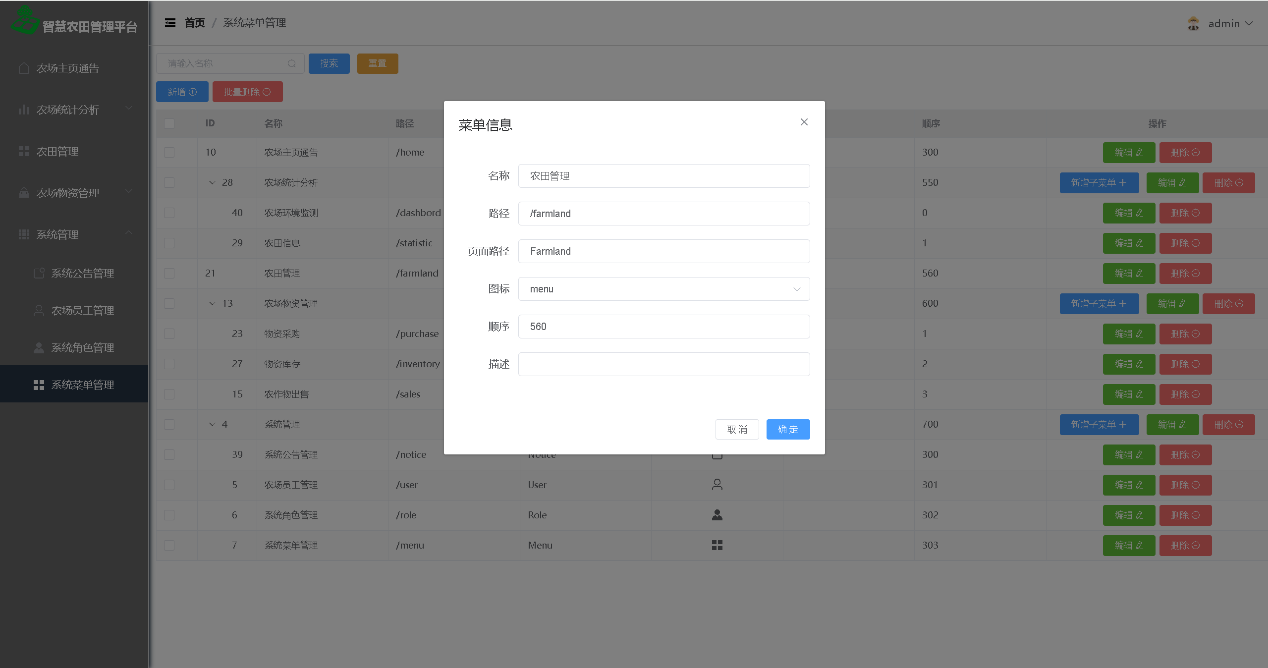
**图4-36 系统菜单管理图**

新增系统菜单。如图4-37所示。



**图4-37 新增系统菜单图**

编辑系统系统。如图4-38所示。



**图4-38编辑系统菜单**

模块对应文件详细分析

前端模块分析

登录注册模块

文件：Login.vue、Register.vue

功能：用户登录和注册操作，验证用户身份，提交用户信息到后端进行处理。

实现方式：通过表单提交用户输入的用户名和密码等信息，使用HTTP请求与后端交互，获取登录结果。

用户管理模块

文件：User.vue、Person.vue、Password.vue

功能：用户信息管理、个人信息查看和编辑、密码修改等操作。

实现方式：从后端获取用户数据并展示，通过表单提交修改后的用户信息，使用HTTP请求与后端交互更新用户数据。

农田管理模块

文件：Farmland.vue

功能：农田信息的展示和管理，包括农田的基本信息、位置、面积等。

实现方式：通过HTTP请求从后端获取农田数据，以表格或列表形式展示，支持添加、编辑和删除农田信息。

库存管理模块

文件：Inventory.vue

功能：管理库存中的物品信息，包括物品名称、类型、数量、单位、位置等。

实现方式：从后端获取库存数据并展示，支持添加、编辑和删除库存物品，通过表单提交数据到后端。

采购管理模块

文件：Purchase.vue

功能：记录和管理采购信息，包括采购物品名称、类型、数量、单位、供应商等。

实现方式：通过表单提交采购信息到后端，从后端获取采购数据并展示，支持添加、编辑和删除采购记录。

销售管理模块

文件：Sales.vue

功能：管理销售信息，包括销售物品名称、类型、数量、单位、客户等。

实现方式：从后端获取销售数据并展示，支持添加、编辑和删除销售记录，通过表单提交数据到后端。

通知管理模块

文件：Notice.vue

功能：发布和管理通知信息，包括通知标题、内容等。

实现方式：通过表单提交通知信息到后端，从后端获取通知数据并展示，支持添加、编辑和删除通知记录。

角色管理模块

文件：Role.vue

功能：管理用户角色信息，包括角色名称、权限等。

实现方式：从后端获取角色数据并展示，支持添加、编辑和删除角色信息，通过表单提交数据到后端。

统计分析模块

文件：Statistic.vue

功能：对农田、库存、采购、销售等数据进行统计分析，生成统计图表。

实现方式：从后端获取统计数据，使用图表库（如ECharts）进行数据可视化展示。

菜单管理模块

文件：Menu.vue

功能：管理应用的菜单配置，包括菜单名称、路径、组件等。

实现方式：从后端获取菜单数据并动态生成菜单，支持添加、编辑和删除菜单项。

文件管理模块

文件：FileController.java（后端）

功能：管理文件的上传、下载和删除操作。

实现方式：通过HTTP请求处理文件上传和下载，存储文件信息到数据库，支持文件的增删改查操作。

后端模块分析

用户管理模块

文件：User.java、UserMapper.java、UserService.java、UserServiceImpl.java、UserController.java

功能：用户信息的增删改查操作，用户登录认证，用户角色关联管理。

实现方式：通过MyBatis-Plus进行数据库操作，使用JWT进行用户认证和授权，控制器处理HTTP请求并调用服务层方法。

角色管理模块

文件：Role.java、RoleMapper.java、RoleService.java、RoleServiceImpl.java、RoleController.java

功能：角色信息的增删改查操作，角色与菜单、用户的关联管理。

实现方式：使用MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑，包括角色与菜单、用户的关联操作。

菜单管理模块

文件：Menu.java、MenuMapper.java、MenuService.java、MenuServiceImpl.java、MenuController.java

功能：菜单信息的增删改查操作，菜单与角色的关联管理。

实现方式：通过MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑，包括菜单与角色的关联操作。

农田管理模块

文件：Farmland.java、FarmlandMapper.java、FarmlandService.java、FarmlandServiceImpl.java、FarmlandController.java

功能：农田信息的增删改查操作。

实现方式：使用MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑。

库存管理模块

文件：Inventory.java、InventoryMapper.java、InventoryService.java、InventoryServiceImpl.java、InventoryController.java

功能：库存信息的增删改查操作。

实现方式：通过MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑。

采购管理模块

文件：Purchase.java、PurchaseMapper.java、PurchaseService.java、PurchaseServiceImpl.java、PurchaseController.java

功能：采购信息的增删改查操作。

实现方式：使用MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑。

销售管理模块

文件：Sales.java、SalesMapper.java、SalesService.java、SalesServiceImpl.java、SalesController.java

功能：销售信息的增删改查操作。

实现方式：通过MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑。

通知管理模块

文件：Notice.java、NoticeMapper.java、NoticeService.java、NoticeServiceImpl.java、NoticeController.java

功能：通知信息的增删改查操作。

实现方式：使用MyBatis-Plus进行数据库操作，控制器处理HTTP请求，服务层实现业务逻辑。

统计分析模块

文件：Statistic.java、StatisticMapper.java、StatisticService.java、StatisticServiceImpl.java、StatisticController.java

功能：对各模块数据进行统计分析，生成统计结果。

实现方式：通过MyBatis-Plus进行数据库操作，使用SQL语句进行数据统计，控制器处理HTTP请求，服务层实现业务逻辑。

文件管理模块

文件：Files.java、FileMapper.java、FileService.java、FileServiceImpl.java、FileController.java

功能：文件的上传、下载、删除等操作。

实现方式：处理文件上传请求，将文件存储到指定路径，将文件信息存储到数据库，支持文件的下载和删除操作。

权限管理模块

文件：AuthAccess.java、JwtInterceptor.java、Constants.java、Result.java、RoleEnum.java

功能：用户认证和授权，权限验证。

实现方式：使用JWT进行用户认证，通过拦截器验证用户权限，常量类和结果类用于统一返回格式和常量定义。

WebSocket模块

文件：WebSocketServer.java、WebSocketConfig.java

功能：实现WebSocket通信，用于实时消息推送。

实现方式：配置WebSocket端点，实现消息发送和接收功能，维护在线用户列表。

配置模块

文件：CorsConfig.java、InterceptorConfig.java、MybatisPlusConfig.java、SwaggerConfig.java

功能：跨域配置、拦截器配置、MyBatis-Plus配置、Swagger配置。

实现方式：通过Spring Boot的配置类进行相关配置，实现跨域支持、拦截器注册、MyBatis-Plus插件配置和Swagger文档生成。

工具模块

文件：TableColumn.java、TokenUtils.java

功能：提供表格列配置和Token工具方法。

实现方式：表格列类用于前端表格列的配置，Token工具类用于JWT的生成和解析。

# 小程序功能设计文档

**2. 系统架构设计**

**2.1 系统整体架构**

系统采用分层架构设计，包括：

* **前端展示层：** 小程序及H5界面（支持移动端及PC端访问）
* **业务逻辑层：** RESTful API 服务、微服务架构（Spring Cloud Gateway集成）
* **数据层：** 数据库存储（农田信息、设备状态、监控数据、历史记录等）
* **第三方集成层：** GIS地图服务（Leaflet.js、WGS84坐标转换）、流媒体服务（RTSP/WebRTC）、AI分析接口（OpenCV、DNN模型）

**2.2 技术选型**

* **前端：** 小程序框架 微信小程序、H5技术、Leaflet.js
* **后端：** Java/Spring Boot、Spring Cloud Gateway、MyBatis
* **数据库：** MySQL/PostgreSQL（支持GIS数据扩展）、Redis缓存
* **数据分析：** ECharts、Spark离线计算、机器学习库（随机森林、XGBoost）
* **安全：** Spring Security、JWT、AES加密

**3. 功能模块设计**

**3.1 农田管理（核心场景支撑）**

* **农田地理信息：**
  + **功能描述：** 展示农田分布、面积测量、边界绘制及坐标定位。
  + **关键技术点：**
    - 基于Leaflet.js实现地图可视化
    - WGS84坐标系转换
    - 地图数据与数据库（田块ID）绑定
* **设备监控：**
  + **功能描述：** 展示摄像头实时画面、监控设备状态（在线、离线、预警）。
  + **关键技术点：**
    - RTSP/WebRTC流媒体传输
    - 多终端适配（移动端H5播放器）
    - 状态数据实时更新与异常预警
* **田块属性：**
  + **功能描述：** 管理作物类型、种植周期、土壤信息等元数据；支持批量导入/导出与关联分析。
  + **关键技术点：**
    - Excel模板数据导入导出
    - 属性数据的批量处理及统计分析

**3.2 数据分析与可视化（决策支持核心）**

* **环境监测看板：**
  + **功能描述：** 实时展示温湿度、光照、土壤数据等环境参数仪表盘。
  + **关键技术点：**
    - ECharts动态图表展示
    - MQTT数据流订阅机制
* **统计分析：**
  + **功能描述：** 提供多维数据筛选（时间、区域、作物类型）、历史数据对比及趋势预测。
  + **关键技术点：**
    - SQL聚合查询优化
    - Spark离线计算框架支持大数据分析
* **产量预测模型：**
  + **功能描述：** 通过历史数据构建机器学习模型（如随机森林、XGBoost）进行产量预测，并可视化展示预测结果。
  + **关键技术点：**
    - 数据预处理（缺失值填充、标准化）
    - API网关集成，提供预测服务接口

**3.3 农资供应链协同**

* **采购管理：**
  + **功能描述：** 对接供应商、订单跟踪及电子合同归档。
  + **关键技术点：**
    - PDF电子签章（如DocuSign API）
    - 多级审批流程管理
* **库存管理：**
  + **功能描述：** 自动记录物资出入库情况，支持条码或RFID扫描及低库存预警。
  + **关键技术点：**
    - WebSocket实现实时库存同步
    - 条码生成与扫描规则引擎
* **销售溯源：**
  + **功能描述：** 追踪农产品从种植到消费者的全流程，提供销售数据分析（渠道效益热力图）。
  + **关键技术点：**
    - 区块链存证接口（可选）
    - 数据仓库建模（星型模型设计）

**3.4 系统管理与安全**

* **RBAC权限体系：**
  + **功能描述：** 动态角色分配（管理员、采购员、农场主），细化字段级权限控制。
  + **关键技术点：**
    - Spring Security与JWT令牌机制
    - 数据库行级权限控制
* **运维监控：**
  + **功能描述：** 展示系统健康状态（CPU、内存、磁盘使用率）及日志分析、异常检测。
  + **关键技术点：**
    - Prometheus监控告警
    - ELK日志聚合系统
* **数据安全：**
  + **功能描述：** 实现敏感操作二次验证及敏感数据脱敏存储。
  + **关键技术点：**
    - MyBatis拦截器防SQL注入
    - AES加密算法集成

**3.5 新增功能模块**

* **视频监控集成：**
  + **功能描述：** 集成视频监控模块，实现摄像头实时监控及历史视频回放。
  + **关键技术点：**
    - RTSP流解析与WebRTC低延迟传输
    - 多终端适配，确保移动端、PC端均可稳定观看
* **GIS地图集成：**
  + **功能描述：** 通过GIS地图展示农田及设备分布，实现区域定位、边界绘制和空间数据查询。
  + **关键技术点：**
    - 基于Leaflet.js实现交互式地图展示
    - WGS84坐标系与本地坐标转换支持
* **智能算法扩展：**
  + **功能描述：** 利用AI能力实现移动侦测和作物病害识别，提供自动预警与决策支持。
  + **关键技术点：**
    - OpenCV结合DNN模型进行图像识别
    - 边缘计算实现本地数据预处理，降低云端计算压力

**4. 技术亮点与创新点**

**4.1 智能物联集成**

* **多协议设备接入：** 支持MQTT、HTTP、ONVIF协议，实现多种智能设备的数据采集。
* **边缘计算网关：** 在本地处理初步数据，减少云端负载，并提升实时响应能力。

**4.2 AI能力扩展**

* **移动侦测与预警：** 利用视频监控实时检测异常活动，自动触发预警机制。
* **作物病害识别：** 集成机器学习模型，对作物病害进行自动识别，支持预防与精细化管理。

**5. 详细设计**

**5.1 数据结构与数据库设计**

* **农田与设备信息：** 建立农田、田块、设备、供应链等模块的关联数据表，支持GIS数据存储。
* **日志与监控数据：** 设计高性能日志数据表和时序数据库存储环境监测、运维监控数据。

**5.2 API设计与接口说明**

* 采用RESTful风格设计接口，确保前后端解耦。
* 对外提供农田数据、设备状态、监控视频、数据分析结果、权限管理等各类服务接口。

**5.3 前端设计与交互流程**

* **小程序端：**
  + 提供清晰的导航菜单，用户可根据角色（农场主、管理员、采购员等）快速定位常用功能。
  + 集成GIS地图、实时视频流展示、数据图表及预警信息。
* **交互流程：**
  + 用户登录 → 首页仪表盘 → 模块选择（农田管理、数据分析、供应链协同、系统管理） → 具体业务操作
  + 视频监控、GIS地图、智能算法相关功能均在各自模块中集成，确保数据无缝切换

**6. 项目实施计划与风险控制**

**6.2 风险控制措施**

* **技术风险：** 针对关键技术（如视频流传输、AI识别）进行技术预研与原型验证。
* **集成风险：** 制定详细的接口标准，确保第三方服务的稳定对接。
* **安全风险：** 加强权限管理、数据加密、日志监控，确保系统数据安全。
* **进度风险：** 采用敏捷开发模式，定期评审进度，及时调整资源分配。

**7. 附录**

**7.1 缩略词说明**

* **GIS：** 地理信息系统
* **RTSP：** 实时流协议
* **WebRTC：** 实时通信技术
* **MQTT：** 消息队列遥测传输协议
* **DNN：** 深度神经网络
* **RBAC：** 基于角色的访问控制

**7.2 参考文献**

* GIS技术文档与Leaflet.js官方文档
* 流媒体传输与WebRTC相关技术白皮书
* AI图像识别及OpenCV/DNN应用指南

**8. 总结**

本小程序通过整合农田管理、数据分析、供应链协同与系统安全等核心功能，并新增视频监控、GIS地图和智能算法扩展，实现了端到端的智能农业管理解决方案。项目在架构设计、技术选型及功能集成上均体现出前沿性和实用性，旨在为用户提供高效、智能、可扩展的农业管理平台。

1. 总结和展望

智慧农田管理平台开发是一项复杂的设计题材。它可以为农场在线管理农田提供便利利和提高效率。在本次个人毕业性项目中，我成功地设计和开发了一个基于网络的智慧农田管理平台，并完成智慧农田管理平台的开发。

首先，在设计和开发过程中，我充分理解了农场所需要系统的需求和功能。我分析了农场管理员和员工的权限，并根据这些需求设计了相应的功能模块，智慧农田管理平台：主要分为农场主页系统公告通知、农场统计分析、农田管理、农作物资管理、系统管理，五大模块。整个过程参与有管理员和农场员工两种用户类型。按照功能和权限进行角色划分，可划分为：管理员：管理系统中所有的功能，可以自行定义系统。农场员工：农场员工，可以划分为为采购人员、销售人员、农田管理人员。

其次，在技术实现方面，我选择了springboot+vue前后端分离的技术架构，完成了智慧农田管理平台的开发。我使用了vue+elementUI+ echarts实现系统的前端界面，使用了springboot框架来开发系统的后端逻辑。同时，我还使用了MySQL数据库来存储系统信息。这些主流技术的选择和应用使得智慧农田管理平台系统具有良好的性能和稳定性。

展望未来，智慧农田管理平台的发展不仅将改变传统的农业生产方式，还将推动农业向数字化、智能化转型，为农业生产带来革命性的变化。随着技术的不断创新和应用，智慧农田管理平台有望成为未来农业发展的新引擎。

总之，通过本次个人毕业性项目的设计和开发，我对在智慧农田管理平台的设计和实现有了更深入的了解，并积累了丰富的经验和技能。我相信，在未来的工作中，我将能够更好地应对类似的挑战，智慧农田管理平台有望成为未来农业发展的新引擎。

参考文献

[1] 汤鑫,杨贞萍,杨灿,等.基于SpringBoot和Vue的图书馆座位预约系统的设计与实现[J].电脑知识与术,2024,20(11):87-90.DOI:10.14004/j.cnki.ckt.2024.0519.

[2] 杜瑾.基于区块链与SpringBoot的渔业溯源管理系统设计[J].电脑知识与技术,2024,20(10):56-60.DOI:10.14004/j.cnki.ckt.2024.0552.

[3] 林静,文银刚.基于SpringBoot+Vue的医学科研数据管理平台的设计与实现[J].价值工程,2023,42(12):126-128.

[4] 王琪.SpringBoot在线学习系统的开发[J].互联网周刊,2023,(05):60-62.

[5] 董袁泉,贾苏,钱梦颖.基于SpringBoot的自动化车座安排系统[J].电脑知识与技术,2023,19(02):47-49.DOI:10.14004/j.cnki.ckt.2023.0058.

[6] 高迎.基于Vue技术框架的短视频广告交易系统开发[J].科技与创新,2024,(12):26-29+33.DOI:10.15913/j.cnki.kjycx.2024.12.006.

[7] 王淑娥.基于Vue和HTML5的响应式Web设计在信息技术领域的应用[J].信息记料,2024,25(05):121-123.DOI:10.16009/j.cnki.cn13-1295/tq.2024.05.001.

[8] 陈络琦,李博,孙榜,等.基于Vue+Spring Boot的智慧平煤神马青年云平台[C]//中国煤炭学会煤矿自动化专业委员会.第31届全国煤矿自动化与信息化学术会议暨第12届中国煤矿信息化与自动化高层论坛论文集.中平信息技术有限责任公司;,2023:9.DOI:10.26914/c.cnkihy.2023.072118.

[9]陈新龙.怎样进入MySQL数据库的学习[N].电脑报,2024-06-10(045).DOI:10.28184/n.cnki.ndina.2024.000458.

[10] 柳青,程晨.MYSQL数据库技术应用一体化课程开发研究[J].造纸装备及材料,2024,53(05):251-253.

[11] 陈成岗.业务型主站数据库从Oracle到MySQL数据架构迁移的设计[J].电工技术,2024,(06):114-116.DOI:10.19768/j.cnki.dgjs.2024.06.031.

[12] 白添予.基于MyBatisPlus的数据库框架优化综述[J].电脑与信息技术,2024,32(03):75-77+133.DOI:10.19414/j.cnki.1005-1228.2024.03.021.

[13] 桑冉航,李晓明.基于Spring Boot的健身房管理系统的设计与实现[J].电脑知识与技术,2023,19(22):54-56.DOI:10.14004/j.cnki.ckt.2023.1239.