

班 级 2103018

学 号 20009100359

西安电子科技大学

本科毕业设计论文



题 目 农业果实称重云端软件的设计与实现

学 院 计算机科学与技术院

专 业 计算机科学与技术

学生姓名 江志航

导师姓名 霍秋艳

摘 要

在现代农业向智慧化转型的进程中，精准收获成为提升农业生产效率与质量的关键环节。其中，果蔬采摘的称重工作是精准收获的重要组成部分。随着电子秤的广泛应用，其在智慧农场果蔬采摘中的作用日益凸显。然而，目前果蔬采摘很大程度上依旧依赖人工操作，这就使得对采摘果实进行准确的称重记录以及对果蔬产量进行科学管理变得尤为迫切。

为满足上述需求，本研究基于 Java 技术栈和 Vue 框架设计并实现了一款前后端分离的果实称重云端软件。在前端部分，选用 Vue 框架进行构建。Vue 具有简洁的语法、高效的虚拟 DOM 机制以及良好的组件化开发模式，能够为用户提供流畅、美观且交互性强的操作界面，使用户可以方便地进行电子秤配置、数据查看与管理等操作。后端则采用 Java 技术栈，利用其丰富的类库和强大的跨平台性能，确保系统的稳定性和可扩展性。通过 Spring Boot 框架简化开发流程，使用 Spring JPA 进行数据库操作，保证系统能够高效处理大量的称重数据。

该软件具备多项核心功能。首先，支持基于 HTTP、MQTT 等不同协议的电子秤，能够兼容市场上多种类型的电子秤设备，增强了系统的通用性和适应性。其次，提供了灵活的配置功能，用户可以方便地新增电子秤，设置对应的密钥，以及根据不同电子秤的数据特点配置数据格式，确保能够准确解析和处理来自各种电子秤的称重数据。此外，软件还具备强大的统计分析能力，可以按照批次、采摘人员等维度对称重数据进行深入分析，能够将分析结果导出为常见的数据格式，并自动生成详细的统计报告，为农场管理者提供决策依据。

为了验证软件的功能和性能，本研究模拟了称重终端与云端软件的交互过程。通过模拟不同协议的电子秤向云端软件发送称重数据，全面测试软件在数据接收、解析、存储和处理等方面的功能完整性。同时，从多个维度对软件性能进行评估，包括可用性、每秒查询率（QPS）、并发用户数和响应时间等。可用性方面，通过长时间运行测试，确保软件在各种情况下都能稳定可靠地运行；QPS 测试用于评估软件在单位时间内处理查询请求的能力；并发用户数测试模拟多个用户同时使用软件的场景，检验系统在高并发情况下的性能表现；响应时间测试则关注用户操作后软件的响应速度，以保证良好的用户体验。

综上所述，本研究实现的果实称重云端软件为智慧农场的果蔬采摘称重管理

提供了一个高效、稳定且功能丰富的解决方案，有助于推动智慧农场的精准收获和数字化管理。

关键词：智慧农业 称重系统 Spring Boot Vue MQTT 信息管理系统

ABSTRACT

In the process of the transformation of modern agriculture towards intelligence, precise harvesting has become a crucial part in improving the efficiency and quality of agricultural production. Among them, the weighing of fruits and vegetables during the picking process is an important component of precise harvesting. With the widespread application of electronic scales, their role in the fruit and vegetable picking in smart farms has become increasingly prominent. However, at present, fruit and vegetable picking still largely relies on manual labor, which makes it particularly urgent to accurately record the weight of the picked fruits and scientifically manage the fruit and vegetable yields.

To meet the above - mentioned requirements, this study designed and implemented a cloud - based fruit weighing software with a front - end and back - end separation architecture based on the Java technology stack and the Vue framework. For the front - end, the Vue framework was selected for construction. Vue features a concise syntax, an efficient virtual DOM mechanism, and a good component - based development model, which can provide users with a smooth, aesthetically pleasing, and highly interactive operation interface. Users can conveniently configure electronic scales, view and manage data through this interface. The back - end adopts the Java technology stack, leveraging its rich class libraries and strong cross - platform capabilities to ensure the stability and scalability of the system. The Spring Boot framework is used to simplify the development process, and Spring JPA is employed for database operations, guaranteeing that the system can efficiently process a large amount of weighing data.

This software has multiple core functions. Firstly, it supports electronic scales based on different protocols such as HTTP and MQTT, enabling it to be compatible with various types of electronic scale devices on the market, which enhances the system's universality and adaptability. Secondly, it provides flexible configuration functions. Users can easily add new electronic scales, set corresponding keys, and configure data formats according to the data characteristics of different electronic scales, ensuring accurate parsing and processing of weighing data from various electronic scales. In addition, the software has powerful statistical analysis capabilities. It can conduct in - depth analysis of weighing

data according to dimensions such as batches and pickers, export the analysis results in common data formats, and automatically generate detailed statistical reports, providing decision - making basis for farm managers.

To verify the functionality and performance of the software, this study simulated the interaction process between the weighing terminal and the cloud - based software. By simulating electronic scales with different protocols sending weighing data to the cloud - based software, a comprehensive test was carried out on the software's functionality integrity in data reception, parsing, storage, and processing. Meanwhile, the software performance was evaluated from multiple dimensions, including availability, Queries Per Second (QPS), concurrent user count, and response time. In terms of availability, long - term operation tests were conducted to ensure that the software can run stably and reliably under various conditions. The QPS test was used to evaluate the software's ability to process query requests per unit time. The concurrent user count test simulated scenarios where multiple users used the software simultaneously to examine the system's performance under high - concurrency situations. The response time test focused on the software's response speed after user operations to ensure a good user experience.

In conclusion, the cloud - based fruit weighing software implemented in this study provides an efficient, stable, and feature - rich solution for the weighing management of fruit and vegetable picking in smart farms, which contributes to promoting precise harvesting and digital management in smart farms.

Keywords: Smart agriculture Weighing System Spring Boot Vue MQTT
Information management system

目 录

第一章 绪论

1.1 研究背景与意义

在全球人口持续增长以及人们对农产品品质与安全关注度日益提升的大背景下,农业的发展面临着前所未有的挑战与机遇。智慧农业作为一种融合了现代信息技术、生物技术、工程技术等多学科技术的新型农业发展模式^{赵春江 2021 智慧农业的发展现状与未来展望},正逐渐成为推动农业现代化进程的关键力量。它通过对农业生产过程中的各种数据进行实时采集、传输、分析和处理,实现了农业生产的精准化、智能化和自动化管理^{李道亮 2012 物联网与智慧农业},有效提高了农业生产效率、降低了生产成本、保障了农产品质量安全,为解决当前农业发展面临的诸多问题提供了新的思路和方法。

果实称重作为精准农业的重要环节,对于农业生产管理和决策具有不可或缺的作用^{罗锡文 2016 信息技术提升农业机械化水平}。准确的果实称重数据能够为农民提供有关农作物产量的精确信息,帮助他们及时了解农作物的生长状况,^{翁杨 2019 基于深度学习的农业植物表型研究综述}而合理安排农事活动,如施肥、灌溉、病虫害防治等,以提高农作物的产量和质量。通过对果实称重数据的分析,还可以为农产品的市场定价、销售策略制定以及供应链管理提供有力的数据支持,有助于优化农产品的流通环节,提高农业经济效益^{Lipcsei2021AnalysisOA}。在水果采摘环节,精确的称重能够实现水果按重量分级,满足不同消费者的需求,提升水果的市场竞争力^{Ji2019};在农产品加工行业,准确的原料称重是保证产品质量一致性和生产工艺稳定性的关键因素。

然而,传统的果实称重方式主要依赖人工操作,存在着诸多明显的不足。人工称重效率低下,在大规模的农业生产中,需要耗费大量的人力和时间,尤其是在果实收获季节,繁重的称重工作往往使人力成本大幅增加^{Jiang2012}。人工称重容易受到人为因素的影响,导致称重数据的准确性和可靠性难以保证,如操作人员的疲劳、疏忽以及读数误差等,都可能使称重数据出现偏差,从而影响后续的生产决策^{Chen2002}。传统称重方式的数据记录和管理也较为繁琐,难以实现数据的实时共享和高效分析,无法满足现代精准农业对数据快速处理和深度挖掘的需求^{Widagdo2020RecordingSO}。

随着云计算、物联网、大数据等信息技术的飞速发展,将这些先进技术应用于农业领域,构建农业果实称重云端系统,成为推动农业数字化转型的必然趋势。云

端系统能够实现果实称重数据的实时采集、传输和存储,打破了时间和空间的限制,使得农民和农业管理者可以随时随地获取和管理称重数据^{李道亮 2012 物联网与智慧农业}。通过对海量称重数据的分析挖掘,云端系统能够为农业生产提供更加科学、精准的决策支持,如产量预测、市场趋势分析等,有助于农业生产者更好地应对市场变化,降低生产风险^{韩佳伟 2022 装备与信息协同促进现代智慧农业发展研究}。云端系统还可以与其他农业信息化系统进行集成^{刘洋 2013 基于物联网与云计算服务的农业温室智能化平台研究与应用术},实现农业生产全过程的信息化管理,进一步提升农业生产的智能化水平,推动智慧农业的发展。

基于以上背景,本文研究将深入调研农村电子秤称重的流程,了解农场电子秤称重数据的特点和应用场景,设计并利用软件技术,实现对农场电子秤称重数据的实时采集,并通过云端数据库存储和分析,实现对数据的实时共享和高效分析,为农业生产提供更加科学、精准的决策支持。同时系统还将提供与称重数据相关的果实、员工、采摘作业等信息的管理和可视化展示,帮助农业生产者更好地管理农业生产环境,提高农业生产效率。

1.2 国内外研究现状

在国外,智慧农业的发展起步较早,农业果实称重技术与云端系统的应用研究也取得了较为显著的成果。美国作为农业科技强国,在农业领域广泛应用了先进的传感器技术和物联网技术^{赵春江 2021 智慧农业的发展现状与未来展望}。例如,一些大型农场采用了高精度的电子称重传感器,能够实时采集果实的重量数据,并通过无线传输技术将数据发送至云端服务器^{陈学庚 2020 农业机械与信息技术融合发展现状与方向}。在加利福尼亚州的部分果园,利用智能称重设备与云端系统相结合,实现了对水果采摘过程的精准监控和管理^{Ampatzidis2011},农场主可以通过手机或电脑随时查看每个采摘区域的果实产量、重量分布等信息,从而及时调整采摘计划和资源分配。美国还在研发基于图像识别和深度学习的果实称重与分级系统,通过对果实图像的分析^{Anisha2019FruitRU},不仅能够准确获取果实的重量,还能对果实的品质进行评估,进一步提高了果实称重与分级的智能化水平。

欧洲在农业果实称重云端系统的研究和应用方面也处于领先地位。德国的一些农业企业开发了集成化的农业生产管理平台,其中包含了果实称重与数据分析模块^{Yin2020}。这些平台通过与各种农业设备的连接,实现了果实称重数据的自动采集和实时上传,同时利用大数据分析技术对数据进行深度挖掘,为农业生产提供

了全面的决策支持，如预测产量、优化种植方案等^{Phate2021}。荷兰则专注于温室农业中的果实称重技术研究，通过在温室中安装传感器网络，实现了对温室作物果实生长过程的全程监测和重量测量^{Graaf2004}，利用云端系统对数据进行分析和管理，为温室作物的精准种植和高效管理提供了有力保障。

在国内，随着智慧农业的快速发展，农业果实称重云端系统的研究和应用也逐渐受到重视。近年来，国内高校和科研机构在该领域开展了大量的研究工作，并取得了一系列的成果。一些研究团队开发了基于物联网的果实称重系统，通过将电子秤与物联网模块相结合，实现了果实称重数据的无线传输和远程监控^{Zhu2013}。在山东的一些果园，应用了这种物联网称重系统，果农可以通过手机 APP 实时查看果实的称重数据，方便了果实采摘和销售管理^{Gao2023}。国内企业也积极投入到农业果实称重云端系统的研发和推广中。一些科技公司推出了针对不同农业场景的果实称重解决方案^{Ningbo2019}，不仅提供了硬件设备，还开发了相应的软件平台，实现了数据的云端存储、分析和可视化展示，帮助农业生产者更好地管理果实称重数据，提高农业生产效率。

然而，目前国内外的研究仍存在一些不足之处。一方面，部分果实称重系统的精度和稳定性有待提高，在复杂的农业生产环境中，容易受到温度、湿度、电磁干扰等因素的影响，导致称重数据出现偏差^{汤建华 2018}。另一方面，现有的云端系统在数据安全和隐私保护方面还存在一定的风险，随着农业数据的价值不断提升，如何保障数据的安全传输和存储，防止数据泄露和篡改，成为亟待解决的问题。此外，不同地区和不同作物的果实称重需求存在差异，现有的系统在通用性和适应性方面还需要进一步优化，以满足多样化的农业生产需求。

1.3 研究目标与内容

本研究旨在设计并实现一个高效、稳定、功能强大的农业果实称重云端系统，以满足现代农业生产对果实称重数据精准管理和深度分析的需求。具体研究目标如下：

一、系统设计与实现：运用先进的信息技术，构建一个基于云端架构的农业果实称重系统，实现果实称重数据的实时采集、传输、存储和处理。确保系统具备良好的用户界面，操作简便，易于推广使用，为农业生产者和管理者提供便捷的服务。

二、多协议兼容与设备管理：使系统支持多种常见的电子秤通信协议，如

HTTP、MQTT 等，能够与市场上不同类型的电子秤设备进行无缝对接，实现数据的准确接收和解析。同时，提供完善的电子秤设备管理功能，包括设备注册、配置、状态监测等，方便用户对电子秤进行统一管理。

三、数据分析与决策支持：开发强大的数据统计分析功能，能够对称重数据进行多维度的分析，如按批次、采摘人员、时间等维度进行统计分析，挖掘数据背后的潜在信息。通过数据分析，为农业生产提供决策支持，如产量预测、成本分析、市场趋势判断等，帮助农业生产者优化生产策略，提高经济效益。

四、系统性能优化与测试：对系统的性能进行全面优化，确保系统在高并发、大数据量的情况下能够稳定运行，具备良好的响应速度和扩展性。通过一系列的性能测试，评估系统的可用性、每秒查询率（QPS）、并发用户数和响应时间等指标，验证系统是否满足设计要求。

围绕上述研究目标，本研究的主要内容包括以下几个方面：

一、系统架构设计：深入研究云计算、物联网、大数据等相关技术，结合农业果实称重的业务需求，设计出合理的系统架构。包括云端软件前端界面设计、电子秤终端模拟界面设计、后端服务架构、数据存储方案以及通信协议的选择等，确保系统的稳定性、可扩展性和安全性。

二、功能模块开发：根据系统设计方案，开发各个功能模块，主要包括工作服务模块、称重服务模块、用户管理模块、果实模块等。在开发过程中，遵循软件工程的原则，采用先进的开发框架和技术，确保代码的质量和可维护性。

三、性能测试与优化：制定详细的性能测试计划，运用专业的测试工具对系统进行性能测试，分析测试结果，找出系统存在的性能瓶颈。针对性能瓶颈，采取相应的优化措施，如优化数据库查询语句、调整服务器配置、采用缓存技术等，不断提升系统的性能。

1.4 研究方法与技术路线

本研究综合运用多种研究方法，以确保研究的科学性、全面性和实用性，包括具体文献研究法、案例分析法、系统开发方法、测试验证法。各研究方法具体内容如下：

一、文献研究法：广泛收集和查阅国内外关于智慧农业、果实称重技术、云计算、物联网等领域的相关文献资料，包括学术期刊论文、学位论文、研究报告、专利文献以及行业标准等。对这些文献进行系统梳理和分析，了解该领域的研究现

状、发展趋势以及存在的问题，为本研究提供理论基础和研究思路，避免重复研究，确保研究的创新性和前沿性。通过对相关文献的研究，深入了解现有的果实称重技术和云端系统架构，为系统的设计与实现提供参考。

二、案例分析法：选取国内外具有代表性的农业果实称重项目和智慧农业应用案例进行深入分析，研究其系统架构、功能实现、应用效果以及面临的挑战等。通过案例分析，总结成功经验和不足之处，为本研究的农业果实称重云端系统的设计与优化提供实践参考。例如，对美国某大型农场采用的智能称重设备与云端系统相结合的案例^{Anisha2019FruitRU}进行分析，了解其在提高果实称重效率和管理水平方面的具体做法和成效，从中汲取有益的经验。

三、系统开发方法：依据软件工程的原理和方法，本项目将采用瀑布模型的开发方法来组织开发工作。瀑布模型又称生存周期模型，由 B.M.Boehm 提出，是软件工程的基础模型。其核心思想是按工序开发软件，将功能的分析、设计与实现分开，便于分工协作^{叶俊民 2006 软件工程}。农业果实称重云端系统的开发将采用结构化的分析与设计方法，把逻辑实现与物理实现分开，各项软件工程活动包括：制定开发计划、进行需求分析和说明、软件设计、程序编码、测试及运行维护。

四、测试验证法：运用专业的测试工具和方法，对开发完成的农业果实称重云端系统进行全面的测试。功能测试主要验证系统是否满足各项功能需求，如电子秤数据采集、数据传输、数据分析、报表生成等功能是否正常；性能测试主要评估系统在高并发、大数据量情况下的性能表现，包括系统的响应时间、吞吐量、并发用户数等指标；安全测试主要检测系统的安全性，如数据加密、用户认证、权限管理等方面是否存在漏洞。通过测试，及时发现系统存在的问题和缺陷，并进行优化和改进，确保系统能够稳定、可靠地运行。

本研究的技术路线主要包括需求分析、系统设计、功能模块开发、系统集成与测试、系统部署与应用以及研究总结与展望。下面对这些步骤进行具体阐述。

首先是需求分析，调研农业的实际生产流程，了解在果实称重管理方面的实际需求 and 业务流程，分析现有果实称重方式存在的问题和痛点，明确系统的功能需求、性能需求、安全需求以及用户体验需求等，整理相关的需求文档，为后续的系统设计提供依据。

接下来是系统设计。基于需求分析的结果，结合云计算、物联网、大数据等技术，进行农业果实称重云端系统的架构设计，确定系统的前端界面设计、后端

服务设计、数据库设计、数据库存储方案以及通信协议等。根据需求文档所描述的业务流程，设计并书写接口文档，方便前后端的并行开发。对于具体的业务细节，可以利用 UML 图来描述业务的具体流程，比如可以使用时序图和活动图来描述称重的流程，以及使用类图来描述各个模块的关系。在前端设计中，注重用户界面的友好性和易用性，采用响应式设计，确保系统能够在不同设备上正常运行；在后端设计中，注重服务的高可用，提供响应速度良好的接口，提高系统的可扩展性和维护性；在数据库设计中，可以遵循改进后的新奥尔良方法（New Orleans Method），将数据库设计分为需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施和数据库运行和维护这六个阶段，设计出良好、可维护的数据库苗雪兰 2001 数据库系统原理及应用教程；在数据存储方面，选择合适的数据库管理系统，如关系型数据库和非关系型数据库相结合，以满足不同类型数据的存储需求；在通信协议方面，选用 HTTP、MQTT 等常见协议，确保电子秤与云端系统之间的数据传输稳定可靠。

完成系统设计之后，即可开始进行功能模块开发。根据系统设计方案，开发各个功能模块。利用前端开发技术，如 Vue.js、Vite、TypeScript 等，实现系统的前端界面，包括电子秤配置界面、数据查看界面、报表生成界面等；利用后端开发技术，如 Java、Spring Boot、Spring Security 等，实现系统的后端服务，包括工作服务模块、称重服务模块、用户管理模块、果实模块等；利用数据库开发技术，如 MySQL、Redis 等，实现数据的存储和管理；利用通信技术，如 MQTT 客户端库、HTTP 客户端库等，实现电子秤与云端系统之间的数据传输。

完成各个模块的开发之后，即可开始进行系统集成与测试。将开发完成的各个功能模块进行集成，搭建完整的农业果实称重云端系统。制定详细的测试计划，对系统进行全面的测试，包括单元测试、集成测试、系统测试和验收测试等。在测试过程中，记录测试结果，分析测试数据，及时发现并解决系统中存在的问题和缺陷。对系统的性能进行优化，如优化数据库查询语句、调整服务器配置、采用缓存技术等，提高系统的响应速度和吞吐量，确保系统能够满足实际应用的需求。

测试完成，确保没有问题之后，即可进行系统部署与应用。将测试通过的系统部署到生产环境中，进行实际应用。在应用过程中，收集用户的反馈意见，对系统进行持续优化和改进。对系统的应用效果进行评估，分析系统在提高果实称重效率、数据管理水平以及为农业生产决策提供支持等方面的作用，总结经验教训，

为进一步完善系统和推广应用提供参考。

最后，进行研究总结与展望：对整个研究过程和结果进行总结，归纳研究成果和创新点，分析研究过程中存在的问题和不足之处，提出未来的研究方向和展望。为农业果实称重云端系统的进一步发展和应用提供理论支持和实践指导，推动智慧农业的发展。

通过以上研究方法和技术路线，本研究旨在构建一个高效、稳定、功能强大的农业果实称重云端系统，为现代农业生产提供有力的支持和保障。

第二章 系统相关技术概述

2.1 通信协议选型

本系统支持 HTTP 和 MQTT 两种通信协议，电子秤可以通过调用 HTTP 接口或者发送 MQTT 消息来完成称重信息的上传。

HTTP (HyperText Transfer Protocol) 即超文本传输协议，是一种用于分布式、协作式和超媒体信息系统的应⤵用层协议。在电子秤系统中，通过调用 HTTP 接口上传称重信息具有以下特点：

稳定性和通用性：HTTP 是互联网上广泛使用的协议，具有很高的稳定性和通用性。几乎所有的网络设备和软件都支持 HTTP 协议，这使得电子秤可以很容易地与各种不同的系统进行集成。无论是传统的服务器架构，还是现代的云服务平台，都可以通过 HTTP 接口接收电子秤上传的称重信息。例如，许多企业的内部管理系统、电商平台的库存管理系统等都可以通过 HTTP 接口与电子秤进行数据交互，实现实时的库存更新和物流跟踪^{Zhao2016}。**易于开发和调试：**HTTP 协议的开发和调试工具非常丰富。开发人员可以使用常见的编程语言和开发框架，如 Java、Python、Node.js 等，轻松地实现 HTTP 接口的调用和数据处理。同时，各种网络调试工具，如 Postman、Fiddler 等，可以帮助开发人员快速地测试和调试电子秤与服务器之间的通信，确保数据的准确传输。**支持多种数据格式：**HTTP 协议支持多种数据格式的传输，如 JSON、XML 等。这使得电子秤可以根据不同的应用需求，选择合适的数据格式上传称重信息。例如，对于需要与第三方系统进行数据集成的场景，可以选择使用 JSON 格式，因为 JSON 格式具有简洁、易读、易于解析等优点，被广泛应用于 Web 开发和数据交换领域。

2.2 数据存储技术

MySQL + Redis

在农业果实称重云端系统中，数据的存储、管理和安全至关重要。数据库技术作为数据管理的核心，为系统提供了高效的数据存储和查询功能。根据农业果实称重数据的特点和系统需求，本研究综合考虑了关系型数据库和非关系型数据库的特性，并结合使用这两种类型的数据库，以满足系统对数据管理的不同需求。

关系型数据库以其结构化的数据存储方式和强大的事务处理能力，在农业果

实称重云端系统中发挥着重要作用。常见的关系型数据库如 MySQL，它基于关系模型，将数据组织成表格形式，每个表格包含多个字段和记录。在本系统中，MySQL 主要用于存储结构化程度较高、对数据一致性和完整性要求严格的数据，如称重数据、用户信息、电子秤设备信息等。对于称重数据，每条记录包含果实的重量、采摘时间、采摘地点、采摘人员等字段，通过关系型数据库的表结构设计，可以清晰地组织和管理这些数据，方便进行数据的插入、更新和查询操作。

关系型数据库的优势在于其强大的 SQL 查询语言，能够支持复杂的查询操作。在农业果实称重云端系统中，用户可能需要根据不同的条件对称重数据进行查询，如按时间段查询某个果园的果实产量、按采摘人员查询其采摘的果实重量总和等。通过 SQL 语句，可以轻松实现这些复杂的查询需求，快速准确地获取所需数据。关系型数据库还提供了 ACID（原子性、一致性、隔离性、持久性）特性，确保在数据更新和事务处理过程中，数据的一致性和完整性不会受到破坏。在同时进行多个称重数据的插入和更新操作时，关系型数据库能够保证这些操作要么全部成功执行，要么全部回滚，避免数据出现不一致的情况。然而，关系型数据库在面对一些非结构化数据和高并发读写场景时，存在一定的局限性。对于系统产生的大量日志数据、图片数据（如果实的外观图片用于品质评估）以及一些实时性要求较高的传感器数据，关系型数据库的处理效率较低。此时，非关系型数据库则展现出其独特的优势。非关系型数据库，如 MongoDB，以其灵活的数据模型和高扩展性，适用于存储非结构化和半结构化数据。MongoDB 采用文档型数据模型，数据以 BSON（Binary JSON）格式存储，每个文档可以包含不同的字段和数据结构，无需预先定义固定的表结构。在农业果实称重云端系统中，MongoDB 主要用于存储日志数据、图片数据以及一些实时性要求较高的传感器数据。日志数据记录了系统的操作日志、错误日志等信息，其数据结构相对灵活，使用 MongoDB 可以方便地存储和查询这些日志数据。对于果实的外观图片，以二进制数据的形式存储在 MongoDB 中，结合其高效的查询性能，可以快速检索和展示相关图片。非关系型数据库的高扩展性使其能够轻松应对大规模数据和高并发读写的场景。在农业生产过程中，随着时间的推移和数据量的不断增加，系统需要具备良好的扩展性，以满足日益增长的数据存储和处理需求。MongoDB 采用分布式架构，通过分片技术将数据分散存储在多个节点上，当数据量增加时，可以通过添加更多的节点来扩展存储容量和处理能力，实现系统的横向扩展。非关系型数据库在读写性能方

面也表现出色，能够快速处理大量的并发读写请求，满足系统对实时性的要求。在果实采摘高峰期，大量的称重数据需要实时上传和存储，非关系型数据库能够快速响应这些请求，确保数据的及时处理。在数据管理方面，为了确保数据的准确性和完整性，系统采用了一系列的数据验证和清洗机制。在数据采集阶段，对电子秤上传的称重数据进行实时验证，检查数据的格式是否正确、数值是否在合理范围内等。对于不符合要求的数据，及时进行提示和纠正，确保进入系统的数据质量可靠。在数据存储过程中，定期对数据库中的数据进行清洗和整理，删除过期数据和重复数据，优化数据库的存储结构，提高数据的查询效率。为了提高数据查询的效率，系统对数据库进行了索引优化。根据常用的查询条件，为相关字段创建合适的索引。如果经常按照采摘时间查询称重数据，就可以在采摘时间字段上创建索引，这样在查询时可以大大加快数据的检索速度。同时，采用缓存技术，将频繁访问的数据缓存到内存中，减少对数据库的直接查询次数，进一步提高系统的响应速度。在数据安全方面，系统采取了多种措施来保障数据的安全性和隐私性。对数据库进行加密存储，采用 SSL/TLS 加密协议对数据传输过程进行加密，防止数据在传输过程中被窃取或篡改。通过用户认证和权限管理机制，确保只有授权用户才能访问和操作数据库中的数据。为不同的用户角色分配不同的权限，如管理员拥有最高权限，可以进行所有的数据管理操作；普通用户只能查看和管理自己相关的数据，从而有效保护数据的安全。定期对数据库进行备份，将备份数据存储在不安全的位置，以便在数据丢失或损坏时能够及时恢复数据，确保数据的可靠性和可用性。

2.3 后台 Spring 技术栈

2.4 前台 Vue 技术栈

第三章 系统需求分析与设计

3.1 系统服务需求分析与设计

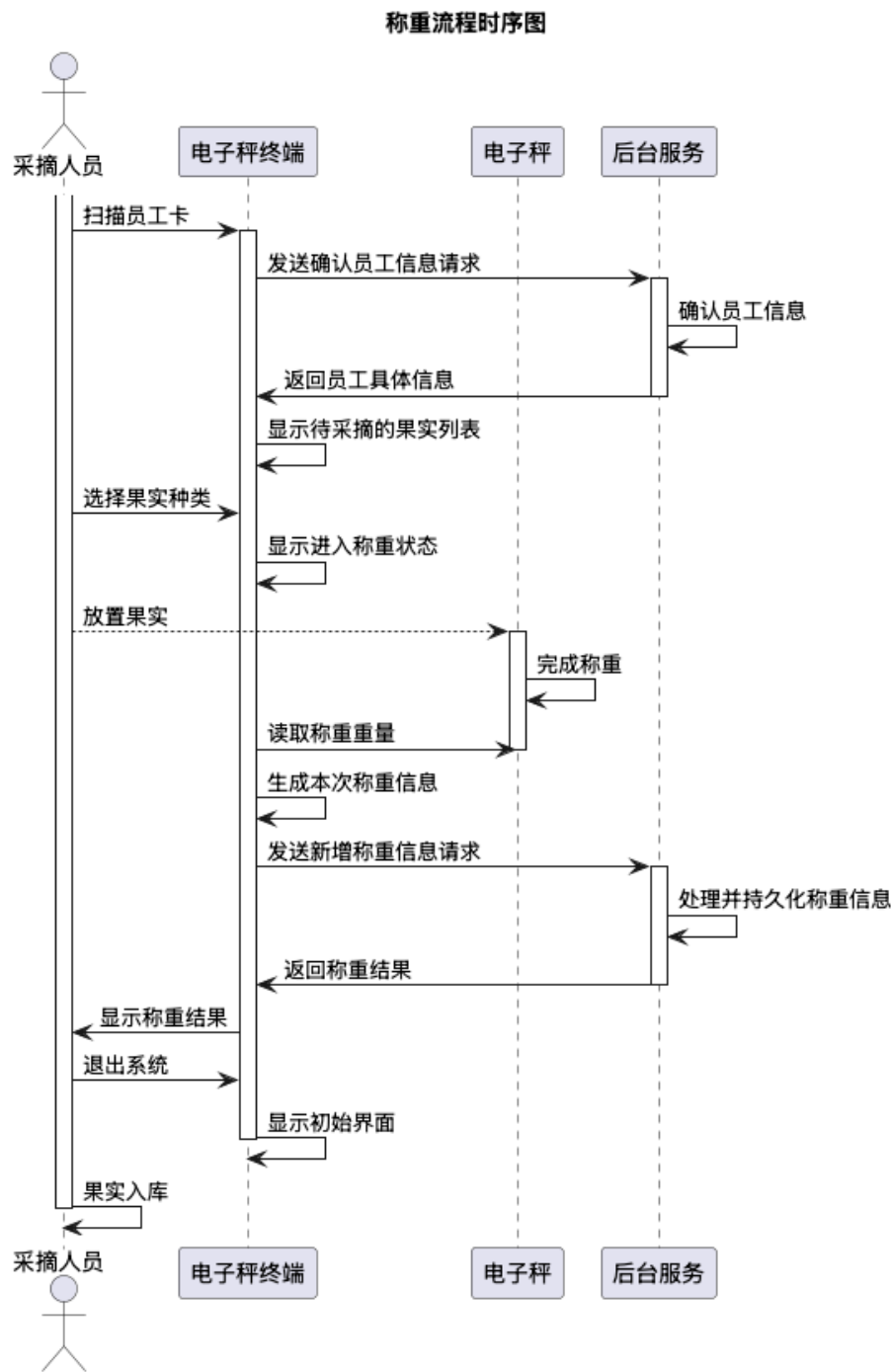


图 3.1 称重流程时序图

3.2 服务接口设计

3.3 软件架构设计

第四章 系统实现与关键技术细节

第五章 系统测试与性能评估

第六章 农业果实称重云端系统的应用案例分析

第七章 结论与展望

致 谢

参考文献