A Remote Wireless Meter Reading System Website Based on Django Development

Zixin Song[1], Zhuo Liu[1], Jinlian Liang[2], Wenbin Zheng[3], Di Zhou[4,*]
[1]*Shenzhen Techstar Vista Technology Co., Ltd., Shenzhen, Guangdong, China*
[2]*School of Science, Harbin University of Science and Technology, Harbin, Heilongjiang, China*
[3]*Harbin Institute of Technology, Harbin, Heilongjiang, China*
[4]*Shenzhen Academy of Metrology & Quality Inspection, Shenzhen, Guangdong, China*
[*]*Corresponding author.*

Abstract: With the acceleration of urbanization and the increasing annual tension between supply and demand of freshwater resources, improving water resource utilization is urgent. In order to solve the problems of high cost and low timeliness of the traditional manual meter reading mode, this article develops a remote wireless meter reading system website based on the Django framework. The research focuses on the software design of the two key parts of the front-end display and back-end management of the website, and selects the appropriate software development platform and the website background database. In terms of technical implementation, the project adopts programming languages such as Python, SQL (Structured Query Language) and HTML (Hyper-Text Markup Language) for coding design, which effectively realizes the interactive function of the front and back ends of the website. Through this design, the data transmitted by the remote wireless meter reading device can be integrated and processed, and the powerful background management function of the Django framework is used to accurately distinguish the network node data of the remote wireless meter reading device in different regions. Through online and experiment in this article, not only does it has advantages in efficiency, security, and compatibility, but it also provides a strong support for the construction of water information.

Keywords: Django Framework; Remote Wireless Meter Reading Device; Database; Python; Website Design

1. Introduction

China possesses abundant total freshwater resources, yet faces a severe per capita water scarcity [1]. With the advancement of socioeconomic development and urbanization, traditional meter reading practices are challenged by their inability to rapidly grasp and analyze data. The implementation of remote wireless meter reading systems, grounded on internet-based infrastructures, emerges as a solution. It facilitates real-time monitoring and management of urban water usage, enhancing data management efficiency. Consequently, this optimizes the utilization and distribution of water resources, catalyzing the digital transformation and innovation within the water industry [2-4].

Singapore, Japan, and the OPTIMATICSTM company, among other nations and conglomerates, have achieved digitalization in their water utilities systems. In China, Shenzhen, Chongqing, Fuzhou and other cities have also established characteristic water network projects [5-7]. However, there are numerous deficiencies in the information construction of water affairs:

1. Technical specifications are not unified, which makes it difficult data exchange and integration between different systems. This not only increases the development and maintenance costs, but also limits the scalability and flexibility of the system.

2. The construction of water information system is scattered and lacks overall planning and coordination. This leads to the serious phenomenon of information island, and it is difficult to realize the centralized management and sharing of data.

3. The information process of water utilities is more important to construction than application, the operation and maintenance is unbearable, and the research on the system

website is not deeply analyzed from the perspective of life construction. In the delay between management and maintenance personnel, it is difficult to repair in time when the system fails, which affects the stable operation of the system [8,9].

To address these issues, this article proposes a remote wireless meter reading system website based on the Django framework.

The outline of the paper is as follows: Section II elaborates on the website requirements analysis. Section III presents the overall design scheme, followed by Section IV, which details the front-end design. The back-end development is described in Section V. Subsequently, Section VI tests for functionality, launch, performance, and usability of the website. Conclusions are summarized in Section VII.

2. Website Requirements Analysis

The login user of the remote wireless meter reading system website is the company maintenance personnel and management personnel. A perfect remote wireless meter reading website system should have the following functions: front and back interaction of the website, effective management of remote wireless meter reading data, user login authority, and reasonable analysis of device data. Maintenance personnel are primarily responsible for providing feedback on the usage of devices, while management personnel are mainly in charge of overseeing users and the system.

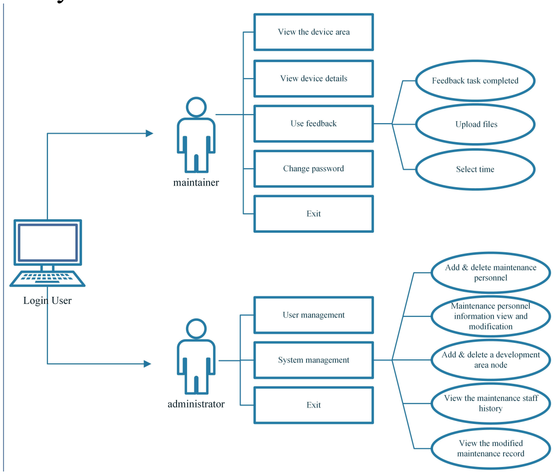
View the device area
View device details
Use feedback
Change password
Exit
Feedback task completed
Upload files
Select time
maintainer
Login User
User management
System management
Exit
administrator
Add & delete maintenance personnel
Maintenance personnel information view and modification
Add & delete a development area node
View the maintenance staff history
View the modified maintenance record

Figure 1. Use Case Diagram

The core function of the website is to manage the wireless device nodes. The use case diagram of the maintenance personnel and managers is shown in Figure 1. The usage authority of the maintenance personnel is mainly responsible for the feedback of the device and the uploading of relevant usage documents; the management personnel are mainly responsible for the management of users and the system, checking the maintenance personnel information, checking and modifying the history of maintenance personnel, and modifying the maintenance records.

The core business of the system is remote maintenance of devices. An analysis yields the core business process diagram of the system, as shown in Figure 2. After registering and logging in, the user enters the system where they create their network node information upon successful login. The website then stores this information for future use and maintenance. Once the node is created, the system regularly checks the device status to ensure its proper functioning, notifying administrators if any issues are detected. Additionally, to secure the system, only authorized users can access and manage devices; therefore, during the login process, users undergo identity verification and authorization operations. If a device malfunctions or requires maintenance, administrators must promptly repair or replace it to guarantee the smooth operation of both the equipment and the system. Upon completion of these necessary maintenance and management tasks, the process ends. This entire sequence demonstrates the powerful functionality and flexibility of the Django framework, providing an efficient and convenient remote management platform for users.

The following is the website page design. For the user login interface, the implemented functions include username and password login, username and password logout, and username and password registration, etc. For the homepage, it should display the homepage upon opening the website, and be able to return to the homepage after performing any operation. For the navigation bar, the navigation should clearly list the device partition homepage, device information page, and device detail page. Finally, pay attention to the web page design style. The design of font style, size, color, etc. should facilitate reading needs. The layout of input and output pages should be uniform and standardized with clear layout. There should be corresponding prompts for operational errors.

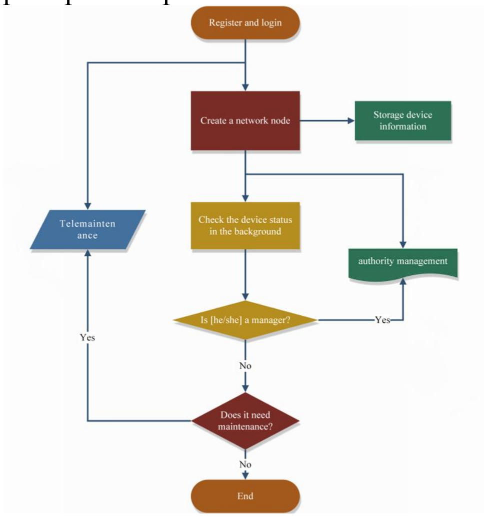

Figure 2. Core Business Flow Chart of the System

3. Overall Design Scheme

3.1 User Permission Design

The system is divided into two permission levels: administrators and maintenance personnel. Maintenance personnel can perform data entry, query data, and modify their own unapproved data entries. Once the data has been approved, the user cannot make further modifications. Administrators have the ability to enter, query, modify, and approve data. Data entered by administrators is marked as approved directly.

3.2 Database Table Design

The remote wireless meter reading website system includes multiple database classes, each with its corresponding data tables. They are as follows:

1) Records configuration parameters for device replacement and recovery.
2) Records liquid level values and pressure values based on current sensors.
3) Records pressure values and voltage values based on RS485 data.
4) Records the creation timestamp when a meter reading task is completed.
5) Records the update timestamp when meter reading data is received.
6) Records whether the specified meter reading tasks are complete.
7) Records the voltage value of the device's battery.
8) Records the temperature of the working environment.
9) he website should be able to view dial photos.
10) The website should be able to view environmental photos.

3.3 Data Analysis Design

In the process of database design, the key is to create a data model that accurately reflects the relationships between data and meets application requirements.

4. Front-end Engineering

Next, we will introduce the web front-end design languages.

Bootstrap is an open-source front-end development framework that includes HTML (Hyper-Text Markup Language), CSS (Cascading Style Sheets), and JavaScript frameworks [10]. DOM (Document Object Model) element event handling is divided into event capturing and event bubbling. jQuery is a cross-browser JavaScript library that simplifies the interaction between HTML and JavaScript. Cookies and sessions are two mechanisms used in web development to store user states. Cookies are stored in the user's browser, while sessions are stored on the server. They each have their advantages and disadvantages, such as security, storage size, and performance impact. Developers need to choose which to use based on specific requirements.

5. Back-end Development

This system uses Django as the framework. Django follows a comprehensive approach, and it is best known for its fully automated admin back-end: by simply using its ORM (Object-Relational Mapping) and defining objects, it can automatically generate database structures and a full-featured admin interface. Django offers extremely high development efficiency, but its performance scalability is limited; projects using Django often need to be refactored when traffic reaches a certain scale to meet performance requirements [11].

Django adopts the MTV (Model-Template-View) software design pattern, breaking down the software system into three parts: Model, Template, and View. The Model handles all data-related transactions, including data access, validation, and relationships between data behaviors and data itself. The View is primarily responsible for displaying data, managing model access, and correctly calling templates, serving as the link between models and templates. The Template embodies business logic, responsible for presenting relevant transactions and displaying data in an appropriate format on web pages or other types of documents.

The general process of using Django for development is as follows: First, you need to define the database model (Model) and input data into the view according to the corresponding view function in views.py; then, the view accepts the data from the model and inputs the data to be displayed into the template HTML file based on its requirements. Next, use the template to render the HTML page and add URL mappings if necessary. Finally, the controller part is responsible for handling user requests and responses; it receives requests from the URL and calls the corresponding view function based on the content of the request. Additionally, the controller also handles some extra logic, such as verifying user identity and handling errors, which flow path are shown in Figure 3.

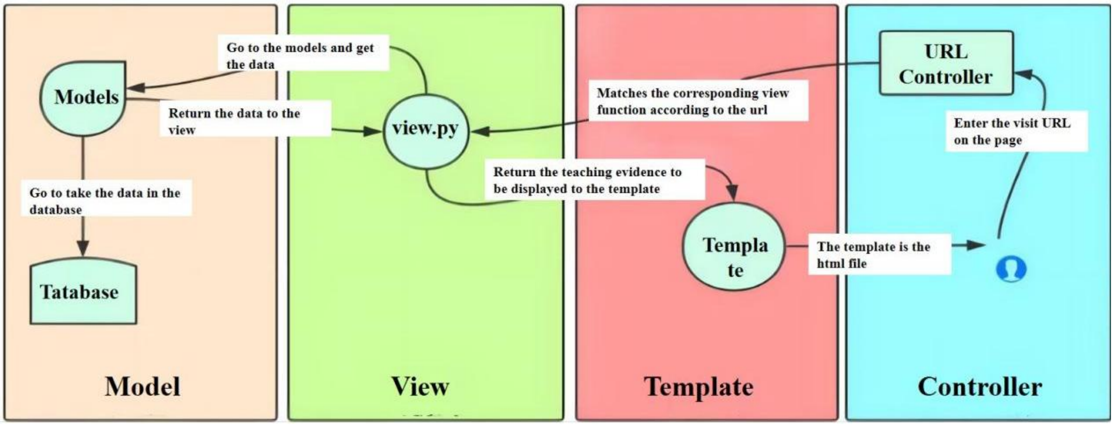
Go to the models and get the data
Models
Return the data to the view
view.py
Go to take the data in the database
Tatabase
Matches the corresponding view function according to the url
URL Controller
Enter the visit URL on the page
Return the teaching evidence to be displayed to the template
Templa te
The template is the html file
Model
View
Template
Controller

Figure 3. MTV (Model-Template-View) Model

Below are the main parts of the backend implementation, including region modules, messaging functionality, and device detail page functionality: create the Model, create the View, and create the template. Implementing features like user profile picture uploads and comments requires creating their respective data models. User registration and login can utilize Django's provided User model, with the registration and login process as shown in Figure 4. Upon user access, the system will first perform identity verification to determine if the user has already registered. If the user has registered, they will be directed to the next step directly; otherwise, they need to proceed with the registration process. For users who have not registered, the system will provide an "Link to register more face" option to guide them in completing the registration. After submitting user information, the system will send an authentication mail to the user's email address. The user needs to click on the link in the mail to activate the interface and enter their username and password, then jump to the login page. On the login page, users need to fill in the correct username and password for login operations. If the information is correct, the system will automatically jump to the homepage, completing the entire registration and login process.

Background management can create users and groups to distinguish different business divisions. As a user, you can have the permission in the group or a single permission. Custom action is a powerful feature of Django background management, allowing developers to expand the capabilities of background management.

Log files play an important role in network security. The log level indicates the severity of the message that the logger will process. Python Five log levels are defined. They are: DEBUG, INFO, WARNING, ERROR, and CRITICAL. These log levels can help managers filter and process based on the severity of the information [12].

6. Website Launch and Test

Cloud servers allow users to enjoy simple, efficient, secure, and reliable computing services with flexible scaling and processing capacity without purchasing hardware. This significantly reduces the difficulty of development and operations as well as overall

IT (Information Technology) costs. A virtual host is created on top of a physical server through software emulation, effectively sharing resources among multiple users.

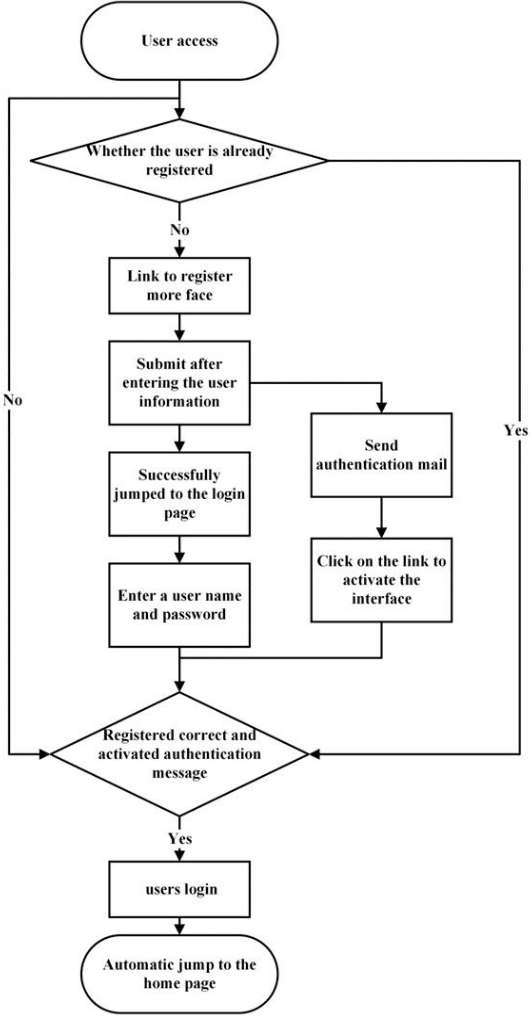
User access
Whether the user is already registered
No
Link to register more face
Submit after entering the user information
Send authentication mail
Successfully jumped to the login page
Click on the link to activate the interface
Enter a user name and password
No
Yes
Registered correct and activated authentication message
Yes
users login
Automatic jump to the home page

Figure 4. User Registration and Login Process

Next, apply for a cloud server, complete the primary assembly, and then install tools such as Python, Django, and PyCharm on a Windows 7 system to create a Django project and application, setting up the Web development environment.

Functional testing can then be conducted to ensure that the software system accurately implements the predetermined requirements of the system design. Functional tests for Web systems include link testing, form testing, cookie testing, and database testing. Link testing involves verifying whether all links direct to the correct target pages, ensuring the existence of linked pages, and checking for orphan pages within the Web system. Form testing verifies each input form against design specifications. Cookie testing checks the functionality of cookies, their lifespan, and the impact of page refreshes on cookies.

Subsequently, conduct live testing using the SSH protocol, which effectively prevents information leakage during remote management.

Performance testing follows, focusing on the system's response time, throughput, and resource usage (processor and memory). Performance tests are crucial for Web systems due to high user volumes and sudden traffic surges. Performance test includes three aspects: 1. Response time test.2. Load test: The load test checks how much load the Web system can bear at the same time under the premise of ensuring the normal service quality.3. Pressure test: Pressure test checks the performance of the system in the case of overload. The most important thing for stress test is to test some "hot spots" of the system, find the system bottleneck, such as login page, form input page, and so on, to improve the overall access speed of the system.

Finally, the usability test is divided into the following three aspects: First, in the interface design, the integrity of the interface is detected first, whether the use of visual elements is unified, whether the layout is consistent, and whether the location of information placement can be found through intuition. Second, check for good navigation. The second aspect is user compatibility testing, which tests for each popular browser and ensures that the application system has acceptable visual effects at a variety of different resolutions. Third, the security test, to ensure the server security, database security and application system security.

The comprehensive system testing process—encompassing functional, usability, performance, and security testing—ensures correct implementation of website subsystems, clear and attractive interfaces, a unified website style, support for simultaneous user access, and meets operational requirements, laying the foundation for subsequent business activities.

7. Conclusions

The engineering project in this article is based on the design of a remote wireless meter reading website system using the Django framework. Each module of the project was analyzed separately, including the design and implementation of the database, form design and implementation, configuration, template and view design and implementation, as well as comprehensive debugging and operation. The design and implementation of the project were ultimately completed. The debugging results show that the system can perform functions such as information browsing, registration, login, device node creation and maintenance, real-time page refreshing, and result statistics, achieving the expected goals satisfactorily. Given the complex nature of real-world issues and the substantial workload, automating operational management has become urgent for enterprises. By applying remote wireless meter reading system technology and introducing automated operational tools and techniques, operational efficiency can be significantly improved. At the same time, reduce human error to ensure the stable and reliable operation of the water information system.

Therefore, further development and optimization of the remote wireless meter reading system technology website and the implementation of automated operational management will be crucial directions for enhancing enterprise competitiveness in future developments.

Acknowledgment

This work is supported by the key project of Shenzhen science and technology plan under Grant No. JCYJ20220818103416035 and the Key-Area Research and Development Program of Guangdong Province under grant 2023B0909050006.

References

[1] Tong Sun, Fubin Zhang, Xiao Liu. Exploration of Water Governance Development Model and Enablement Mechanism Based on New Development Concepts. Water Resources Planning & Design, 2024, 9(03): 19-22+49.

[2] Z. Hong, J. Wang, S. Ma. Application Research of Wireless Communication Technology in Remote Meter Reading System. Applied Mechanics and Materials, 2014, 68(513-517): 1494-1498.

[3] H. Heng Z J, S. Yi Z. Design of Wireless Remote Water Meter for Automatic Meter Reading System. International Journal of Smart Home, 2015, 9(12): 289-298.

[4] F. Lei, Y. Chuang, W. Wenbin Z, et al. Android APP Development of Remote Wireless Automatic Meter Reading System based on 3G. International Journal of Online Engineering (iJOE), 2017, 13(02): 18-18.

[5] Ming Deng. Development Pattern and Intelligent Water Network System Construction of the Economic Belt on the Northern Slope of Tianshan Mountains. Arid Land Geography, 2020, 43(05): 1155-1168.

[6] Chuanlian Hu. Thoughts on Innovation of "Smart Water Network" Construction Management Model Based on the Internet Era. Water Resources Informatization, 2017, (04): 1-5+16. DOI: 10.19364/j.cnki.cn1674-9405.2017.04.001 .

[7] Zhi Cao. Analysis and Experience Enlightenment of Typical Cases of Domestic and Foreign Backbone Water Networks. Hunan Water Conservancy and Hydropower, 2023, (06): 19-21. DOI: 10.16052/j.cnki.hnslsd.2023.06.002.

[8] Huan Yao Hu, Wei Kui, Dongchuan Lu, et al. Research on Security Protection System Construction of E-Government Information Systems. Computer Times, 2022, (10): 147-149. DOI: 10.16644/j.cnki.cn33-1094/tp.2022.10.036.

[9] Hongxia Xie. Construction and Management of E-Government System Based on Computer Information Security Technology. Communications World, 2024, 31(08): 40-42.

[10] K. T. Mohd, J. Thompson, A. Carmine, et al. Comparative Analysis on Various CSS and JavaScript Frameworks. Journal of Software, 2022, 17(6): 282-291.

[11] D. Xuan, Y. Yuxuan B. A Pharmacy Drug Information Management System Based on Django Development. Academic Journal of Engineering and Technology Science, 2024, 7(4): 12-16.

[12] Juan Huang. Research on Automated Software Testing Platform Based on Python. Wireless Internet Technology, 2021, 18(23): 53-54.