**文献原文**

Abstract: This paper designs and implements an energy management system based on the Spring Boot framework. The system mainly includes three layers, which are the data collection layer, the business logic layer, and the display interface layer from bottom to top. The data collection layer is based on the RS-485 electrical standard and the MODBUS communication protocol. The two protocols connect all energy consumption monitoring points into a mixed topology communication network in the enterprise. The programs in the data collection layer poll each energy consumption monitoring point in the network to collect the data and transmit to the business logic layer. The business logic layer is developed on the basis of the Spring Boot framework and mainly includes two parts: the MySQL database and Tomcat server. In the MySQL database, the stored data are horizontally split according to the time column and stored in different data tables. The split of data reduces the load of a single data table and improves the query performance of the database. The Tomcat server is built into the Spring Boot framework to provide a basic environment for system operation. The Spring Boot framework is the core component of the system. It is responsible for collecting, storing, and analyzing data from energy consumption monitoring points, receiving and processing data requests from the display interface layer. It also provides standard data interfaces to the external programs. The display interface layer is developed on the basis of the Vue framework and integrated into the Spring Boot framework. The display layer combines an open-source visualization chart library called ECharts to provide users with a fully functional and friendly human–computer interaction interface. Through the calculation of hardware and software costs, considering the personnel cost in different regions, the total cost of the energy management system can be estimated. The cost of construction was approximately 210,000 USD in this paper. Since the system was actually deployed in a manufacturing company in December 2019, it has been operating stably for more than 600 days. Keywords: Spring Boot framework; hybrid topology; horizontal database split; Tomcat server

. Introduction Energy is the lifeblood of the national economy and the foundation of industrial development. Modern enterprises, especially the manufacturing industry, consume huge amounts of energy such as water and electricity in the process of production. These high-energy-consuming enterprises have to pay high energy costs each year, but the energy utilization rate cannot be effectively improved. The high energy cost not only affects the energy conservation and consumption reduction, but also weakens the market competitiveness of the enterprise [1]. It is of great significance to establish an energy management system and fully control the total energy consumption of the enterprise to improve the energy utilization rate [2]. In the power market, there are large fluctuations in the demand for electrical energy at different times, which causes a huge load on the power system. Demand response [3] has become an important part of the smart grid system. The energy management system can calculate and analyze the energy consumption of theenterprise to determine the peak and valley period of energy consumption. The enterprise can dynamically adjust the production plan according to the analysis, cut the peak, and flatten the valley, further reducing energy costs and increasing corporate profits. At the same time, the system also helps to achieve a balance between energy supply and demand, as well as to ensure the stability of the power market. In the mid-1960s, Japan first began to research and develop energy management systems, which were mainly used in iron and steel plants around the country [4]. The early energy management system was relatively small and simple in function. It was mainly used for the collection and monitoring of energy consumption data and the control of equipment. In developed countries represented by the United States and Germany, the industrial and social systems attach great importance to energy consumption measurement and control, and many companies have their own mature energy management systems. The Ford Company (USA), General Motors Company (USA), and Siemens Company (Germany) have established their own unique energy consumption management platforms and have obtained great benefits in terms of energy saving and consumption reduction [5]. China’s energy management system started in the 1980s [6]. China National Heavy-Duty Truck, Shanghai Automotive, and other manufacturing companies are actively building energy management systems, collecting, processing, and analyzing the energy consumption data of the enterprises. These companies also gradually constructed energy management centers and carried out energy consumption optimization work. After more than 30 years of development, the energy management situation has been greatly improved. In the traditional energy measurement and management systems, the energy consumption of various areas and equipment relies on manual observation and recording. The energy consumption of some areas and equipment can only be estimated, and the error of data recording is extremely large. The data are manually copied after being summarized, and the transmission is seriously delayed. It is also hard to detect and deal with abnormal situations in time such as water leakage and electric leakage [7]. Manually recorded data are mainly handwritten and stored on paper, resulting in many inconveniences in the periodic analysis of regional energy consumption. With the development of modern Internet technology, technologies such as servers and databases have gradually been applied to the industrial field [8]. People hope to be free from heavy meter-reading tasks and realize remote automatic collection and storage of energy consumption data, which will make energy consumption data timely and accurate, thus effectively preventing late reporting, underreporting, and fraudulent data. Enterprise managers can complete complex calculation tasks with the help of computers, conduct a comprehensive analysis of energy consumption data, find problems in time, and rectify them, ultimately improving energy utilization. Focusing on the above problems, this paper designs and implements an energy management system based on the Spring Boot framework. The system combines modern Internet technology and industrial data collection technology to integrate data collection, data storage, real-time monitoring, energy consumption analysis, consumption forecasting, etc. The functions realize the management and control of the energy consumption in the enterprise, so as to accurately manage and analyze the energy consumption of each region and each production process, as well as provide a reliable and scientific basis for the enterprise managers to formulate energy-saving plans.

Design of System Architecture The energy management system designed in this paper is based on the Spring Boot framework and integrates the MySQL database and the Vue front-end framework. The system design follows the three-tier architecture principle [9] and consists of three layers: from bottom to top, the data collection layer, business logic layer, and display interface layer. The overall system architecture is shown in Figure 1

In the overall system architecture shown in Figure 1, the data collection layer connects all energy consumption monitoring points into a communication network, polls every node for collecting the original data, and transmits the data to the business logic layer. The business logic layer parses the received data and stores it in the database. After the calculation and analysis, the Tomcat server can provide data services for the display interface layer. The display interface layer is used to display the results of data analysis and provide a human–computer interaction interface. Both the display interface layer and the business logic layer are constructed by the Spring Boot framework.

Data Collection Layer Many energy consumption monitoring points are set up in the enterprise and production areas to monitor the energy consumption of the area or production facilities. Each energy consumption monitoring point is equipped with at least one standard measuring device, such as electric energy meters, flow meters, and totalizers. The electric energy meter is used to monitor the consumption of electric energy, voltage value, and electric current of the area or equipment. The flow meter is used to monitor the instantaneous flow and the cumulative flow of water. The totalizer is used to monitor the instantaneous flow rate, cumulative flow rate of steam or compressed air, and the temperature and pressure of the gas. Energy consumption monitoring points and standard measuring devices can accurately reflect the energy consumption of enterprise and production facilities; hence, they are widely used. RS-485 is an electrical interface standard which is widely used in physical layer connections in the industrial control field [10]. RS-485 usually uses two signal lines, A and B, adopts differential signal transmission, works in half-duplex mode, and has a maximum transmission distance of 1200 m. RS-485 supports multipoint connection, and it has the characteristics of high transmission rate and good anti-noise interference performance. MODBUS-RTU is an open serial communication protocol, which has now become an industry standard for communication protocols. It is also a widely used communication protocol between industrial electronic devices [11]. The MODBUS-RTU protocol is based on the master/slave architecture. The master sends requests to slaves, and the slaves respond to the master with their own data. The MODBUS-RTU protocol stipulates the data format of the requests and responses and uses CRC to ensure the correctness and integrity of the data.

The data collection layer uses the RS-485 bus to connect all energy consumption monitoring points into a mixed topology node communication network [12]. The communication network combines a bus topology and a star topology. The node communication network is shown in Figure 2.

In the communication network shown in Figure 2, several communication nodes and repeaters are connected to the RS-485 bus, and all communication nodes share the RS-485 common bus. The bus-type topology is simple and has high reliability. The failure of a single node will not affect the communication between the bus and other nodes. The bus is convenient to connect new communication nodes or repeaters. The main function of the repeater is to enhance the transmission signal, extend the transmission distance, and ensure the stability and integrity of the signal during long-distance transmission. Each repeater on the bus serves as the central node of the star network and can be connected to multiple communication nodes. Repeaters are mainly used in energy monitoring groups far away from the bus. The RS-485 hub is the collection terminal of the communication network. The collection terminal is connected to a computer or server through a serial converter which can convert the RS-485 interface to an RS-232 or USB interface. The collection terminal is the only master station in the communication network, and the other nodes are slave stations. Every slave station has a unique MODBUS address to distinguish it from other slave stations. The repeater is only used to enhance and forward signals; it is not used as a communication node. According to the data format specified in the MODBUS protocol, the collection terminal sends requests to the communication network to collect the data of the slave station with the specified address. All slave stations receive the request. If its own address is different from the address in the request, the request is ignored by the slave station. If the address is the same, the slave station returns its own data to the collection terminal. The collection terminal polls all nodes one by one to collect the data in the network. The collection terminal is directly controlled by the computer. The system verifies the data returned to ensure the Integrity. If the verification fails or the return times out, the data are discarded and the request is retransmitted. If failure occurs again, the system skips the node and logs the time of occurrence. 2.2. Business Logic Layer The business logic layer was constructed using the Spring Boot framework, and it mainly includes two parts: the database and the server. The database is used to store energy consumption data and user information. The program running in the server calculates and analyzes the data, provides data services for the display interface layer, and provides standard data interfaces for other users or systems. 2.2.1. Design of the Database Database design refers to the design and optimization of the logical mode and physical structure of the database under a given application environment. It establishes a database and application system to effectively store and manage data [13]. The data in the databaseare organized, described, and stored according to a certain data model. This results in several advantages such as less redundancy, higher independence, and easy scalability. MySQL is a relational database management system [14]. This database is small and easy to install and maintain. MySQL is open-source and free, with popular architectures such as LAMP and LNMP. The MySQL database has become the first choice of enterprise users because of its excellent performance and stable service. An E-R (entity relationship) diagram [15] is used to describe the relationship between entity types, attributes, and connections. The E-R diagram is a method of describing conceptual models to facilitate the design of the logical structure of the database. The main entities in this paper’s database are users, the energy consumption monitoring point, and its data. Figure 3 shows the connections between entities. The field designs of the energy consumption monitoring point ant its data are shown in Table 1

In order to ensure the continuity and accuracy of the energy data in the enterprise, the data collection layer collects data from the node communication network 24 h a day. In the actual operation of the system, it is found that RS-485 communication completes the data collection of one node in an average of 2 s. Calculated in this way, 43,200 records are generated every day, 1,296,000 records are generated every month, and 15,768,000 records are generated every year. Under the premise that the design life of the system is 20 years, it will generate billions of data. Storing such a large amount of data in a single data table will seriously affect the read and write performance of the database. The split of a data table [16] means that the data stored in a single data table is distributed to multiple data tables according to a certain dimension, so as to reduce the load of a single data table and improve the performance. Common database splitting methods include vertical splitting and horizontal splitting. Vertical splitting is to split a table with many attributes into different tables according to different attributes. The structure of each table after splitting is different, and there is at least one column of intersection between each table. Horizontal splitting refers to splitting a table into multiple tables on the basis of a certain field and according to certain rules. The structure of each table is the same after splitting, but the data of each table are different and there is no intersection.

The database designed in this paper adopted a horizontal split method. The data in the data table were split according to the field of time. Whenever the value of the time column entered a new month, the data were put into a new table. This method realized a data table split by month based on the time field. For example, the data of the year 2020 were scattered into 12 tables. Table 2 shows the data volume of each table and the error rate. In Table 2, the data table data\_202005 stores the data of May 2020. The table has a total of 1,285,634 records. The time value of the first record is “1 May 2020 00:00:00” and the time value of the last record is “31 May 2020 23:59:58”. The data table data\_202006 has a total of 1,293,600 records, of which the time value of the first record is “1 June 2020 00:00:00”. A total of 12 tables were established in 2020, and each table corresponds to one month’s data. The data volume of each month is less than the estimated maximum data volume of 1,296,000. Taking into account the power outage and computer restart in the enterprise, the maximum data volume error rate is 0.49%, which belongs to the normal range. In the actual operation of the system, the horizontal split of the data table reduces the data load of a single table, keeps the data volume of each table at the million level, and ensures the speed of data query. The system is able to withstand high concurrency and maintain high stability. 2.2.2. Application of Spring Boot Framework In the business logic layer, the Spring Boot framework is the core component, which is responsible for collecting, storing, and analyzing data from the data collection layer. It also receives and processes data requests from the display interface layer, as well as provides standard data interface to the external programs. The Spring Boot framework [17] is a subproject under the Spring project and is currently the most popular Java enterprise-level development framework. The Tomcat server is integrated inside the Spring Boot framework, which can be run directly without deployment. The Spring Boot framework abandons the cumbersome XML configuration files in the traditional framework and is developed on the basis of annotations. Development based on annotations simplifies the configuration and greatly reduces the difficulty of development. When integrating third-party frameworks, Spring Boot can automatically configure and manage dependencies, allowing developers to focus on the design and implementation of business logic. Figure 4 shows the main steps of a Spring Boot application startup process. From Figure 4, we can analyze the main steps of the Spring Boot application startup process. 1. The system looks for the annotation named @SpringBootApplication and creates an in-stance object of SpringApplication as the main entrance of the program; 2. The type of application is determined (a web application in this case); 3. All initializers and all listeners are loaded; 4. Application parameters and environment variables are set; 5. The context environment is prepared. This is the core step of the Spring Boot framework startup. The dependencies of third-party libraries, automatic configuration, and the startup of the built-in server Tomcat are established in this step; 6. User-defined methods are executed. This marks the completion of the Spring Boot application before normal work can begin. In step 3, the Spring Boot framework loads all listeners. The listeners can listen to the data request from the front end, forward the request to the handler, and call the userdefined business processing function to respond to the request. In step 5, the Spring Boot framework starts the built-in Tomcat server, provides a basic operating environment, and loads third-party libraries and frameworks. The Vue front-end framework in the interface display layer is integrated in this step. In step 6, the Spring Boot application executes user-defined methods to complete multiple tasks such as database reading and writing, data processing, and performing timing tasks. At this point, a Spring Boot application is started successfully. The powerful features of Spring Boot provide developers with great convenience. The data query function is taken as an example. First, we add the annotation “@RestController” to the public class DataController to indicate that the class is a control class. Then, we add the annotation “@RequestMapping(“/getDataById/{id}”)” to the method public String getDataById (@PathVariable Integer id) in the class. The Spring Boot framework automatically maps to the corresponding method according to the address of the request url, extracts the parameter id in the url, and provides it to the method for internal use. Finally, we write a specific business logic code in the method body, e.g., executing SQL statements and returning the queried data in the form of a string. The development of the function of data query according to the id is completed. In the Spring Boot framework, all configurations are in a file named application.yaml, i.e., the database driver, login name, and password. The framework automatically loads the configuration file and establishes a database connection. Developers only need to write SQL statements to complete data query.

Display Interface Layer The display interface layer is used to display the data analysis results of the business logic layer. This layer realizes several functions such as login, registration, and query. It also provides a human–computer interaction interface. The display layer is developed on the basis of the Vue framework, combined with the ECharts, an open-source visualization chart library, to provide users with a set of front-end display interfaces with complete functions and convenient operation. Vue is a progressive framework for building user interfaces [18]. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. It is not only easy to use, but also easy to integrate with third-party libraries or existing projects. According to the design idea of MVVM (model–view–viewmodel), Vue realizes the two-way binding of data and view, reduces the coupling degree of each part, and is more flexible to use. The Vue framework can be directly integrated into the Spring Boot framework, and the Spring Boot framework can help complete automatic configuration. Figure 5 is a schematic diagram of the MVVM mode of the Vue framework.

ECharts is an open-source visualization library based on JavaScript [19], covering various industry charts. It can run smoothly on desktop and mobile terminals, and it is compatible with multiple browsers. The introduction of ECharts is simple. Through the setting of configuration items, you can control the data presentation and visual effects. ECharts realizes the interaction between users and data through rich components and highly personalized visualization solutions, thereby enhancing users’ ability to acquire knowledge and optimize data analysis. This system uses the Vue front-end framework and ECharts to design the web display interface of the energy management system. In the B/S architecture, the web interface can be accessed by a browser, with a rich interface and convenient operation. 2.4. Development Cost Calculation The display interface layer is used to display the data analysis results of the business logic layer. This layer realizes several functions such as login, registration, and query. It also provides a human–computer interaction interface. In the data collection layer, the energy consumption monitoring point and the node communication network are composed of various hardware equipment, and the construction cost is relatively high. In the energy consumption monitoring point, the cost of one electric energy meter is about 300 USD. The flow meters are available in large and small diameters. The costs are 400 and 250 USD, respectively. The cost of one totalizer is approximately 150 USD. In the node communication network, the twisted pair shielded wire used for connection is 30 USD per 100 m. The price of hubs and repeaters is approximately 100 and 50 USD, respectively. In the enterprise, a total of 196 electric energy meters, 84 flow meters including 57 largediameter and 27 small-diameter meters, and 52 totalizers are deployed. The total cost isabout 101,150 USD. The total cost of hubs, repeaters, and twisted pair shielded cables is approximately 55,000 USD. In the data collection layer, the cost of hardware equipment is difficult to reduce and, thus, it costs a lot of money. We hope that, at the software level, we can appropriately reduce costs while ensuring safety and functionality. Therefore, the open-source framework represented by Spring Boot has come into focus. The business logic layer and the display interface layer make extensive use of various open-source frameworks and components. MySQL is a high-performance open-source database, and the Spring Boot framework is also the most popular Java enterprise-level development framework. Vue and Echarts are also widely used front-end open-source frameworks. Whether using Spring Boot or Vue, technically, you can get substantial support and assistance from the community. In particular, the Spring Boot framework has been maintained and updated by the authorities in the past 10 years to ensure security and stability. In these two layers, the cost of system construction mainly comes from hiring developers and software debugging and maintenance. Taking into account the differences in different regions and the length of the construction period, the cost of hiring workers and developers is quite different. In this project, the personnel cost was approximately 50,000 USD. The construction cost of the entire energy management system was approximately 210,000 UAD.

System Operation and Analysis The energy management system was officially deployed in a manufacturing enterprise in December 2019 and has been operating stably for more than 600 days. Figure 6 shows the large-screen display interface of the energy management system.

Because it is outside of the actual production environment, it is impossible to obtain the latest data. The data in the figure are simulated values in a laboratory environment. In Figure 6, the main interface shows the consumption of electricity, water, and steam in the enterprise. It also displays the electricity consumption in different months and the proportion of energy consumption in the main area. A brief alarm message is displayed in the lower left corner of the interface. The large-screen display interface is mainly used by visitors to determine the energy management of the enterprise. Figure 7 shows the backend interface of the energy management system

From Figure 7, we can see today’s energy consumption, yesterday’s energy consumption, and the cumulative energy consumption of this month in the enterprise. At the same time, we can view the daily consumption within a certain period of time by setting the time range. The ranking on the right is the energy consumption monitoring points with the most usage during the time period and their usage. The background interface of the energy management system provides data visualization and analysis functions for enterprise managers. Figure 8 shows the real-time data interface.

From Figure 8, we can see the real-time data of the energy consumption monitoring points in the production area. The electric energy monitoring points display the electricity consumption, the water and steam energy consumption monitoring points display the instantaneous flow and cumulative flow, and the steam and compressed air energy consumption monitoring points also record pressure and temperature. Figure 9 shows the historical data query interface. Water consumption is taken as an example. In this interface, you can query the water consumption of a single or multiple energy consumption monitoring points.

. Conclusions This paper designed and implemented an energy management system based on the Spring Boot framework, which has run stably for more than 600 days in the actual production environment. The system has a three-tier architecture: from bottom to top, data collection layer, business logic layer, and display interface layer. The data collection layer monitors all energy consumption monitoring points in the enterprise in real time, ensuring the integrity and continuity of the data. In the business logic layer, the database is split horizontally, which greatly reduces the data load of a single table. The maximum data storage capacity of a single table is 1,296,000, and the data query efficiency is extremely high. The Spring Boot framework can provide stable data services, making it easy to maintain and extend functions. At the display interface layer, the display interface developed using Vue and ECharts is rich in functions and easy to operate. The various functions of the system can operate normally and stably, which provides convenience for the management and control of enterprise energy consumption. For the cost of construction, the hardware cost is based on the quantity and price of the device, and the personnel cost is based on the working hours. Accordingly, the total cost of the energy management system can be estimated. In this paper, the total cost of construction was approximately 210,000 USD. The system is currently deployed in the internal server of the enterprise, and the energy management interface can only be accessed by computers in the same local areanetwork. In the future, the system can be deployed in a cloud server, and mobile phone-side adaptation and enhanced identity authentication can be implemented to realize any device to access and manage the system anytime and anywhere. For the node communication network, wireless transmission technology such as 5G will be adopted in the future, and data will be directly transmitted to the server in the cloud to reduce the local residence time of data and avoid possible data injection and tampering. For the data security issues of cloud deployment, in the future, in addition to relying on Spring Boot’s security framework and regular database backups, the system will enhance the identity authentication of logged in users, record the user’s IP address and sensitive operation behavior, and perform data transmission during transmission. The secondary encryption increases the difficulty of data cracking, thereby improving the stability of the communication network and the data security of the cloud system.

From Figure 7, we can see today’s energy consumption, yesterday’s energy consumption, and the cumulative energy consumption of this month in the enterprise. At the same time, we can view the daily consumption within a certain period of time by setting the time range. The ranking on the right is the energy consumption monitoring points with the most usage during the time period and their usage. The background interface of the energy management system provides data visualization and analysis functions for enterprise managers.

From Figure 8, we can see the real-time data of the energy consumption monitoring points in the production area. The electric energy monitoring points display the electricity consumption, the water and steam energy consumption monitoring points display the instantaneous flow and cumulative flow, and the steam and compressed air energy consumption monitoring points also record pressure and temperature. Figure 9 shows the historical data query interface. Water consumption is taken as an example. In this interface, you can query the water consumption of a single or multiple energy consumption monitoring points.

Figure 10 shows the settings interface. In this interface, all alarm messages are shown and can be sent to a specific email or phone. Managers can change the times that reports are automatically printed.

References 1. Wei, J.Y.; Zhao, X.Y.; Sun, X.S. The evaluation model of the enterprise energy efficiency based on DPSR. Environ. Sci. Pollut. Res. 2019, 26, 16835–16846. [CrossRef] [PubMed] 2. Goncalves, V.S.; Jose, M. Energy management system iso 50001:2011 and energy management for sustainable development. Energy Policy 2019, 133, 110868. 3. Irtija, N.; Sangoleye, F.; Tsiropoulou, E.E. Contract-Theoretic Demand Response Management in Smart Grid Systems. IEEE Access 2020, 8, 184976–184987. [CrossRef] 4. Yu, M.; Yue, G.; Song, J.; Pang, X. Research on intelligent city energy management based on internet of things. Clust. Comput. 2019, 22, S8291–S8300. [CrossRef] 5. Raji, M.N.; Maksimovi, R.M.; Milosavljevi, P.; Pavlovi, D. Energy Management System Application for Sustainable Development in Wood Industry Enterprises. Sustainability 2019, 12, 76. [CrossRef] 6. Jia, B. Research and Application of Small Area Energy Management System. Master0 s Thesis, Shandong University, Jinan, China, 8 June 2020. 7. Sa, A.; Thollander, P.; Rafiee, M. Industrial Energy Management Systems and Energy-Related Decision-Making. Energies 2018, 11, 2784. [CrossRef] 8. Choi, J.S. Energy Management Agent Frameworks: Scalable, Flexible, and Efficient Architectures for 5G Vertical Industries. IEEE Ind. Electron. Mag. 2021, 15, 62–73. [CrossRef] 9. Barabanova, I.A.; Kravets, O.J.; Tkalich, S.A.; Muting, D.I. Analysis of the intermediate layer work in the three-tier architecture “client-server” of automation engineering problems. IOP Conf. Ser. Mater. Sci. Eng. 2019, 537, 032011. 10. Zhao, Y.J.; Ma, J.N.; Li, S.J.; Li, J. Design of Modbus Wireless Communication System Based on Remote Data Transmission. In Proceedings of the 2016 International Forum on Mechanical, Control and Automation, Shenzhen, China, 30–31 December 2016. 11. Yue, G. Design of Intelligent Monitoring and Control System Based on Modbus. In Proceedings of the 5th International Conference on Communication, Image and Signal Processing, Chengdu, China, 13–15 November 2020. 12. Wang, S.; You, H.; Yue, Y.; Cao, L. A novel topology optimization of coverage-oriented strategy for wireless sensor networks. Int. J. Distrib. Sens. Netw. 2021, 17. Available online: https://journals.sagepub.com/doi/full/10.1177/1550147721992298 (accessed on 2 November 2021). [CrossRef] 13. Yang, Y.; Mei, Z.; Zheng, B.; Qiu, S. Design of Enterprise Management System Based on Edge Computing Architecture. Mob. Inf. Syst. 2021, 2021, 5512958. 14. Eyada, M.; Saber, W.; Genidy, M.; Amer, F. Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments. IEEE Access 2020, 8, 110656–110668. [CrossRef]

15. Ordonez, C.; Al-Amin, S.T.; Bellatreche, L. An ER-Flow Diagram for Big Data. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020. Available online: https://www.webofscience.com/ wos/alldb/full-record/WOS:000662554705140 (accessed on 2 November 2021). 16. Liu, T.Y. Design and Implementation of Distributed Mall Based on SSM. Master0 s Thesis, China University of Geosciences, Beijing, China, May 2020. 17. Miao, K.H.; Li, J.; Hong, W.X.; Chen, M.T. A Microservice-Based Big Data Analysis Platform for Online Educational Applications. Sci. Program. 2020, 2020, 6929750. Available online: https://www.hindawi.com/journals/sp/2020/6929750/ (accessed on 2 November 2021). [CrossRef] 18. Li, M.H.; Hu, J.H.; Lin, X.W. The Development of Web Application Front-End of Intelligent Clinic Based on Vue.js. In Proceedings of the 2019 Chinese Intelligent Automation Conference, Nanjing, China, 20–22 September 2019; Lecture Notes in Electrical Engineering. Available online: https://link.springer.com/chapter/10.1007/978-981-32-9050-1\_77 (accessed on 2 November 2021). 19. Ban, X.J.; Wang, B.; Cheng, C.X.; Taghzouit, S. Iron and Steel Enterprises Big Data Visualization Analysis Based on Spark. In Proceedings of the 15th International Conference on Cooperative Design, Visualization and Engineering, Hangzhou, China, 21–24 October 2018; Lecture Notes in Computer Science. Available online: https://link.springer.com/chapter/10.1007%2F978-3-030-0 0560-3\_39 (accessed on 2 November 2021)

翻译：

能源管理系统的设计与实现 基于Spring  Boot框架

摘要：本文设计并实现了一个基于Spring  Boot框架的能源管理系统。该系统主要包含三层，自下而上分别为数据采集层、 业务逻辑层和展现界面层。数据采集层基于RS‑485电气标准和MODBUS通信协议，这两个协议将企业内所有能耗监测 点连接成一个混合拓扑通信网络。数据采集层中的程序轮询网络中的各个能耗监测点，收集数据并传输至业务逻辑层。 业务逻辑层基于Spring  Boot框架开发，主要包含两部分：  MySQL数据库和Tomcat服务器。MySQL数据库中存储的 数据根据时间列进行水平拆分，并存储在不同的数据表中。数据的拆分降低了单个数据表的负载，提高了数据库的查询性 能。Tomcat服务器内置于Spring  Boot框架中，为系统运行提供基础环境。Spring  Boot框架是该系统的核心组件。负责 对能耗监测点数据进行采集、存储和分析，接收并处理显示接口层的数据请求，并为外部程序提供标准的数据接口。 展现界面层基于Vue框架开发，并集成到Spring  Boot框架中。展现层结合开源可视化图表库ECharts，为用户提供功能 ⻬全、友好的人机交互界面。通过对软硬件成本的测算，并考虑不同区域的人员成本，可以估算出该能源管理系统的总成 本。本文估算的建设成本约为21万美元。该系统自2019年12月在某制造企业实际部署以来，已稳定运行超过600天。

简介 能源是国民经济的命脉，是工业发展的基础。现代企业尤其是制造业在生产过程中消耗大量的 水、电等能源。这些高耗能企业每年都要支付高昂的能源成本，但能源利用率却得不到有效提高。高昂 的能源成本不仅影响节能降耗，而且削弱了企业的市场竞争力[1]。建立能源管理体系，全面控制企业 能源消耗总量，对提高能源利用率[2]具有重要意义。在电力市场中，不同时段的电能需求波动较大， 给电力系统造成巨大的负荷。需求响应[3]已成为智能电网系统的重要组成部分。 能源管理系统可以计算和分析能源消耗企业能够确定能源消费的峰谷时段。企业可以根据分析结果动态调整生产计划，削峰平谷，进 一步降低能源成本，提高企业利润。同时，该系统也有助于实现能源供需平衡，保障电力市场的 稳定。 20  世纪  60  年代中期，日本最早开始研究和开发能源管理系统，主要应用于全国各地的钢铁厂 [4]。早期的能源管理系统规模相对较小，功能简单，主要用于能耗数据的采集、监测以及设备的控制。 以美国、德国为代表的发达国家，工业和社会系统非常重视能源消耗的测量和控制，许多企业都拥有自己成 熟的能源管理体系。美国福特公司、美国通用汽车公司、德国西门子公司等都建立了自己独特的能源消耗管 理平台，在节能降耗方面取得了巨大效益[5]。我国的能源管理体系起步于20世纪80年代[6]，中国重汽、上海 汽车等制造企业积极建设能源管理系统，收集、处理和分析企业的能源消耗数据，并逐步建设能源管理中心， 开展能耗优化工作。经过30多年的发展，能源管理状况得到了很大改善。 传统的能源计量与管理体系中，各区域、各设备的能耗主要依赖人工观测和记录。部分区域、设备的能 耗仅能估算，数据记录误差极大。数据汇总后需人工抄录，传输时延严重，且难以及时发现和处理漏水、漏电 等异常情况[7]。人工记录数据主要以手写和纸质存储为主，给区域能耗的定期分析带来诸多不便。 随着现代互联网技术的发展，服务器、数据库等技术逐渐应用于工业领域[8]。人们希望从繁重的抄表 任务中解脱出来，实现能耗数据的远程自动采集和存储，使能耗数据及时准确，有效防止迟报、漏报和数据造 假。企业管理者可以借助计算机完成复杂的计算任务，对能耗数据进行全面分析，及时发现问题并予以纠正， 最终提高能源利用率。 针对以上问题，本文基于Spring  Boot框架设计并实现了一套能源管理系统，该系统结合现代互联网 技术与工业数据采集技术，集成数据采集、数据存储、实时监控、能耗分析、能耗预测等功能，实现对企业能源 消耗的管控，从而能够精准地管理和分析各区域、各生产环节的能耗情况，为企业管理者制定节能计划提供 可靠、科学的依据。

 系统架构设计本文设计的能源管理系 统基于Spring  Boot框架，集成了MySQL数据库和Vue前端框架。系统设计遵循三层架构原则[9] ，由三 层组成：自下而上分别为数据采集层、业务逻辑层、展现界面层。系统整体架构如图1所示。业务逻辑层解析接收到的数据并存储到数据库中。经过计算分析后，Tomcat服务器可以为展现界面层提供数据服务。展现界面层用于 展示数据分析的结果，并提供人机交互界面。展现界面层和业务逻辑层均由Spring  Boot框架构建。

 数据采集层企业和生产区域设置 了多个能耗监测点，用于监测该区域或生产设施的能耗情况。每个能耗监测点至少配备一个标准计量装置，例如电能表、流量计 和积算仪。电能表用于监测区域或设备的电能消耗、电压值和电流值。流量计用于监测水的瞬时流量和累计流量。积算仪用于监测蒸汽或 压缩空气的瞬时流量、累计流量以及气体的温度和压力。能耗监测点和标准计量装置能够准确反映企业和生产设施的能耗情况，因此被 广泛应用。

采集终端为通讯网络中唯一主站，其他节点为从站。每个从站拥有唯一的MODBUS地址，用于区别于其他从站。中继器仅 用于信号增强和转发，不作为通讯节点使用。采集终端按照MODBUS协议规定的数据格式，向通讯网络发送请求，采集指定地 址从站的数据。所有从站均接收该请求，若自身地址与请求中的地址不同，则忽略该请求；若地址相同，则将自身数据返回给采集 终端。采集终端逐个轮询网络中所有节点，采集数据。采集终端由计算机直接控制，系统对返回的数据进行校验，以确保数据的 完整性。若校验失败或返回超时，则丢弃数据并重新发送请求。若再次失败，系统将跳过该节点并记录失败时间。

  业务逻辑层业务逻辑层采用 Spring  Boot框架构建，主要包含数据库和服务端两部分。数据库用于存储用电数据和用户信息。服务端运行的程序对数 据进行计算和分析，为展现界面层提供数据服务，并为其他用户或系统提供标准的数据接口。 2.2.1.  数据库设计 数据库设计是指在给定的应用环境下，对数据库的逻辑模式和物理结构进行设计和优化，建立 数据库和应用系统，以有效地存储和管理数据[13]。数据库中的数据根据特定的数据模型进行组织、描述和存储。这导致 冗余度低、独立性高、易于扩展等优点。 MySQL  是一个关系数据库管理系统[14]。该数据库很小，并且 易于安装和维护。MySQL  开源且免费，并采用流行的架构 例如LAMP和LNMP。MySQL数据库已经成为企业级的首选 凭借其卓越的性能和稳定的服务，深受用户喜爱。ER（实体关系） 图[15]用于描述实体类型、属性和 连接。ER  图是一种描述概念模型的方法，用于促进 数据库逻辑结构的设计。本文数据库中的主要实体 分别是用户、能耗监测点及其数据。图3显示了 实体之间的连接。能耗监测的现场设计为了保证企业能源数据的连续性和准确性， 数据收集层每天24小时从节点通信网络收集数据。在 系统实际运行过程中发现，RS‑485通信完成 平均2秒采集一个节点的数据，这样算下来，43200条记录 每天生成  1,296,000  条记录，每月生成  15,768,000  条记录 每年都会产生。在系统设计寿命为20年的前提下， 将产生数十亿的数据。将如此大量的数据存储在单个数据表中将 严重影响数据库的读写性能。 数据表的拆分[16]是指存储在单个数据表中的数据 按照某个维度分布到多个数据表，以减少负载 单个数据表的拆分，提升性能。常见的数据库拆分方法 包括垂直拆分和水平拆分。垂直拆分是指将一个表拆分成 根据不同的属性，将许多属性分成不同的表。每个表的结构 拆分后的表不同，并且至少有一列与 每个表。水平拆分是指将一个表拆分成多个表，拆分依据是 按照一定的规则，对某个字段进行排序。排序后，每个表的结构都相同 评论 唯一编号 仪表名称 Modbus地址 仪表类型 数据收集时间 瞬时流量 累计流量 温度 压力 分裂，但每个表的数据都不同，没有交集。

 展示界面

层展示界面层 用于展示业务逻辑层的数据分析结果，实现登录、注册、查询等功能，并提供人机交互界面。展示 层基于Vue框架开发，结合开源可视化图表库ECharts ，为用户提供一套功能⻬全、操作便捷的前端 展示界面。 Vue  是一个用于构建用户界面的渐进式框架[18]。与其他单体式框架不同，Vue  从一开始就被 设计为可逐步采用的。 Vue  不仅易于使用，而且易于与第三方库或现有项目集成。Vue  遵循  MVVM（model‑view‑viewmodel）的 设计思想，实现了数据和视图的双向绑定，降低了各部分的耦合度，使用更加灵活。Vue  框架可以直接集成到   Spring  Boot框架中，并由  Spring  Boot  框架协助完成自动化配置。

ECharts  是一个基于  JavaScript  [19]的开源可视化库，涵盖各种行业图表。它能够在桌面和移动端流 畅运行，并兼容多种浏览器。ECharts  的入门非常简单，通过配置项的设置，即可控制数据呈现和视觉效果。 ECharts通过丰富的组件和高度个性化的可视化方案实现用户与数据的交互，从而增强用户获取知 识、优化数据分析的能力。 本系统采用Vue前端框架和ECharts设计能源管理系统的网页展示界面，采用B/S架构，网页界 面可以通过浏览器访问，界面丰富，操作便捷。

 开发成本计算

展示界面层用于展示业务逻辑层的数据分析结果，实现登录、注册、查询等功能，提供人机交互 界面。 在数据采集层，能耗监测点和节点通信网络由各种硬件设备组成，建设成本较高。能耗监测点中，电能表 单只成本约为2500元；流量计有大口径和小口径两种，成本分别为1700元和1500元；积算仪单只成本约为 1000元。节点通信网络中，连接所用的双绞屏蔽线每100m成本为1600元；集线器和中继器的价格分别约为 700元和300元。企业共布设电能表196块、流量表84块（其中大口径57块、小口径27块）、积算仪52台，总成本为 Machine Translated by Google 信息2021,  12,  457 13中的9 约101,150美元。集线器、中继器和双绞屏蔽电缆的总成本约为55,000美元。 在数据采集层，硬件设备成本难以降低，花费巨大。我们希望在软件层面，在保证安全性和功能性的前提 下，能够适当降低成本。因此，以  Spring  Boot  为代表的开源框架成为关注的焦点。业务逻辑层和展现界面层 则大量使用了各种开源框架和组件。

考虑到不同地区的差异以及建设周期的长短，雇佣工人和开发人员的成本差异较大。本 项目中，人员成本约为5万美元。整个能源管理系统的建设成本约为21万UAD。

系统运行与分析该能源管理系统于 2019年12月在某制造企业正式部署，至今已稳定运行600多天，图6为该能源管理系统大屏 显示界面。

结论 本文基于Spring  Boot框架设计并实现了一套能源管理系统，并在实际生产环境中稳定运行超过600 天。系统采用三层架构，自下而上依次为数据采集层、业务逻辑层、展现界面层。数据采集层实时监控企业内 所有能耗监测点，保证数据的完整性和连续性。在业务逻辑层，对数据库进行了水平拆分，大大降低了单表的 数据负载，单表最大数据存储量达129.6万条，数据查询效率极高。 Spring  Boot框架能够提供稳定的数据服务，易于维护和功能扩展。在展现界面层，采用Vue和ECharts开发 的展现界面功能丰富，操作简便，系统各项功能能够正常稳定运行，为企业能耗管控提供了便利。在建设成本 方面，硬件成本根据设备数量及价格计算，人员成本根据工时计算，据此估算出能源管理系统的总建设成本。 本文估算的总建设成本约为21万美元。 系统目前部署在企业内部服务器，能源管理界面只能由同一局域网内的电脑访问网络。未来可将系统部署在云端服务器中，并实现手机端适配和增强身份认证，实现任何设备随时随地访问 和管理该系统。对于节点通信网络，未来将采用5G等无线传输技术，将数据直接传输到云端服务器，减少数据 在本地的驻留时间，避免可能的数据注入和篡改。对于云端部署的数据安全问题，未来除了依赖Spring  Boot 的安全框架和定期进行数据库备份外，系统还将增强对登录用户的身份认证，记录用户的IP地址和敏感操作 行为，并在数据传输过程中进行二次加密，增加数据破解难度，从而提高通信网络的稳定性和云端系统的数 据安全性