****

**本科毕业设计**

**（2025届）**

|  |  |
| --- | --- |
| **题 目** | **Design and Implementation of Information Management System for Production Equipment in Intelligent Factory** |
| **学 院** | **圣光机学院** |
| **专 业** | **计算机科学与技术** |
| **班 级** | **21320112** |
| **学 号** | **21321229** |
| **学生姓名** | **袁之腾** |
| **指导教师** | **李运发 Belozubov Aleksandr** |
| **完成日期** | **2025年6月** |

**诚 信 承 诺**

我谨在此承诺：本人所写的毕业论文《 Design and Implementation of Information Management System for Production Equipment in Intelligent Factory 》均系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释，若有不实，后果由本人承担。

**承诺人（签名）：**

**2025年 5 月 6 日**

**Abstract**

In this paper, we design and implement an intelligent factory production equipment information management system, which adopts lightweight Web architecture and has core functions such as user management, equipment temperature monitoring, CPU utilization monitoring, spare parts management and fault records, etc. The system is based on B/S three-tier architecture. The system is based on B/S three-tier architecture, with responsive design in native JavaScript on the front-end, RESTful API in Flask framework on the back-end, and MySQL database for data storage. Through Python script, the system can collect and process equipment temperature and CPU utilization data in real time, support multi-source data collection and parallel monitoring of multiple devices, and also adopt JWT authentication mechanism to ensure security. The system implements an alarm mechanism, data export function and intuitive visualization interface, which greatly improves the user experience. Tests show that the system is stable and practical under high concurrency. In the future, RAG technology will be introduced, combined with a large model more suitable for the company's small assistant, the development of mobile applications, and the realization of multi-sensor data fusion, in order to enhance the level of intelligence.

**Key words:** Industrial equipment monitoring Flask framework MySQL database Temperature monitoring CPU utilization monitoring Management of spare parts Artificial intelligence Large model Data visualization

**Contents**

[**Introduction** 1](#_Toc196074243)

[**Chapter I. Introduction** 2](#_Toc196074244)

[**1.1 Background and significance of the study** 2](#_Toc196074245)

[**1.1.1 Development trend of smart factory** 2](#_Toc196074246)

[**1.1.2 Demand analysis of industrial equipment monitoring and management system** 3](#_Toc196074247)

[**1.1.3 The Value of Lightweight Web Architecture in Industrial Surveillance** 5](#_Toc196074248)

[**1.2 Current status of domestic and international research** 5](#_Toc196074249)

[**1.2.1 Research Dynamics of Domestic Industrial Equipment Monitoring System** 5](#_Toc196074250)

[**1.2.2 Equipment management solutions in the context of Industry 4.0 abroad** 6](#_Toc196074251)

[**1.2.3 Shortcomings of existing systems and innovations in this topic** 7](#_Toc196074252)

[**1.3 Research objectives and content** 9](#_Toc196074253)

[**1.3.1 Research objectives** 10](#_Toc196074254)

[**1.3.2 Content of the study** 10](#_Toc196074255)

[**Chapter II. System requirements analysis** 11](#_Toc196074256)

[**2.1 Functional requirements** 12](#_Toc196074257)

[**2.1.1 User management requirements** 12](#_Toc196074258)

[**2.1.2 Equipment Monitoring Requirements** 13](#_Toc196074259)

[**2.1.3 Spare parts management requirements** 13](#_Toc196074260)

[**2.1.4 Alarm system requirements** 13](#_Toc196074261)

[**2.1.5 Data export requirements** 14](#_Toc196074262)

[**2.1.6 Artificial intelligence requirements** 14](#_Toc196074263)

[**2.2 Non-functional requirements** 14](#_Toc196074264)

[**2.2.1 Reliability and Stability** 14](#_Toc196074265)

[**2.2.2 Interface friendliness and responsiveness** 14](#_Toc196074266)

[**2.3 Technical feasibility analysis** 14](#_Toc196074267)

[**2.3.1 Advantages of Flask Framework in Industrial Applications** 14](#_Toc196074268)

[**2.3.2 Application of MySQL Database in Equipment Data Storage** 14](#_Toc196074269)

[**2.3.3 Realization scheme of front-end and back-end separation architecture** 15](#_Toc196074270)

[**Chapter III. System design** 15](#_Toc196074271)

[**3.1 Overall architectural design** 15](#_Toc196074272)

[**3.1.1 B/S architecture design** 15](#_Toc196074273)

[**3.1.2 System Module Division and Interaction Relationships** 16](#_Toc196074274)

[**3.2 Power module design** 17](#_Toc196074275)

[**3.2.1 User management module** 17](#_Toc196074276)

[**3.2.2 Equipment Monitoring Module** 17](#_Toc196074277)

[**3.2.3 Spare Parts Management Module** 17](#_Toc196074278)

[**3.2.4 Fault logging module** 18](#_Toc196074279)

[**3.2.5 Alarm System Module** 18](#_Toc196074280)

[**3.2.6 Data export module** 18](#_Toc196074281)

[**3.3 Database design** 18](#_Toc196074282)

[**3.3.1 E-R diagrams and relationship models** 18](#_Toc196074283)

[**3.3.2 Core table structure design** 19](#_Toc196074284)

[**3.3.3 Data Integrity and Index Optimization** 20](#_Toc196074285)

[**3.4 Interface design** 20](#_Toc196074286)

[**3.4.1 RESTful API Design Principles** 20](#_Toc196074287)

[**3.4.2 Front-end and back-end data interaction process** 20](#_Toc196074288)

[**Chapter IV. System implementation** 21](#_Toc196074289)

[**4.1 Technology stack** 21](#_Toc196074290)

[**4.1.1 Data acquisition realization** 21](#_Toc196074291)

[**4.2 Core functionality realization** 21](#_Toc196074292)

[**4.2.1 User Authentication and Privilege Management Implementation** 21](#_Toc196074293)

[**4.2.2 Device Monitoring and Status Management Implementation** 23](#_Toc196074294)

[**4.2.3 Spare parts inventory management realization** 25](#_Toc196074295)

[**4.2.4 Alarm system realization** 26](#_Toc196074296)

[**4.2.5 Data export function realization** 28](#_Toc196074297)

[**4.2.6 Smart Factory Assistant** 28](#_Toc196074298)

[**4.3 System Deployment** 29](#_Toc196074299)

[**4.3.1 Deployment Architecture** 29](#_Toc196074300)

[**4.3.2 Windows environment deployment process** 30](#_Toc196074301)

[**4.3.3 System maintenance** 30](#_Toc196074302)

[**4.3.4 Key Technical Difficulties Solved** 30](#_Toc196074303)

[**Chapter V. System testing** 31](#_Toc196074304)

[**5.1 Test environment and methodology** 31](#_Toc196074305)

[**5.1.1 Test environment configuration** 31](#_Toc196074306)

[**5.1.2 Test methods** 32](#_Toc196074307)

[**5.2 Test Case Design** 32](#_Toc196074308)

[**5.2.1 Functional Test Cases** 32](#_Toc196074309)

[**5.2.2 Performance testing** 33](#_Toc196074310)

[**5.2.3 Compatibility test cases** 34](#_Toc196074311)

[**5.3 Test results and analysis** 34](#_Toc196074312)

[**5.3.1 Functional test results** 34](#_Toc196074313)

[**5.3.2Performance Test Results** 37](#_Toc196074314)

[**5.3.3 Compatibility test results** 42](#_Toc196074315)

[**Chapter VI. Systematic evaluation and outlook** 42](#_Toc196074316)

[**6.1Advantages and disadvantages of the system** 42](#_Toc196074317)

[**6.1.1 System advantages** 42](#_Toc196074318)

[**6.1.2 System shortcomings** 43](#_Toc196074319)

[**6.2 Future prospects and functional extensions** 44](#_Toc196074320)

[**6.2.1 Technological upgrading** 44](#_Toc196074321)

[**Conclusion:** 45](#_Toc196074322)

[**Acknowledgements** 46](#_Toc196074323)

[**References** 46](#_Toc196074324)

**Introduction**

With the arrival of the industry 4.0 era and the in-depth implementation of the strategy of “Made in China 2025”, the smart factory, as the core carrier of the transformation and upgrading of the manufacturing industry, is promoting the development of the production mode in the direction of digitization, networking, and intelligence. In the construction process of smart factory, the information management of production equipment becomes a key link, which directly affects the production efficiency, product quality and resource utilization. However, at present, many manufacturing enterprises still have problems in equipment management such as information silos, difficult data collection, and complex management systems, which are difficult to meet the needs of intelligent manufacturing. This study aims to design and implement an industrial equipment monitoring and management system based on Flask framework and MySQL database, while accessing artificial intelligence to keep up with the times and solve the pain points of equipment information management in smart factories. The system adopts lightweight Web architecture with easy deployment, flexible expansion and friendly user experience, and is capable of real-time monitoring of equipment temperature and CPU utilization, dynamic tracking of equipment status, inventory management of spare parts, and recording and analyzing of equipment failures. Compared with the traditional equipment management system, this system is more lightweight in architecture design, avoiding the development and deployment difficulties brought by the complex framework; more focused on the realization of the function, focusing on solving the equipment monitoring and maintenance problems that manufacturing enterprises are most concerned about; more flexible in technology selection, combining the simplicity of the Flask framework and the stability of the MySQL database, to build an efficient and reliable management platform. This paper first analyzes the development trend of intelligent factory and the demand of equipment management system, and then elaborates the design ideas and implementation methods of the system, including the overall architecture design, functional module design, database design and interface design. Then the specific realization process and test results of the system are introduced to verify the functional integrity and performance stability of the system. Finally, the research results are summarized, the innovations and limitations of the system are analyzed, and future research directions are proposed. Through this research, it is expected to provide manufacturing enterprises with a lightweight, easy-to-deploy, high-efficiency equipment management solution to help them realize intelligent management of the production process, improve production efficiency and product quality, reduce operating costs, and promote the transformation and upgrading of the manufacturing industry and sustainable development.

**Chapter I. Introduction**

**1.1 Background and significance of the study**

**1.1.1 Development trend of smart factory**

Against the background of increasingly fierce competition in the global manufacturing industry, smart factories have become a key strategic choice for enhancing the core competitiveness of the manufacturing industry. Smart factory refers to a new production mode that realizes the intelligence, flexibility and efficiency of the production process through the deep integration of information technology, automation technology and advanced manufacturing technology. In recent years, with the rapid development of the Internet of Things, cloud computing, big data, artificial intelligence and other emerging technologies, the practical application of smart factories is accelerating globally. In China, the “Made in China 2025” strategy put forward by the government explicitly takes the development of intelligent manufacturing as the main attack direction of China's manufacturing transformation and upgrading, and puts forward the strategic goal of “taking intelligent manufacturing as the main attack direction”. Under the impetus of this national strategy, China's manufacturing industry is accelerating to the direction of digitalization, networking, intelligence, intelligent factory construction is gradually becoming an inevitable choice for manufacturing enterprises to enhance competitiveness. The development trend of smart factory is mainly reflected in the following aspects:

Deep integration of informationization and automation: modern smart factory is no longer satisfied with the simple automated production line, but the pursuit of information systems and automation equipment in-depth integration. Through the Internet of Things (IoT) technology to achieve equipment interconnection and interoperability, through big data analysis to achieve the optimization of the production process, through artificial intelligence technology to achieve intelligent production decision-making, so as to build a highly integrated intelligent manufacturing system.

Production mode: With the increasingly obvious trend of individualization and diversification of market demand, the traditional high-volume, standardized production mode is difficult to adapt to the rapid changes in market demand. By upgrading the production system, the smart factory can quickly adjust the production plan and production line configuration according to the changes in market demand, and realize the customized production of small batch and multi-species.

Green Manufacturing: Under the background of tightening resource constraints and increasing environmental pressure, green manufacturing has become an important concept in the construction of smart factories. By optimizing the production process, improving the efficiency of resource utilization, reducing energy consumption and pollution emissions, smart factories achieve the coordinated development of economic and environmental benefits.

Globalization development: In the context of globalization, the production activities of manufacturing enterprises have broken through the geographical limitations, and the design centers, production bases, suppliers and customers distributed all over the world need to collaborate efficiently. The smart factory realizes worldwide collaborative design, collaborative manufacturing and collaborative service through networked and informatized means.

**1.1.2 Demand analysis of industrial equipment monitoring and management system**

In the process of construction and operation of smart factory, industrial equipment as the core production resources, its operation status directly affects the production efficiency, product quality and enterprise profitability. With the increase in the number and complexity of production equipment, the traditional equipment management mode has been difficult to meet the needs of modern manufacturing industry. Enterprises need a set of advanced industrial equipment monitoring and management system to realize intelligent equipment management. The following are the core requirements obtained from the research:

Real-time Monitoring Requirements: Modern industrial production requires high stability and reliability of equipment, and any equipment failure may lead to production line downtime, resulting in huge economic losses. Therefore, the system needs to be able to real-time acquisition and monitoring of equipment operating status, temperature, CPU utilization, vibration, noise and other key parameters, and timely detection of abnormalities. Real-time monitoring requires not only the timeliness of data collection, but also the visualization of monitoring information, so that operators can intuitively understand the status of the equipment and respond quickly.

Failure diagnosis and prediction needs: equipment failure often brings production interruption and maintenance costs, so timely detection of potential failures and preventive maintenance is crucial. Advanced equipment monitoring system should have the ability to diagnose and predict faults, through the in-depth analysis of equipment operation data, identify potential failure risks, and provide targeted maintenance recommendations.

Remote management requirements: With the expansion of enterprise scale and decentralization of production layout, equipment managers need to be able to remotely monitor and manage equipment distributed in different locations. The remote management function enables professional and technical personnel to monitor, diagnose and debug equipment without visiting the site, which greatly improves management efficiency and response speed.

Data analysis and decision support needs: in the era of big data, equipment operation data has become an important asset of the enterprise. The system needs to have a strong data analysis capability, able to mine and analyze the massive equipment operation data, extract valuable information, and provide support for the production decision-making of the enterprise.

Alarm and notification requirements: When the equipment is abnormal or faulty, the system needs to be able to immediately issue an alarm notification to ensure that the relevant personnel can respond in a timely manner. The alarm mechanism should support multi-level settings, according to the severity of the event to take different alarm strategies.

Data Export and Reporting Requirements: In order to support the enterprise's daily management and decision-making analysis, the system needs to provide flexible data export and report generation functions. Users should be able to choose the content, time range and format type of exported data according to their needs, which is convenient for in-depth analysis or reporting to management.

Visualization requirements: the system should provide an intuitive, beautiful visualization interface, using a variety of charts and graphs to display the equipment operation data, so that complex data become easy to understand. The visualization interface should support interactive operation to enhance user experience.

Artificial Intelligence Requirements: With the development of the times, Artificial Intelligence will become an inseparable part of people, and accessing the system to Artificial Intelligence will greatly reduce the efficiency of users in solving problems.

**1.1.3 The Value of Lightweight Web Architecture in Industrial Surveillance**

In the development of industrial equipment monitoring and management systems, the choice of technology architecture affects the performance of the system as well as the user experience. Traditional industrial monitoring systems often use heavyweight client software or proprietary monitoring platforms, these solutions are usually more complex to deploy, the user experience is poor. With the rapid development of Web technology, industrial monitoring systems based on lightweight Web architecture have gradually become the mainstream choice:

Lightweight Web architecture usually adopts modular design and modern development frameworks, greatly simplifying the development process and complexity. The rich components provided by the framework enable the development team to quickly build a fully functional application system, shorten the development cycle and reduce development costs.The deployment of the Web architecture is also more simple, only need to install the necessary operating environment on the server, you can realize the system's rapid on-line.

Lightweight Web architecture is usually based on the principle of loosely coupled design, the system components interact with each other through standardized interfaces, and this design makes the system very easy to expand functionality and module replacement. When enterprises need to add new monitoring functions or adjust existing functions, they can make local modifications without affecting the overall structure of the system, which greatly reduces the cost and risk of system maintenance and upgrading.

Lightweight Web architecture can support compatibility of many devices. Based on standard Web technologies, such as Html, css, Javascript, the developed system can run on any device that supports modern browsers, including computers with various operating systems, such as windows, macOS, Linux, and mobile devices, such as ios and android.

Lightweight Web architecture usually takes up less system resources and can run efficiently in resource-limited environments. The requirements for hardware CPU etc. are not very high.

**1.2 Current status of domestic and international research**

**1.2.1 Research Dynamics of Domestic Industrial Equipment Monitoring System**

In recent years, with the accelerated pace of transformation and upgrading of China's manufacturing industry and the in-depth promotion of the intelligent manufacturing strategy, domestic research in the field of industrial equipment monitoring system has made remarkable progress. From academic research to industrial application, it shows a booming trend. Many Chinese universities and research institutions have actively carried out research related to industrial equipment monitoring, and formed a series of valuable theoretical results and technical solutions. Research teams from Tsinghua University, Zhejiang University, Harbin Institute of Technology and other universities have conducted in-depth studies on equipment condition monitoring, fault diagnosis, predictive maintenance, etc., and put forward a variety of data-driven equipment health assessment models and fault prediction algorithms, and the domestic research has gradually formed its own characteristics and advantages. On the one hand, researchers are actively exploring the integration of artificial intelligence, big data analysis and other emerging technologies with traditional equipment monitoring systems to develop monitoring platforms with a higher degree of intelligence; on the other hand, they are also focusing on the research of low-cost, easy-to-deploy solutions suitable for China's manufacturing industry, and are striving to solve the problems of technical thresholds and cost pressures faced by small and medium-sized manufacturing enterprises in the process of intelligent transformation. A number of excellent industrial equipment monitoring system solution providers have emerged in China, such as UFIDA Networks, Oriental Guoxin, Aerospace Cloud Network and other enterprises have launched various types of industrial Internet platforms and equipment management systems, which have been widely used in the iron and steel, chemical industry, energy, equipment manufacturing and other industries. These systems are generally equipped with basic functions such as equipment connection, data acquisition, status monitoring, fault alarms, etc. Leading solutions also provide advanced functions such as predictive maintenance, energy efficiency management, and equipment life prediction. Although domestic industrial equipment monitoring system research has made great progress, but compared with the international advanced level there are still some gaps. For example, the core algorithms and key technologies of independent innovation needs to be strengthened; some foreign systems can be combined with the training of the large model to manage the use of RAG technology through the embedding of the large model converted into vectors stored in the knowledge base, their own cue words and then combined with the vector library to match the ability to form a more excellent answer.

**1.2.2 Equipment management solutions in the context of Industry 4.0 abroad**

Driven by the wave of Industry 4.0, developed countries and regions in Europe and the United States continue to maintain a leading position in industrial equipment management solutions, and continue to introduce innovative technologies and products, leading the development direction of global industrial equipment monitoring and management systems. Germany, as the proposer of the concept of Industry 4.0, has always been at the forefront of industrial equipment monitoring and management. The German government, industry and academia work closely together to build a series of advanced equipment management solutions around core technologies such as CPS and Industrial Internet of Things. Siemens' MindSphere platform is one of the representative results, which seamlessly integrates industrial equipment with cloud computing to provide a full range of services from equipment connection, data collection, real-time monitoring to advanced analysis. The U.S. is also prominent in the field of industrial Internet, and industrial Internet solutions represented by General Electric's Predix platform and IBM's Watson IoT platform have been widely used in various industrial scenarios around the world. U.S. solutions usually have strong data analysis and artificial intelligence capabilities, and are able to mine valuable information from massive equipment data to realize intelligent diagnosis and predictive maintenance of equipment. Japan is vigorously developing equipment monitoring systems based on IoT technology on the basis of lean production and quality management. Japanese manufacturing enterprises generally adopt the strategy of “small but fine”, focusing on the practicality and stability of the system, and emphasizing the solution of actual production problems. Toyota, Fanuc and other companies to develop equipment monitoring system, especially focusing on the use of equipment efficiency, production beat and other key performance indicators of monitoring and optimization. Common features of advanced foreign solutions include: advanced system architecture, more open and more scalable, easy to integrate with other systems; high degree of standardization, support for a variety of industrial communication protocols; excellent user experience, providing an intuitive and easy-to-use interface and visualization tools; perfect security mechanisms, the use of multi-level security measures to protect the system and data security.

**1.2.3 Shortcomings of existing systems and innovations in this topic**

Although significant results have been achieved in the field of industrial equipment monitoring and management system at home and abroad, there are still a series of deficiencies in the existing system to meet the needs of intelligent manufacturing, and these problems provide innovative space and development direction for this study.

The main deficiencies of the existing system: high system complexity, high deployment and maintenance costs: most mainstream industrial equipment monitoring systems on the market today are large-scale integrated platforms with complex architectures that rely on numerous middleware and third-party components, resulting in a cumbersome system deployment process, high configuration requirements, and difficult maintenance. Especially for small and medium-sized manufacturing enterprises, the deployment and maintenance costs of such systems are often beyond the affordable range.

Limited Data Processing Capability: With the increase in the number and types of industrial equipment, the amount of data that the system needs to process is growing exponentially. Many of the existing systems in the face of multiple devices together to monitor and other scenarios, data processing capacity is obviously insufficient, resulting in data delays, loss and other problems, affecting the monitoring effect and user experience.

Insufficient scalability: The types of equipment and monitoring needs of industrial enterprises vary greatly, but many existing systems use a relatively closed architecture, making it difficult to carry out secondary development according to the specific needs of enterprises. This unchangeable architecture is difficult to meet the individual needs of different enterprises, limiting the value of the system.

Poor user interaction experience: traditional industrial software tends to focus on the realization of the function and ignore the user experience, the interface design is obsolete, the complexity of the operation process, resulting in high user learning costs, low efficiency. With the continuous improvement of user experience requirements, the user experience gap of industrial software is becoming more and more prominent, and has become an important factor restricting the application of the system.

Innovative points of this project: In view of the shortcomings of the existing system, this research proposes a solution for industrial equipment monitoring and management system based on lightweight Web architecture, with the following innovative points:

Lightweight architecture design: adopting lightweight technology stacks, such as Flask framework and MySQL database, to build a simple and efficient system architecture, which dramatically reduces the complexity of the system. The system is simple to deploy and has low maintenance cost, which is especially suitable for the actual needs and technical conditions of small and medium-sized manufacturing enterprises.

Efficient Data Processing Mechanism: Efficient data collection and processing processes are designed to realize real-time processing of multi-source data through Python scripts. The system passes the data to the database, and then through the database to the front-end, so that the system has a higher reliability.

Flexible modular design: The system adopts modular design, with little correlation between each functional module, which facilitates flexible configuration and expansion according to actual needs. Users can select the necessary functional modules according to their own needs, as well as allowing professional developers to develop new modules, while also providing convenience for future functional expansion.

Interface design focusing on user experience: The front-end of the system adopts modern Web technologies and design concepts to provide an intuitive and beautiful user interface and smooth interactive experience. The responsive design ensures that the system can get consistent user experience on different types and sizes of devices.

Intelligent Alarm and Data Visualization: The system implements multi-faceted alarm support, and also provides data visualization tools to support data export, helping users understand the device status more intuitively.

Comprehensive CPU utilization monitoring: The system especially strengthens the ability to monitor the CPU utilization of the equipment. By collecting and analyzing CPU utilization data in real time, the system load abnormality is detected in time, and the status of different CPU utilization is displayed intuitively in different colors.

Flexible data export function: The system provides data export function to meet users' data analysis and reporting needs.

Artificial Intelligence Utilization: The system integrates the open source big model, accesses deepseek interface, and then realizes the small assistant that simply fits the enterprise by adding fixed prompt words to the dify workflow, which can reduce the user's learning cost as well as the problem solving ability.

Through these innovative designs, this system aims to provide small and medium-sized manufacturing enterprises with an intelligent and high-performance equipment monitoring and management solutions to help companies achieve intelligent management of equipment at low cost, improve productivity and equipment reliability.

**1.3 Research objectives and content**

**1.3.1 Research objectives**

The overall goal of the research is to design and implement a lightweight Web architecture of intelligent factory production equipment information management system, to provide an intelligent and efficient equipment monitoring and management solution for manufacturing enterprises. Specific objectives:

Build a lightweight Web application system based on Flask framework and MySQL database to realize the core functions of basic equipment information management, operation status monitoring and historical data query.

Develop real-time temperature monitoring and CPU utilization monitoring modules, so that the data can be transmitted to the front-end in real time to help users find equipment abnormalities in time.

Develop spare parts management functions to realize inventory monitoring and in/out management of spare parts to ensure that the maintenance needs of production equipment can be met in a timely manner.

Design and realize the alarm mechanism, notify according to different rules, and notify relevant personnel in various ways to ensure that the problem can be dealt with in time.

Design the data export function to meet the enterprise's needs for data analysis and report generation.

Introduce artificial intelligence and develop a small assistant to enable users to solve problems in a simple way and meet the enterprise's demand for low-cost training of personnel.

**1.3.2 Content of the study**

In order to realize these functions, technical requirements are needed, then the next is the main technology I studied:

flask framework application research: study the application method of flask framework, learn how to use flask to build lightweight, high-performance web application system. Focus on the study of flask routing mechanism, template engine, extension system and other core components, and how to efficiently integrate with mysql database.

Real-time temperature monitoring: study the collection method and processing flow of equipment temperature data, develop temperature monitoring module, realize real-time collection, transmission, storage and display of temperature data. Realize temperature detection and alarm triggering mechanism to ensure that temperature abnormalities can be handled in a timely manner.

CPU utilization monitoring: study the monitoring method of equipment CPU utilization, develop CPU utilization monitoring module, and realize the collection, transmission and display of CPU data.

Alarm mechanism design: study the realization method of industrial equipment alarm system, design alarm mechanism, support different kinds of alarm processing based on temperature and CPU, such as page reminder and email reminder. The alarm information can be communicated to the relevant personnel in a timely manner and be handled properly.

Data export function realization: study the technical method of data export.

User Management Module: Realization of rights management, operator and administrator rights, restrictions on some important functions of the operator, as well as the administrator to modify the rights of the deletion of the function and so on.

Spare parts management: develop spare parts inventory monitoring and access management functions, realize inventory warning and replenishment reminders, so that the maintenance needs of production equipment can be met in a timely manner.

Artificial Intelligence: Explore how to invoke artificial intelligence and combine it with the system to realize the artificial intelligence of the system, so that the user's problem solving ability can be enhanced.

System Testing and Optimization: Design a comprehensive system testing plan to test and evaluate the system's functionality, performance stability and user experience. Based on the test results, re-system optimization is carried out to improve the stability and ease of use of the system.

Through the study of the above, a well-functioning, stable performance and easy-to-use intelligent factory production equipment information management system will be constructed to provide practical technical solutions for the intelligent transformation of manufacturing enterprises.

**Chapter II. System requirements analysis**

When building a management system, clarifying the requirements of the system is a critical step in ensuring a successful implementation. 图示

AI 生成的内容可能不正确。

*Figure 1. Functional structure diagram*

**2.1 Functional requirements**

**2.1.1 User management requirements**

**User Login:**

The system should provide a secure user login mechanism that supports user name and password authentication, and the login process should use encrypted transmission to prevent leakage of user information.

**Permission assignment:**

The system should assign different permissions according to the roles of users (e.g., administrator, operator); the administrator should have the highest permissions to manage users and view functions that cannot be viewed by the operator; the operator should have the permissions to monitor and manage the equipment, but not to view and modify the user configurations; the system should real-time monitor the online status of the users, and record the user's log-in time; the administrator can view the online status of all users to facilitate management and scheduling.

**2.1.2 Equipment Monitoring Requirements**

Capable of collecting real-time temperature data of the equipment and displaying it visually on the interface; when the temperature of the equipment exceeds the set safety threshold, the system shall automatically issue an alarm to notify the relevant personnel.

Real-time monitoring of the operating status of the equipment, including startup, stopping and other states; the system shall record the running time and downtime of the equipment for statistical analysis; when the equipment fails, the system shall automatically record the failure information and notify the maintenance personnel.

Capable of real-time acquisition and monitoring of CPU utilization of the equipment; the system should adopt different color identification according to different levels of CPU utilization to improve the intuition of monitoring; CPU utilization can be exported through the data to view the important information of the alarm to understand the current state of the equipment.

Provide data visualization, support temperature data and CPU utilization data export.

**2.1.3 Spare parts management requirements**

The inventory of spare parts should be monitored in real time, displaying the current inventory quantity and inventory status. When the inventory is lower than the set minimum inventory quantity, the system should show a reminder of insufficient inventory.

Provide spare parts in and out of the warehouse management function, record the time of each in and out of the warehouse.

**2.1.4 Alarm system requirements**

Support alarm mechanism based on different kinds of alarms, such as page prompts, email alarms. The system should provide a variety of alarm notification methods, including interface prompts, sound alarms and message push, etc. Different notification strategies should be used for different levels of alarms to ensure that important alarms can be handled in a timely manner, and the system should record the history of all alarms and support the statistics and analysis of alarm data.

**2.1.5 Data export requirements**

Support data export , you can export the information of temperature and CPU failure, in order to analyze the hidden danger.

**2.1.6 Artificial intelligence requirements**

Supporting the system's device and AI access enables a better understanding of the device's current status.

**2.2 Non-functional requirements**

**2.2.1 Reliability and Stability**

The system should be designed as a high availability architecture to ensure that the system remains stable over a long period of time and is also able to maintain data persistence and prevent data loss.

**2.2.2 Interface friendliness and responsiveness**

The system should provide an intuitive and easy-to-use user interface with a modernized design style to enhance the user experience, and the interface design should conform to the user's habits. It responds quickly to user operations, reduces waiting time and improves user satisfaction, and also needs to ensure that it can still run smoothly in the case of large data volumes.

**2.3 Technical feasibility analysis**

**2.3.1 Advantages of Flask Framework in Industrial Applications**

flask is a lightweight web framework , suitable for building small and medium-sized applications , can quickly respond to changes in business needs , it provides a flexible extension mechanism , developers can choose the appropriate extensions according to demand , such as databases , authentication , caching , etc. . Simple design , but also easy to learn and use , suitable for rapid development , reducing the cost of development and maintenance .

**2.3.2 Application of MySQL Database in Equipment Data Storage**

MySQL is a high-performance relational database that is capable of handling large amounts of data storage and querying, supporting complex query operations, providing a variety of data backup and recovery mechanisms to ensure the reliability and integrity of data, and is now widely used in a variety of application scenarios, with good community support and rich documentation resources.

**2.3.3 Realization scheme of front-end and back-end separation architecture**

Front-end and back-end separation architecture can improve development efficiency, front-end and back-end can be developed and deployed independently, reducing interdependence, front-end and back-end separation architecture makes the system easier to expand and maintain, the front-end can be more flexible to realize the dynamic interaction, improve user experience.

**Chapter III. System design**

**3.1 Overall architectural design**

The overall architecture design of the system determines the basic structure of the system and the relationship between the components. In order to realize an efficient and stable system, it was decided to use B/S architecture. This architecture model is suitable for modern web application development.

**3.1.1 B/S architecture design**

Front-end is mainly responsible for user interface display and user interaction. Html, css and Javascript technologies are used to realize the design and ensure that a good display effect is achieved. Front-end interface design follows the principles of user experience design to make the interface more beautiful and practical. Through the AJAX technology and the back-end data interaction, to achieve dynamic data updates and no-refresh page switching, the application of AJAX technology makes the user in the operation of the user does not need to frequently refresh the page, improve the user experience.

The back-end uses the Flask framework , responsible for the processing of business logic and data management.Flask's lightweight features make the back-end development more flexible and efficient.Flask framework provides a wealth of extensions and plug-ins to support rapid development and deployment. Data interaction with the front-end through the RESTful API provides a unified data interface.The RESTful API design follows the principle of resource orientation and uses HTTP verbs (GET, POST, PUT, DELETE) to perform operations.

The database uses MYSQL, which is responsible for data storage and management.The high performance of MYSQL ensures secure and fast access to data. The database is designed T to ensure data integrity and consistency. The database design includes user table, equipment table, CPU utilization table, temperature data table, spare parts table and fault record table,mail table and so on.

**3.1.2 System Module Division and Interaction Relationships**

The system is divided into modules based on functional requirements, with each module responsible for specific functions. Modules interact with each other through interfaces to ensure the flexibility and scalability of the system.

User management module: responsible for user registration, login, rights management and online status monitoring. Administrator can pass to monitor online users and manage users. User management module is one of the core modules of the system to ensure the security of the system and the legitimacy of users.

Equipment monitoring module: responsible for real-time monitoring, temperature collection, CPU utilization monitoring and status tracking of equipment. Equipment monitoring module is the key module of the system to ensure the normal operation of equipment and timely detection of faults. It supports real-time monitoring of equipment status.

Alarm system module: responsible for automatically triggering alarms according to preset thresholds and rules. Alarm system is an important guarantee for the safe operation of the equipment to ensure that abnormal situations can be dealt with in time. It provides the configuration of alarm rules, the query of alarm records and the interface of alarm processing.

Spare parts management module: It is responsible for inventory monitoring and in/out management of spare parts. Spare parts management module ensures sufficient supply of spare parts in the production process and prevents production interruptions caused by spare parts shortage. It supports real-time inventory monitoring and management of incoming and outgoing records.

Data export module: responsible for system data export and report generation. Temperature and CPU data can be selected. The data export module meets the enterprise's demand for data analysis and report generation.

Intelligent assistant module: the module plans to integrate AI big language modeling technology to provide users with an intelligent interactive interface. The small assistant will support system operation guidelines and troubleshooting, and realize the popularized interpretation of alarm information and processing suggestions The module adopts a chat-style interface, which does not need to memorize complex commands. It is planned to access deepseek's large model api to realize. The module is designed to lower the threshold of system use so that non-technical personnel can also efficiently complete equipment monitoring and troubleshooting work. The implementation reduces training costs and improves overall system usability.

**3.2 Power module design**

**3.2.1 User management module**

The authentication mechanism uses JWT (JSON Web Token) for user authentication to ensure the security of the user's identity.JWT is an open standard based on JSON, which is used to pass statements between web application environments. After the user has successfully logged in, the system generates a JWT and returns it to the front-end, which carries it in subsequent requests for authentication. the use of JWT simplifies the process of user authentication and improves the security of the system.

In terms of privilege control, the system assigns different privileges according to user roles to ensure that users can only access and operate resources within their privileges. Privilege control is an important part of system security to prevent unauthorized access.

**3.2.2 Equipment Monitoring Module**

For temperature collection, the temperature data of the equipment is collected at regular intervals through a Python script and sent to the back end for storage and processing. Real-time collection and monitoring of temperature data are important guarantees for the safe operation of equipment.

CPU monitoring: Real-time acquisition of CPU utilization data of devices through dedicated apis, supporting parallel monitoring of multiple devices. The front end visually displays the CPU status of different levels through color coding: green (<70%) indicates normal, yellow (70%-90%) indicates warning, and red (>90%) indicates serious warning.

The front end displays the real-time status of the device through charts and dashboards, helping users intuitively understand the operation of the device. The visual display of data improves the decision-making efficiency of users.

**3.2.3 Spare Parts Management Module**

Inventory Alert, the system automatically monitors the inventory of spare parts based on a set inventory threshold. The inventory warning function helps companies to replenish spare parts in time to prevent production interruptions. Users can set and adjust the inventory thresholds in the inventory creation category.

Incoming and outgoing process, provides spare parts in and outgoing management function, records the detailed information of each in and out of the warehouse. The record includes the name, quantity, time and other information of the spare parts.

**3.2.4 Fault logging module**

Fault types, the system supports recording of multiple fault types, such as high temperature, excessive CPU load, etc.

**3.2.5 Alarm System Module**

Alarm triggering mechanism, the system automatically triggers alarms based on preset thresholds, including temperature overruns, excessive CPU utilization, and other types of alarms. Alarm levels are divided into three levels: information alert (green), warning (yellow) and serious warning (red), corresponding to different degrees of urgency. After the alarm is triggered, the system automatically records the alarm event, including the alarm time, alarm type, and device name.

Alarm notification and processing, the system provides a variety of alarm notification methods to ensure that users can receive timely alarm information. Alarm information is displayed in a dedicated area of the system interface and uses eye-catching visuals to draw the user's attention. Users can also use email alerts and fill out their own emails to get alarm information. Alarms can be acknowledged and processed by the user.

**3.2.6 Data export module**

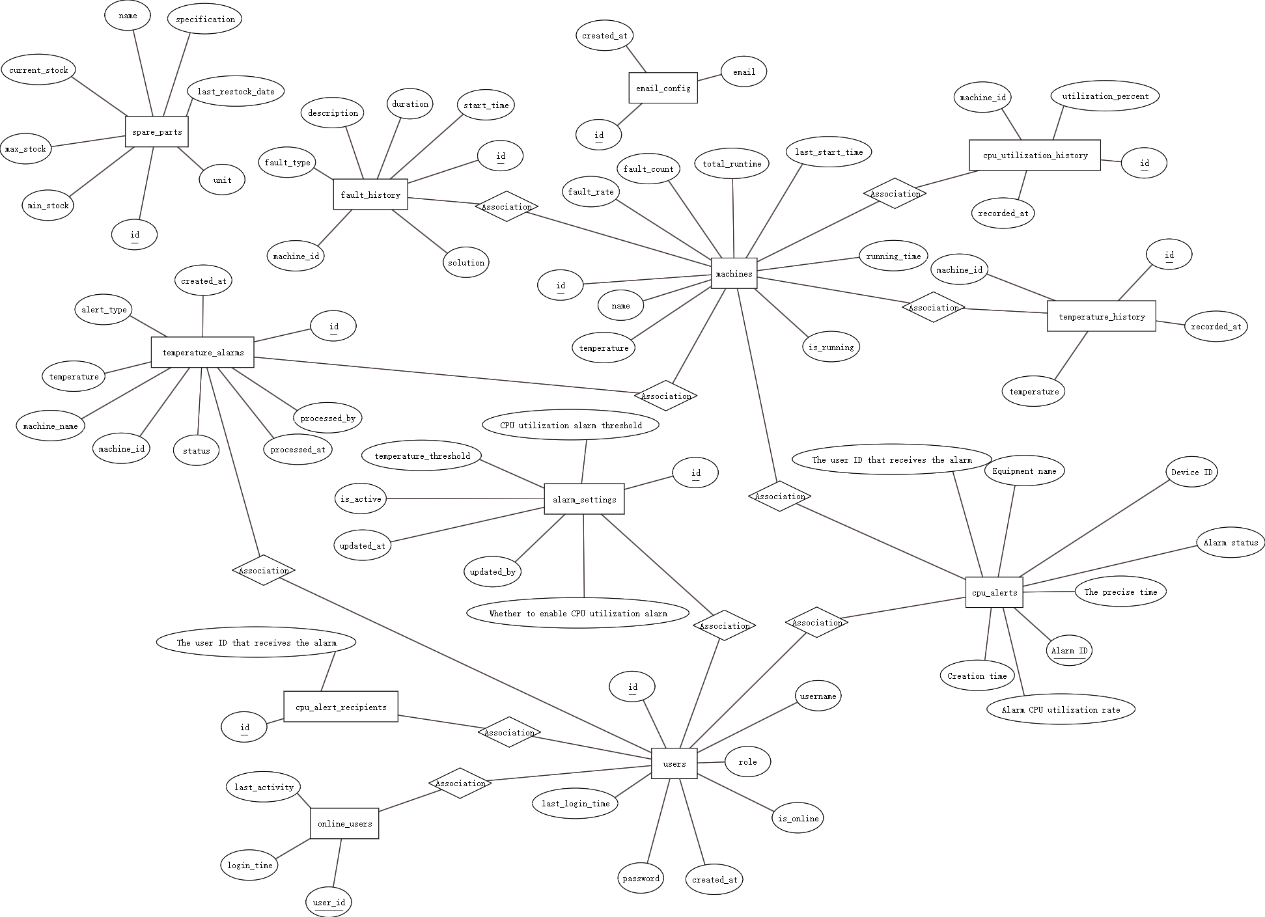
The system provides flexible data export function, support for exporting data and more visualization of information.

**3.3 Database design**

Database design is an important part of system design and determines how data is stored and managed. Good database design improves system performance and data security.

**3.3.1 E-R diagrams and relationship models**

The E-R diagram of the system shows the entities and their relationships, including users, devices, CPU utilization, temperature data, spare parts, fault logs, alarm logs, etc. The E-R diagram is the basis of the database design, which helps the developer to understand the data structure and relationships of the system.



*Figure 2. Database ER diagram*

**3.3.2 Core table structure design**

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

*Figure 3. Database table*

users table: stores user information, including user name, password, role, creation time. Passwords are encrypted to ensure the security of user information.

machines table: stores device information, including device name, type, status, and number of failures.

cpu\_utilization table: stores CPU utilization data of devices, including device ID, collection time, utilization value, and data source.

temperature\_data table: store the temperature data of the device, including device ID, collection time, temperature value, and data source.

spare\_parts table: store spare parts information, including spare parts name, specification, stock quantity, minimum stock.

fault\_history table: store fault history information, including fault type, description, occurrence time, and solution.

alarm\_records table: store the alarm records of the system, including alarm time, alarm type, alarm level, equipment information, processing status.

email\_config table: alarm mailbox information, including email, creation time.

**3.3.3 Data Integrity and Index Optimization**

The system ensures data integrity and consistency through mechanisms such as foreign key constraints and uniqueness constraints. Data integrity is an important principle of database design to ensure data reliability and security. Database design follows the principle of normalization to avoid data redundancy and inconsistency. The normalized design improves data reliability and maintainability.

The system indexes frequently used query fields to improve query performance. The design of indexes needs to balance query performance and storage space to avoid too many indexes affecting data update performance. The optimization of indexes improves the query efficiency and response speed of the system, which can run efficiently.

**3.4 Interface design**

**3.4.1 RESTful API Design Principles**

The API design is resource-centered, with each resource corresponding to a URI that operates using HTTP verbs (GET, POST, PUT, DELETE). The resource-oriented design improves the readability and consistency of the API. The resource design needs to consider the frequency of data access and query performance.The API design follows the principle of statelessness, where each request is independent and does not depend on the state of the previous request.The API design provides a uniform interface style, and also needs to consider the data transfer and storage to allow the API to operate efficiently.

**3.4.2 Front-end and back-end data interaction process**

Data requests, the front-end sends data requests to the back-end through AJAX to get the required data. The design of the data request needs to consider the transmission and storage of data.

Data response, the back-end receives the request, data processing and business logic execution. The design of data response needs to consider data transmission and storage. The back-end returns the processing results to the front-end in JSON format, and the front-end displays and interacts according to the returned data.

For error handling, the system will provide a perfect error handling mechanism to ensure timely feedback of error information when the request fails. The design of error handling needs to consider the transmission and storage of data. The front-end carries out corresponding processing according to the error information, such as prompting the user to retry or contact the administrator.

**Chapter IV. System implementation**

**4.1 Technology stack**

**4.1.1 Data acquisition realization**

The system collects temperature data and CPU utilization data of the device regularly through Python script and sends the data to the backend for storage and processing. Real-time data collection and monitoring is an important guarantee for the safe operation of the equipment.Python script supports multiple data collection methods, such as sensors, API calls. Different collection methods are suitable for different application scenarios to ensure the accuracy and reliability of data.

The system supports multi-source data acquisition, which can obtain temperature data and CPU utilization data from different sensors and devices. Multi-source data acquisition improves data accuracy. The frequency and mode of data acquisition can be configured according to actual needs.

**4.2 Core functionality realization**

**4.2.1 User Authentication and Privilege Management Implementation**

The system provides user registration and login functions to ensure that only authorized users can access the system. During user registration, the system verifies and stores user information. When a user logs in, the system verifies the correctness of the user name and password, and generates a JWT to be returned to the front-end. the use of JWT simplifies the process of user authentication and improves the security of the system.

The system assigns different permissions according to user roles to ensure that users can only access and operate resources within their permissions. Administrators can look at the module Online Users. operators will get a prompt that they do not have permissions. The management of user roles and permissions can be configured through the system interface.图形用户界面, 应用程序

AI 生成的内容可能不正确。

*Figure 4. Admin*图形用户界面, 应用程序

AI 生成的内容可能不正确。

*Figure 5. Operator*

The operator will be prompted with no privileges

**4.2.2 Device Monitoring and Status Management Implementation**

The system collects temperature data from the device at regular intervals through Python scripts and sends the data to the backend for storage and processing. Temperature is monitored through -openhardwaremonitor and the database is designed to support storage and querying of temperature data.

图形用户界面

AI 生成的内容可能不正确。

*Figure 6. Temperature monitoring software*

The system collects real-time CPU utilization data of the device through a dedicated API interface. The collected data is stored in the database after preliminary processing to improve data access efficiency.

Both temperature and CPU keep updated information through high-frequency query of the machine's status, which is a timely information about the machine for the workforce.

The system real-time monitoring of the operating status of the equipment, including start, stop, fault and other states. Real-time monitoring of equipment status improves the safety and reliability of the equipment. The front-end shows the real-time status of the equipment through charts and graphs to help users visualize the operation of the equipment.

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

*Figure 7. Equipment monitoring page*

**4.2.3 Spare parts inventory management realization**

The system monitors the inventory of spare parts in real time, displaying the current inventory quantity and inventory status. When the inventory is lower than the set minimum stock level, the system automatically sends replenishment reminders. Users can set and adjust the inventory threshold in the system to ensure the flexibility and adaptability of inventory management.

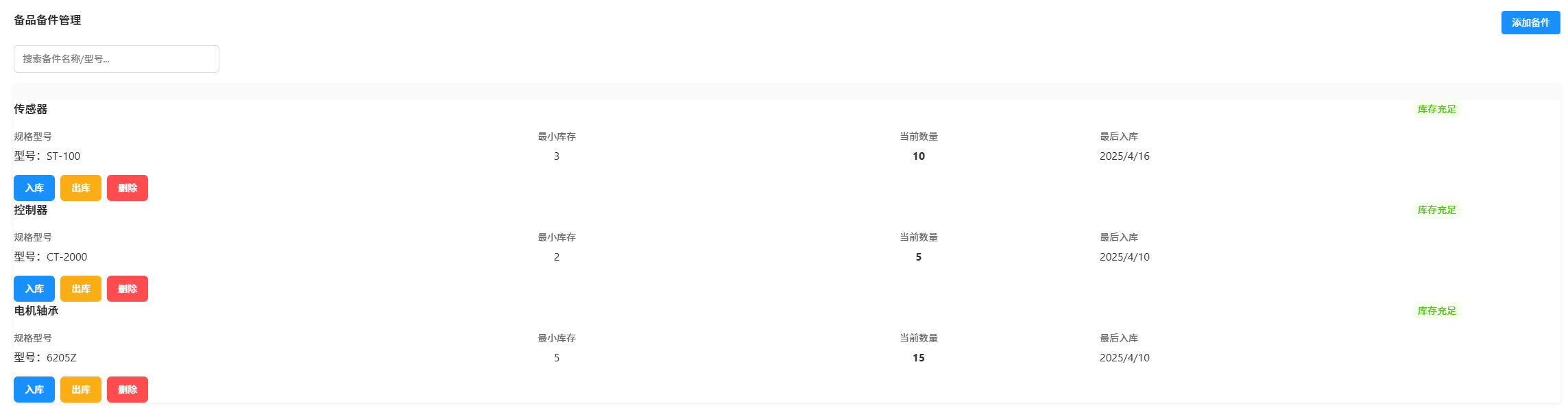
The system provides in/out management function for spare parts, recording the detailed information of each in/out record. The records include spare parts name, quantity, time, operator and other information.

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

*Figure 8. Inventory inbound page*

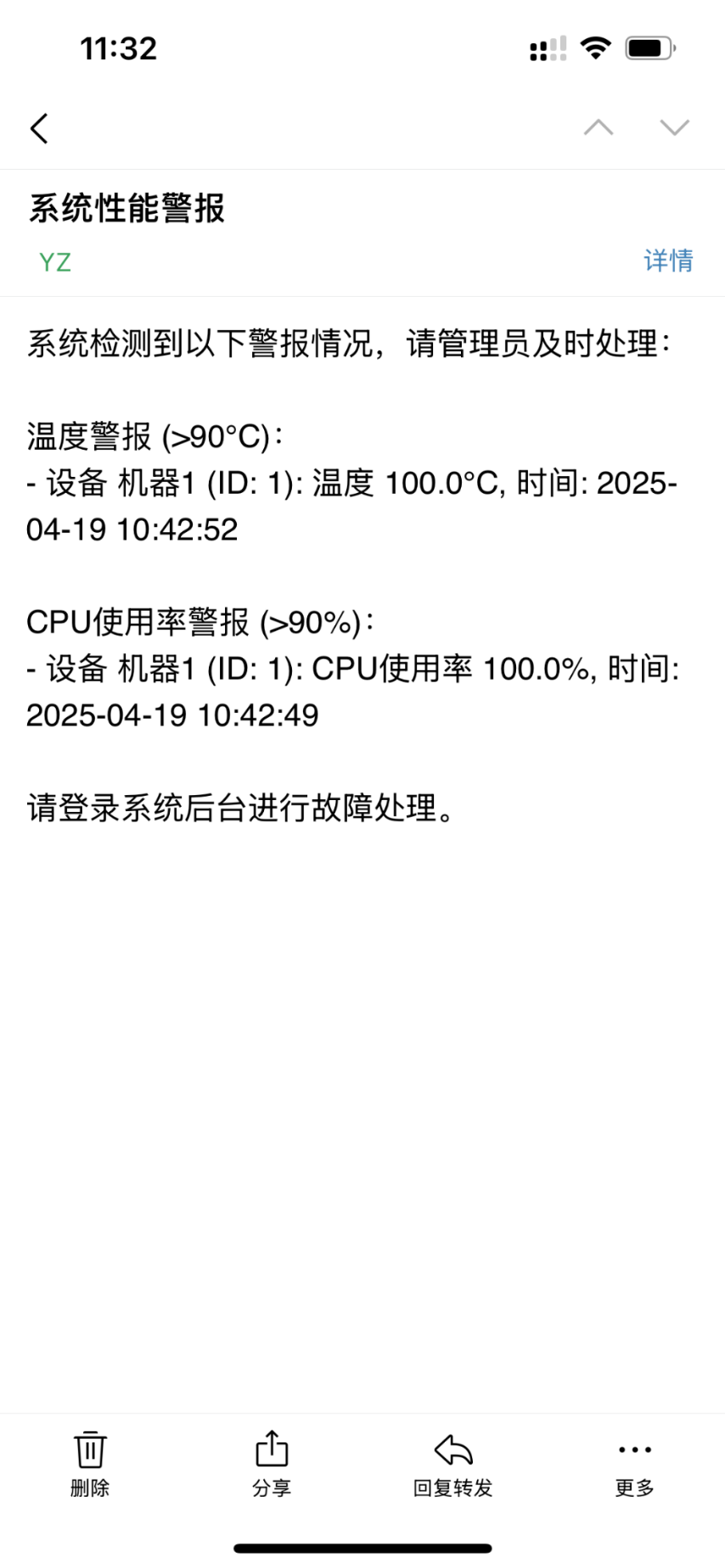
Users can query and export in/out records through the system to ensure transparency and traceability of inventory management.



*Figure 9. Spare parts Management page*

**4.2.4 Alarm system realization**

System through a variety of monitoring data (temperature, CPU utilization, etc.) to achieve the automatic alarm function alarm triggered, the system through the front-end interface prompts and e-mail alarms and other ways to notify the user, so that the alarm information is delivered in a timely manner.



*Figure 10. Email prompt*

Provide alarm management interface, support users to view the alarm information and fill in the mailbox to receive the alarm in time, and realize the alarm history record function. Users need to click on the processing before receiving the next alarm prompts in the interface, mailboxes broadcast alarms every thirty minutes, by querying the judgment database, if there is no data over the threshold for thirty minutes, then no alarm. Users can analyze machine problems based on the frequency of alarms.

图形用户界面, 应用程序

AI 生成的内容可能不正确。

*Figure 11. Email import*

**4.2.5 Data export function realization**

The system is designed with an Excel export format, which can support the export of temperature and CPU information, and the exported information is the time and number of times when the temperature and CPU reach the threshold, which realizes the function of automatic report generation.

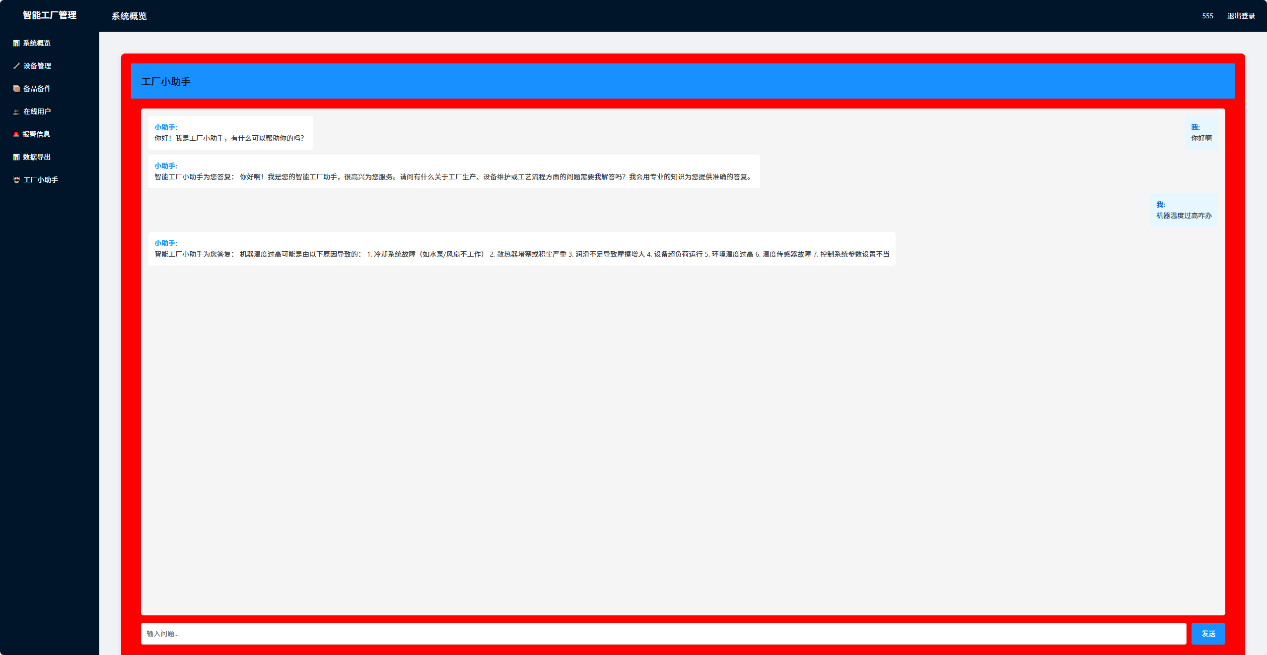
**4.2.6 Smart Factory Assistant**

Intelligent factory small assistant is the system integration of artificial intelligence interaction interface, based on large-scale language model technology, my intelligent factory small assistant is built through the Dify workflow, and add fixed cue words, so that the assistant is more in line with the discriminatory nature of the assistant, has realized the intelligent dialogue function, and the follow-up can be completed through the RAG technology as well as fine-tuning the model of the deeper fit for the enterprise. The small assistant can provide users with intelligent auxiliary services, which can interpret alarm information and provide basic processing suggestions in an easy-to-understand way. It greatly reduces the threshold of system use, so that non-technical personnel can also conveniently carry out equipment monitoring and troubleshooting.

电脑软件截图

AI 生成的内容可能不正确。

*Figure 12. dify workflow*



*Figure 13. Factory Assistant page*

**4.3 System Deployment**

**4.3.1 Deployment Architecture**

Native deployment architecture, running all the components on Windows operating system, including web server, application services and database. This deployment method will be simpler and more direct, suitable for small applications, and can quickly start the whole system. web services, using Flask built-in development server to provide web services, easy to develop and debug. Database, install MySQL database locally to store all the data of the system. Data collection, use Python script to collect device data at regular intervals and store the data into the database. Big model, realized by building the dedify workflow and calling the api interface.

Network configuration, run in local network environment, accessed through localhost port 3001 address.

**4.3.2 Windows environment deployment process**

The first step is environment preparation by installing the Python environment on the Windows operating system. Install MySQL database service and create the database and users required for the system. Install the necessary Python packages, such as Flask, PyMySQL, etc. You can use the pip command to install all dependencies with one click.

Then comes the code deployment, copy the system code to a local directory. Configure the database connection parameters, modify the database connection information in the app.py file to ensure that the backend can connect to the MySQL database correctly. Execute the database initialization script to create the necessary table structures and initial data.

Run the service and start the application using the Python command: python app.py. Start the data collection scripts monitor\_temp.py and cpu\_monitor.py. You can set up a timed task through the Windows Task Planner program to execute the data collection scripts periodically.

Open a browser and visit http://localhost:3001 to verify that the system is working properly. Test the functions to make sure that the various modules of the system are working properly.

**4.3.3 System maintenance**

Data backup, using MySQL's own backup tool or third-party backup software, you can backup your database regularly. Keep multiple versions of backups to ensure timely recovery in case of data loss or corruption.

System updates, regularly update system code and dependent libraries to fix known vulnerabilities and issues. Take a full backup before updating to ensure that you can roll back to a previous version if problems occur during the update process.

**4.3.4 Key Technical Difficulties Solved**

The challenge of multi-source data collection is that the system has to take data from several places, such as device sensors and API interfaces. The data format is different in each place. Solution: design the database into the same structure, and then write a unified data collection interface, which is much easier to deal with.

Difficulties in AI application, when calling the large language model through API, the waiting time is long, which affects the user interaction experience, especially in the case of network fluctuation is more obvious. Subsequently, local deployment will be added to solve the problem in a faster way, which is safer and faster. The future will also be involved in the company's security issues, so the subsequent large model will be transferred to local deployment. dify workflows are emerging, the subsequent will have more put the case, as well as can use RAG technology and fine-tune the large model to make the large model more adaptable to the company, made a set of suitable for the company's workflow, to achieve the ai intelligent.

The logical design of the mail alarm and the implementation of the logic of the mail alarm and the implementation of the technology is a little difficult, the initial design of the mail function and the main application coupled, always interface program can not run, the follow-up is to re-open a new program, timed execution before the solution, there is also the design of the logic, for example, you need to query the database for the existing ultra-threshold data, and then through the time to determine the time not more than 30min to send a mail, and if the ultra-threshold data is not more than 30min, then send a mail. send mail, and if too much over the threshold, the mail will be unlimited, so add a rule is the number of times the mailbox is sent is also 30min, so the problem is solved.

**Chapter V. System testing**

**5.1 Test environment and methodology**

The choice of test environment and methodology directly affects the effectiveness and comprehensiveness of the test. In this section I will introduce the system's test environment and test methods in detail.

**5.1.1 Test environment configuration**

**Hardware environment**：

Computer: E5 2686 V4 processor, 64G memory, 512GB SSD

Monitor: 1920x1080 resolution

**Software environment**：

Operating System: Windows 10 Professional Edition

Python: Version 3.8.5

MySQL: Version 9.1.0

Browsers: Chrome 136, Firefox 133, Edge

**5.1.2 Test methods**

Unit testing, using the pytest framework to unit test the various modules and functions of the system to verify the correctness of the functionality. The unit test covers the core business logic, data processing and interface functions to ensure the correctness of each module.

Interface testing, I chose Apifox to test the system's API interface to verify the correctness and stability of the interface. Interface testing includes parameter validation, return value verification, exception handling and other aspects to ensure the robustness and reliability of the interface.

Functional testing, testing of various functional modules of the system to verify the correctness and completeness of the function. Functional testing is to verify the usability of the system by simulating real user operations and usage scenarios. First of all, I need to write use cases, using the equivalent class division method, scenario method and so on to complete.

Performance testing, using JMeter and packet capture to simulate real user scripts for system performance testing, to assess the system under different loads of response time and resource usage. Considering that the practicality of deploying to the local area is not high, we choose not to carry out performance testing.

Compatibility testing, testing the system on different browsers and devices to verify the compatibility and consistency of the system. The compatibility test includes aspects such as different browsers, different screen resolutions and different operating systems to ensure the usability of the system in various environments.

**5.2 Test Case Design**

**5.2.1 Functional Test Cases**

**User Management Test Cases:**

User Registration: Write use cases by entering registered and unregistered user names.

User Login: Verify the user login function by matching the registered and unregistered username and password.

**Device Monitoring Test Cases:**

Device Status Query: Verify the device status query function, including online status, running status and so on.

Temperature Data Query: Observe whether the temperature is updated in real time by turning on the detection and turning off the detection.

CPU utilization query: Same as temperature, turn on detection and off detection to observe whether the CPU is updated in real time.

**Spare parts management test cases:**

Inventory Lookup: Test the inventory lookup function by searching for words related to the inventory name.

Spare Parts Add-In: Add spare parts to stock to test if they were added successfully.

Outbound and Inbound Operations: Test the Outbound and Inbound function, observe whether the quantity is successfully reduced or increased.

Inventory Alert: Verify the Inventory Alert function, insufficient inventory to see if it will be prompted.

**Alarm system test cases:**

Alarm Trigger: Verify the alarm trigger function, including temperature overrun alarm and CPU utilization overrun alarm.

Alarm Notification: Alarm notification function, including interface prompts and email notification.

Alarm Email Input: Test whether the alarm email is successfully input.

**Data export test cases:**

Data Export: Data export function, observe whether the book is successfully exported.

Report Generation: Test the report generation function and observe whether the temperature and CPU data are correct.

**5.2.2 Performance testing**

Response Time Test: Test the response time of each page and function of the system to ensure a good user experience.

Concurrent User Test: Simulate multiple users accessing the system at the same time to test the concurrent processing capability of the system.

Under the Windows environment, it mainly tests the performance of the system under 5-20 concurrent users.

**5.2.3 Compatibility test cases**

Browser compatibility testing: Test system functionality on mainstream browsers such as Chrome, Firefox, Edge, and so on. Verify the consistency of page layout, interactive functions and data presentation in different browsers.

Resolution compatibility test: Test the system interface under different screen resolutions (1366x768, 1920x1080, etc.).

**5.3 Test results and analysis**

**5.3.1 Functional test results**

**Users manage test results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| User registration | Users register by entering a username and password they haven't used before | The user successfully registered and returned to the login page | Pass |
| User registration | Users register with the registered usernames | User registration failed. | Pass |
| User login | Users log in with the registered username and password | Users log in with the correct registered username and password | Pass |
| User login | The user logs in with the username and wrong password registered | User login failed. | Pass |
| User login | The user logs in with an unregistered username | User login failed. | Pass |

**Device monitoring test results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| Equipment status | Test the equipment to be tested and then observe whether the front end is online | The front-end display device is running | Pass |
| Temperature data query | Log in to the monitoring device, enable the temperature monitoring, and observe whether the temperature is updated in real time | Real-time update of temperature | Pass |
| Temperature data query | Detect a device without temperature monitoring enabled and observe whether the temperature is updated | The temperature is not updated in real time | Pass |
| CPU utilization query | Log in to the monitoring device, enable the CPU utilization monitoring, and observe whether the CPU is updated in real time | Real-time update of CPU | Pass |
| CPU utilization query | Check one that has not enabled CPU detection and observe whether the CPU updates in real time | The CPU does not update in real time | Pass |

**Spare parts management test results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| Inventory inquiry | Search for words related to the inventory name | The corresponding structure appears | Pass |
| Item addition inventory | Enter the name, specification model, current inventory, and minimum inventory | The spare parts have been added successfully | Pass |
| Outbound and inbound operations | Enter the quantity of spare parts that need to be warehoused | The current inventory quantity has successfully increased according to the input quantity | Pass |
| Outbound and inbound operations | Enter the quantity of spare parts that need to be dispatched from the warehouse | The current inventory quantity has been successfully reduced based on the input quantity | Pass |
| Inventory emergency function | Keep the current inventory lower than the minimum inventory | Prompt insufficient inventory | Pass |

**Alarm system test results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| Alarm triggering function | When a machine has no alarms to handle, bring the detection machine (such as temperature and CPU) to the detection threshold | An alarm appears on the page. | Pass |
| Alarm notification function | Adjust the detection machine (such as temperature, CPU) to the detection threshold and observe whether the mailbox has received the email | The email received the detailed information of the corresponding alarm | Pass |
| Alarm email entry | Enter the alarm emails and then conduct email tests to observe if there are any emails | If an email is sent, it indicates that the entry is successful | Pass |

**Data export test results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| Report export function | Click on the data export of CPU and temperature, and observe whether there is an excel file exported | The data was successfully exported. | Pass |
| Data generation function | Observe the data inside after successfully exporting the data | The data is intact. | Pass |

**Factory mini-assistant test:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test items | Test steps | Expected result | Test result |
| The test assistant answered | The input machine temperature is too high or there is a CPU problem | The little assistant answers multiple possibilities to solve the problem | Pass |

**5.3.2Performance Test Results**

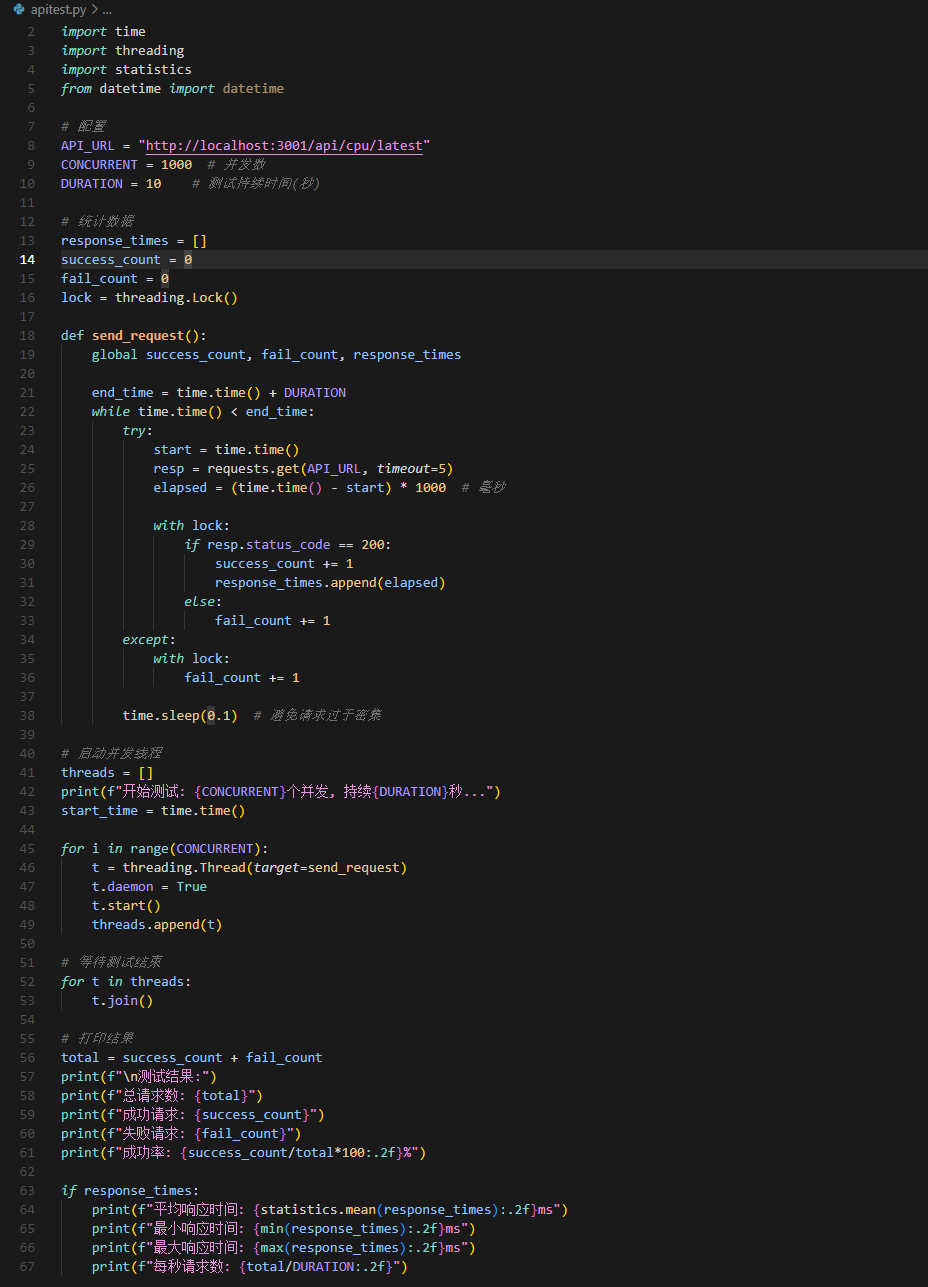
First perform interface testing, then concurrency testing through this interface

图形用户界面, 文本, 应用程序, Teams

AI 生成的内容可能不正确。

*Figure 14. Interface return*

Then concurrency test limit concurrency number via py script, you can choose the number of concurrency and observe the success rate



*Figure 15. Automated test script code*

Try the concurrent numbers of 5, 10, 15 and 20 for testing, and observe the success rate and the average response time  
5:

Total number of requests: 385

Successful request: 383

Failed request: 2

Success rate: 99.48%

Average response time: 29.58ms

Minimum response time: 10.98ms

Maximum response time: 53.94ms

Requests per second: 38.50

文本

AI 生成的内容可能不正确。

10:

Total number of requests: 763

Successful request: 682

Failed request: 81

Success rate: 89.38%

Average response time: 30.05ms

Minimum response time: 11.99ms

Maximum response time: 64.18ms

Requests per second: 76.30

文本

AI 生成的内容可能不正确。

15:

Total number of requests: 1141

Successful request: 955

Failed request: 186

Success rate: 83.70%

Average response time: 29.49ms

Minimum response time: 10.41ms

Maximum response time: 88.93ms

Requests per second: 114.10

文本

AI 生成的内容可能不正确。

20:

Total number of requests: 1,484

Successful request: 1175

Failed request: 309

Success rate: 79.18%

Average response time: 31.75ms

Minimum response time: 12.00ms

Maximum response time: 132.17ms

Requests per second: 148.40

文本

AI 生成的内容可能不正确。

Conclusion: When the concurrent number exceeds 20, the success rate is lower than 80. It is recommended that the number of users online simultaneously does not exceed 20

Resource usage test results: CPU usage: 5% on average during normal operation, with a peak of 20% during high load.

**5.3.3 Compatibility test results**

Browser compatibility test results:

Chrome: all functions are normal, the interface display is consistent.

Firefox: all functions are normal, interface display is consistent.

Edge Browser: Functions normally, interface display is consistent.

Conclusion: The system has good compatibility with mainstream modern browsers.

Resolution compatibility test results:

1920x1080 resolution: interface layout is reasonable, all elements display complete.

1366x768 resolution: interface layout is reasonable, all elements display similar to 1920x1080.

Conclusion: The front-end design of the system is able to adapt to different screen resolutions.

**Chapter VI. Systematic evaluation and outlook**

**6.1Advantages and disadvantages of the system**

**6.1.1 System advantages**

This chapter summarizes and evaluates the results of the system's implementation, analyzes the strengths and weaknesses of the system, and looks forward to future directions. The following makes a personal evaluation of the system as well as a summary.

Functional integrity, the system realizes the equipment monitoring, spare parts management, fault records, alarm management, data export and factory assistant and other functions, to meet the important needs of intelligent factory production equipment information management. The innovation point lies in the factory assistant, keep up with the development of the times, access ai big model to adapt to the development of intelligent factory.

User experience, the system interface design is brief, the operation process is relatively clear, the learning cost is low. The system responds quickly and the interface is beautiful, providing a good user experience.

System performance, in the Windows environment, the system runs stably, responds quickly, and occupies a reasonable amount of resources. The system is able to handle medium-sized data volume and meet the needs of small manufacturing enterprises.

System flexibility, the system adopts modular design, each functional module is relatively independent, easy to maintain and expand. The system provides rich configuration options, allowing users to adjust the system behavior according to actual needs.

Data visualization, the system provides rich data visualization functions, including real-time monitoring charts and reports on statistical failures.

Adding the artificial intelligence module, the system accesses the open source big model, according to the fixed prompt words, the user can ask related simple questions to make the problem more low-cost solution.

**6.1.2 System shortcomings**

Deployment limitations, the system is currently tested and deployed only in the Windows environment, and cross-platform compatibility is yet to be verified. Stand-alone deployment mode limits the scalability and high availability of the system. The system can be deployed on the cloud, which is more convenient and reduces many restrictions, such as power off and power off.

The performance of the system decreases significantly when dealing with large-scale data, and further optimization of data processing and query algorithms is needed. The execution efficiency of complex statistics and analysis functions needs to be improved.

The security functions of the system are relatively basic and easily breached, and stronger security protection may be needed in enterprise-level application scenarios.

The system has limited integration capability with other enterprise information systems (e.g. ERP, MES, etc.). The lack of standardized data exchange interfaces limits the application of the system in complex enterprise environments.

**6.2 Future prospects and functional extensions**

Introducing machine learning algorithms, such as using historical data that has been collected such as temperature and CPU usage, to train simple prediction models, such as the ability to send timely alerts when there are abnormalities in the temperature or CPU, and to realize equipment failure prediction and preventive maintenance functions. More advanced data analysis tools can also be developed to provide in-depth productivity analysis and optimization recommendations.

In the future, a supporting mobile version can be developed, which can be accessed and operated by mobile devices. Realize mobile alarm push and remote control functions to improve system availability and response speed.

Subsequent increase in the camera and video monitoring functions, more intuitive monitoring of the state of the equipment, can also support voice remote control and voice prompts function, call the field workers to operate.

AI knowledge base system, you can pass to the training fine-tuning suitable for their own large model, and then add a RAG technology and their own company's knowledge base to complete a suitable for their own company's assistant, more relevant as well as to prevent information leakage. Summarize and summarize the existing problems to the vector knowledge base, the problem comes more people can respond in time.

**6.2.1 Technological upgrading**

For architecture optimization, the system is transformed into a microservice architecture to improve the scalability and maintainability of the system; support for cloud deployment mode to achieve cross-platform deployment and elastic expansion of the system; implementation of data slicing and partitioning technology to optimize data storage and query performance.

Enabling security enhancement to achieve better authentication and authorization mechanisms, such as two-factor authentication, role fine-grained authority control, etc.; strong data encryption and transmission security measures to protect the security of sensitive data.

Integration capability enhancement, develop standardized API interfaces to support seamless integration with other enterprise systems; introduce industrial IoT standard protocol support to achieve compatibility with more types of equipment.

Smart Factory Panorama Solution extends system functions to cover more aspects of smart factories, such as energy management, environmental monitoring, quality control, etc. Build a smart factory digital twin platform to provide comprehensive factory visualization and management functions.

**Conclusion**

This paper describes in detail the process of designing and realizing the production equipment information management system for smart factories. The system is developed based on Python and Flask framework, and uses MySQL database to store data, realizing the functions of equipment monitoring, spare parts management, fault records, alarm management and data export. The system adopts modularized design, simple and intuitive interface, and clear operation flow, which meets the basic needs of small manufacturing enterprises for equipment information management. The functional integrity, performance performance and compatibility of the system are verified through functional testing, performance testing and compatibility testing of the system. The test results show that the system runs stably in the Windows environment, with fast response speed and reasonable resource consumption, and can meet the expected application requirements. At the same time, the test also identified the system's deficiencies in large data volume processing, security and integration capability, which provided directions for the system's subsequent optimization and upgrading. In the future development, the system will further improve its functions, performance, security and integration capability to adapt to a wider range of application scenarios and user needs. By introducing new technologies such as artificial intelligence, cloud computing and big data, the system will develop in the direction of intelligence, networking and servicing to provide users with a more comprehensive and efficient smart factory equipment management solution. The design and realization of this system reflects the value of information technology application in the industrial field, provides an effective tool for manufacturing enterprises to realize intelligent transformation, and is of positive significance for improving equipment management efficiency, reducing maintenance costs and enhancing productivity. With the continuous improvement and popularization of the system, it will provide support for the intelligent transformation and high-quality development of more manufacturing enterprises.

**Acknowledgements**

Time flies, four years of undergraduate career is coming to an end. When I look back on this period of study, my heart is full of gratitude.

First of all, I would like to express my heartfelt thanks to Prof. Li Yunfa and Prof. Belozubov Aleksandr for their careful guidance. I would like to thank you for your patience and valuable advice during the thesis writing process, which enabled me to successfully complete this research.

At the same time, I would like to thank my alma mater for the four-year training, which not only provides us with high-quality teaching resources, but also creates a free and open academic atmosphere. Every course, every academic lecture, every teamwork here has made me grow continuously.

Finally, I would like to deeply thank my parents. You have always been my most solid support, giving me unconditional support and encouragement in good times and bad. Your love and dedication have been my greatest motivation along the way.

In the future, I will continue to move forward with this gratitude and live up to my expectations.

Thank you for supporting me!

**References**

1. Cheng G J, Liu L T, Qiang X J, et al. Industry 4.0 development and application of intelligent manufacturing[C]//2016 international conference on information system and artificial intelligence (ISAI). IEEE, 2016: 407-410.
2. Shafiq S I, Velez G, Toro C, et al. Designing intelligent factory: conceptual framework and empirical validation[J]. Procedia Computer Science, 2016, 96: 1801-1808.
3. Hozdić E. Smart factory for industry 4.0: a review[J]. International Journal of Modern Manufacturing Technologies, 2015, 7(1): 28-35.
4. Jiao Hongshuo, Lu Jianxia. A review of the current status of research on smart factories and their key technologies[J]. Journal of Mechanical &amp; Electrical Engineering, 2018, 35(12).
5. Zhang Shu. Industry 4.0 and smart manufacturing[J]. Mechanical Design and Manufacturing Engineering, 2014, 43(8): 1-5.
6. Feng Yiping, Rong Gang. Experimental system of smart factory for process industry[J]. Information and Control, 2005, 34(1): 35-39, 43.
7. Du Pingsheng. Smart factory - Germany's first step in Industry 4.0 strategy[J]. Automation Expo, 2014 (1): 22-25.
8. Miao Xueqin. Smart factory and equipment manufacturing transformation and upgrading[J]. Automation Instrumentation, 2014, 35(3): 1-6.
9. Hua Rong. Smart factory of the future[J]. Instrumentation Standardization and Metrology, 2015 (5): 15-18.
10. Ren Chuan, Chen Lei. An enterprise digital twin built on the basis of smart factory[J]. Collection, 2018, 21.
11. Chu J. The future development of process industry smart factory[J]. Science and Technology Herald, 2018, 36(21): 23-29.
12. Shi Yiming. Exploration and practice of process industry smart factory construction[J]. ZTE TECHNOLOGY JOURNAL, 2016.
13. Wang Xi-wen. Industry 4.0: Smart Industry[J]. Internet of Things, 2013, 3(12): 3-4. 3-4.
14. Zheng Hua. Exploration of smart factory construction from concept to practice[J]. China Management Informatization, 2017, 20(22): 66-67.
15. CHEN Guojin, JIANG Zhoushu, SU Shaohui, et al. Construction and application of smart factory experiment system based on industry 4.0[J]. Modern Education Technology, 2017, 27(7). 121-126.
16. Jin Guang. System integration design of wool textile smart factory[J].Wool Textile Journal, 2019, 47(12).
17. Yang, C. H. "Development of intelligent building management system evaluation and selection for smart factory. An integrated MCDM approach." 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).IEEE, 2017.
18. Hwang, Heejoung. "Design of remote management system for smart factory." International Journal of Internet, Broadcasting and Communication 12.4 (2020): 109 121.
19. Romero, David, Ricardo Jardim-Goncalves, and Antonio Grilo. "Factories of the future. Challenges and leading innovations in intelligent manufacturing." International Journal of Computer Integrated Manufacturing 30.1 (2017). 1-3.
20. Goryachev, Alexey, et al. "Smart factory". Intelligent system for workshop resource allocation, scheduling, optimization and controlling in real time." Advanced materials research 630 (2013). 508-513.
21. Zhou, Zhenquan, et al. "The development of the smart factory during the Industry 4.0 period under the study of system thinking." Proceedings of the 2023 International Conference on Artificial Intelligence, Systems and Network Security. 2023.
22. Durakbasa, Numan M., et al. "Intelligent integrated management and advanced metrology for quality toward the factory of the future." Proceedings of 3rd International Conference on the Industry 4.0 Model for Advanced Manufacturing.AMP 2018 3. Springer International Publishing, 2018.
23. Lazaroiu, George, et al. "Artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, and sustainable cyber-physical management systems in big data-driven cognitive manufacturing." Oeconomia Copernicana 13.4 (2022): 1047-1080.
24. Kumar, Prashant, Izaz Raouf, and Heung Soo Kim. "Review on prognostics and health management in smart factory. From conventional to deep learning perspectives." Engineering Applications of Artificial Intelligence 126 (2023). 107126.
25. Okeagu, Fredrick Nnaemeka. Development of an Intelligent System for Maintenance of Equipment in Smart Factories.Diss. Council for the Regulation of Engineering in Nigeria, 2023.
26. Kumar, Sachin, Ajit Kumar Verma, and Amna Mirza. "Artificial Intelligence and Intelligent Factories for the Future." Digital Transformation, Artificial Intelligence and Society. Opportunities and Challenges. Singapore. Springer Nature Singapore, 2024. 91-102.