ISSN: 2710-0588

Design and Implementation of Port Data Analysis and Visualization System

Taizhi Lv[1, *], Peiyi Tang[2]

[1] School of Information Technology, Jiangsu Maritime Institute, Nanjing, Jiangsu 211170, China

[2] R&D Department, Nanjing Huihai Transportation Technology Company Limited, Nanjing, China

*lvtaizhi@163.com

Abstract

Under the background of big data era, ports have realized the informatization of transport data, and has accumulated a large amount of transport data. The potential value contained in the port transportation data needs to be excavated, and a large number of external data also needs to be obtained. However, the traditional statistical analysis system can no longer meet the demand. In order to fully tap the potential value of port data, this paper designs and implements a port data analysis and visualization system to help enterprises master the trend of material flow. The system development is based on Python language. Pandas is used for data analysis, Flask is used to implement the MVC framework, ECharts is used for chart display, and data is imported from the business database to the MySQL database.

Keywords

Big Data Technology; Data Analysis; Data Visualization; Data Cleaning; Flask; Pandas.

1. Introduction

At present, the development of shipping enterprises is facing many difficulties. The continuation of the epidemic in recent years has seriously affected the development of port economy [1]. As a large port country in the world, China's port cargo throughput and container throughput have ranked first in the world for many consecutive years. In order to maintain stable development, port enterprises need to break the shackles of convention, create a platform-based business model, create a new business model, seize the opportunity of big data, make the shipping industry and big data technology perfectly integrated, and create higher value.

In order to give full play to the value of data, this paper designs and implements a port data analysis system. The system mainly presents the complex data in shipping information such as ports, goods and types of goods in different charts. Its main function is data processing, including data analysis and visual operation.

2. Requirement Analysis and System Design

2.1. Requirement Analysis

The trade between countries is growing, and the data volume of ports is growing. Data visualization technology can make data analysis more convenient, track the flow of goods, and provide data support for economic development, port management, etc. The system mainly presents the data of cargo in and out of the port in different chart forms, and its main function is to achieve data analysis and visual operation. In the visualization part, the data can be divided into owner unit, site number, bill of lading tonnage, actual tonnage, cargo type, day and night shift, etc. Various types of charts are important tools for data visualization, and large screens composed of various charts can help people to be very intuitive and concise.

The visual display of port data is to let users experience the changes of data more intuitively, and tell the viewers the relevant information of port cargo transportation through the data. The system has set up seven menu columns, which are realized by using the method of periodic playback to improve the user experience.

2.2. System Architecture

As shown in Figure 1, the architecture of this system is divided into two parts: the front-end and the back-end. The back-end provides interfaces, and the front-end uses asynchronous technologies such as Ajax to obtain data, and visualize them.

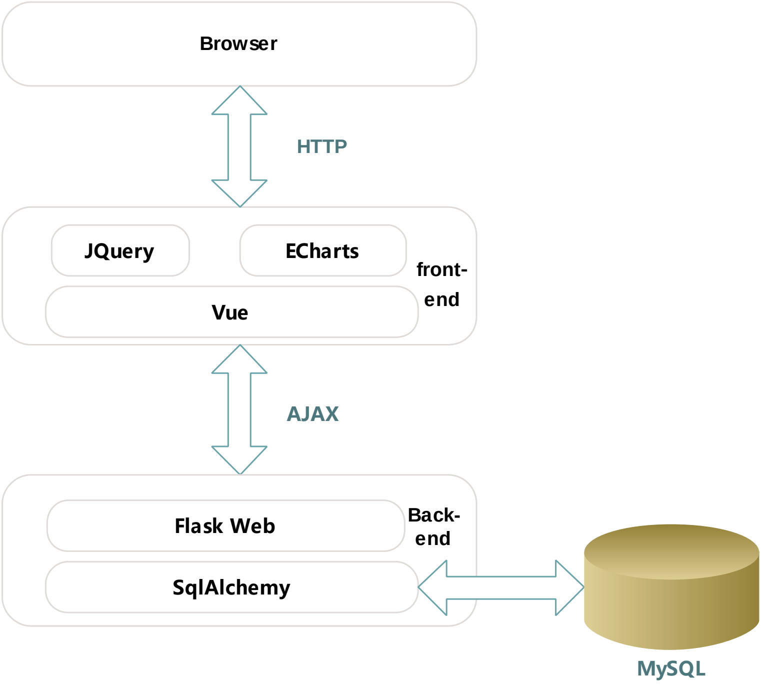

Figure 1. The system architecture

The back-end of the system uses Flask web development framework to write interfaces that can be called by the front-end for data display. FLASK web can be developed well in combination with MVC mode, and it enhances the stability and scalability of the system, reduce the development difficulty, and improve the development efficiency [2]. Based on the advantages of Pandas in the field of data analysis, the system uses Pandas for data analysis [3]. It reads the port data into the DataFrame object through Pandas, and analyze the port data based on the filtering and aggregation methods of DataFrame.

The front-end system uses AJAX technology to obtain back-end data. The JQuery package is introduced to implement AJAX calls, and ECharts is used for graph drawing. ECharts is a commercial data chart tool, which can provide intuitive, vivid, interactive and highly personalized data visualization charts for front-end development [4].

2.3. Functional Design

As shown in Figure 2, the port data analysis and visualization system consists of two modules: data management and data visualization. The data management module realizes data import, cleaning and backup management. The data visualization module is composed of seven views, which are broadcast by hiding and displaying views. The visualization chart makes statistics from multiple dimensions such as goods type, owner, and transport time.

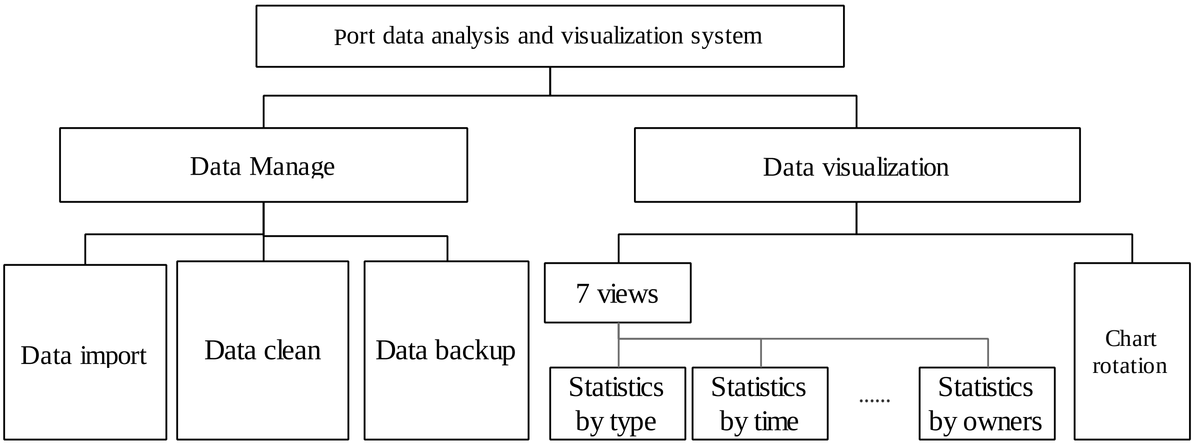

Figure 2. The functional module diagram

3. System Implementation

3.1. Back-end Implementation

In Flask framework, a mapping relationship is formed between each request and function. The function of handling the mapping relationship between Uniform Resource Locator (URL) and function is called routing [5]. For example, when a user's request path is type_count_bar and the HTTP request is got, the type_count_bar function is called automatically. It returns JSON data grouped by different goods types as required.

```
@app. route('/type_count_bar', methods=['GET'])
def type_ count_ bar():
```

The processing flow of this method is as follows:

Step 1: Query the cargo transportation table and cargo data classification table through SQLAlchemy grouping association.

Step 2: Implement cargo classification data through grouping function.

Step 3: store the goods classification name and corresponding loading quantity in the bar object;

Step 4: Convert the bar object to JSON object;

Step 5: Return the JSON object to the front end.

Other data interface processes and type of the system_ count_ Bar is similar. The system provides a total of 67 interfaces for front-end pages to obtain data. In order to simplify the operation of the database, FLASK_SQLAlchemy is used in the system development to operate the database. The database connection and the model object corresponding to the data table need to be initially configured.

3.2. Front-end Implementation

The front-end and back-end transmit data through JSON. Vue is used in front-end, and it binds the background data through the Ajax interface. Through the advantages of Vue's componentization and modularization, code duplication is reduced and development efficiency is improved. The overall page is divided into three parts: template, script, and style. The template tag stores html code to achieve page layout, and combines with the style code to set the page effect. Vue needs to configure the backend service address when binding the backend data interface [6]. The configuration method is as follows.

```
export default {
```

```
ServerHost: " http:// 120.27.151.49:8216/overview ",
axios: axios,
}
```

The statistical chart on the page is implemented by ECharts. The chart drawing process of ECharts is as follows.

Step 1: Write a DOM container with height and broadband, and display the chart and data in the container;

Step 2: Initialize the ECharts instance and bind the corresponding container;

Step 3: Specify the chart parameter configuration items, such as the chart title, color, style, and other information;

Step 4: Bind the corresponding data.

3.3. Rotation Chart Implementation

The page contains 35 statistical charts, which cannot be displayed all at once. In order to solve this problem, the system displays different statistical graphs in the form of carousel based on the time interval. Multi view layout is the basis of the carousel display. Multi view layout is to place the same view in the same location, with different ID numbers. The system designs a total of seven views, from divshow1 to divshow7. At the same time, only one of them is displayed. The initial default is divshow1, that is, the display attribute of divshow1 is set to flex, and the display style of other views is set to none. View carousel, in this case, is to switch the six views defined previously at regular intervals. One of the six views is displayed at regular intervals, and the other six are hidden. The first view is displayed at the beginning, and the other six are hidden. Then the second view is displayed, the third view is displayed, and so on. After the sixth view, the first view is displayed again. The whole process is divided into two steps, one is the initialization before the round robin, the other is to define the round robin function to switch views. The operation flow of the carousel function is shown in Figure 3.

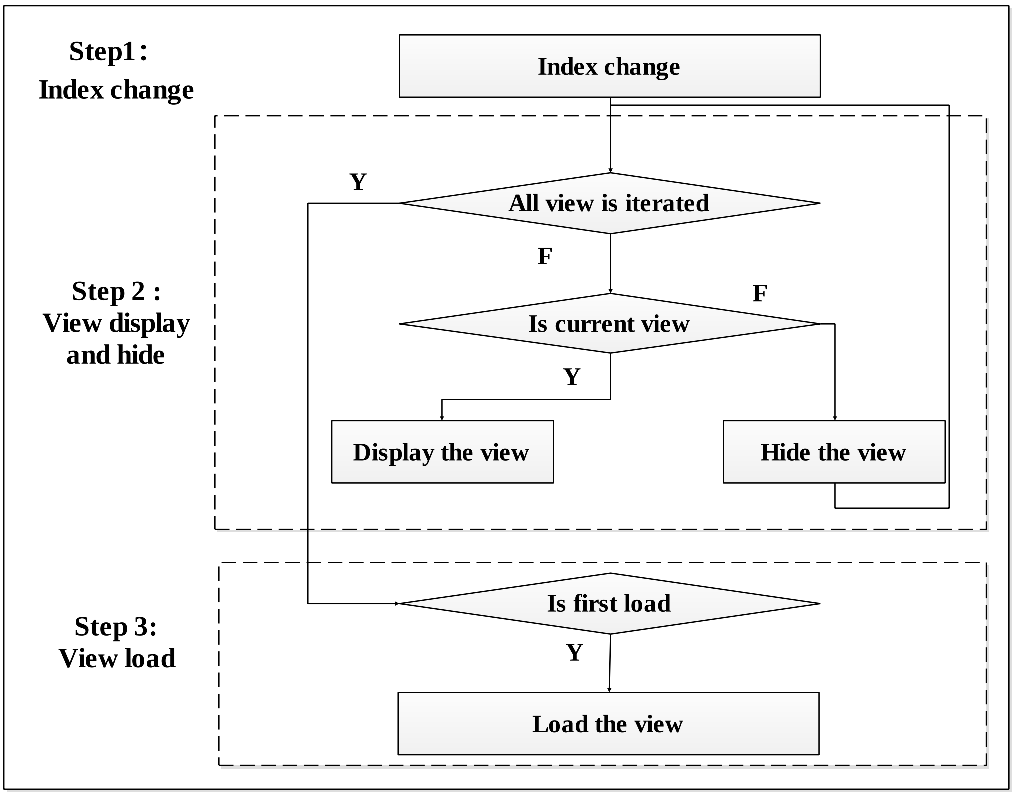

Figure 3. The process of rotation chart implementation

3.4. Implementation Effect

The visualization module is composed of 7 sub views, which make statistics on port data from each dimension. Figure 4 shows the first view ,which displays the statistics by year, quarter and month through radar chart and line chart. Figure 5 shows the second view, which displays the statistics of the number of day shift and night shift operations through a pie chart, and shows the annual freight volume of major enterprises through a histogram.

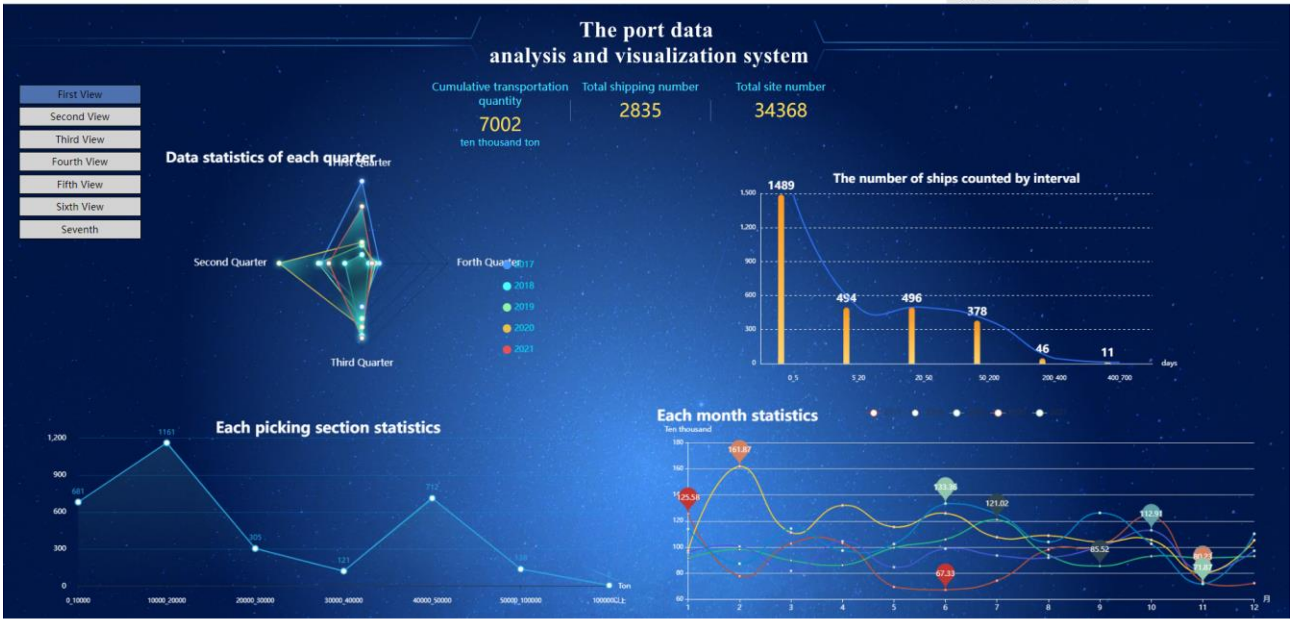

Figure 4. The first view

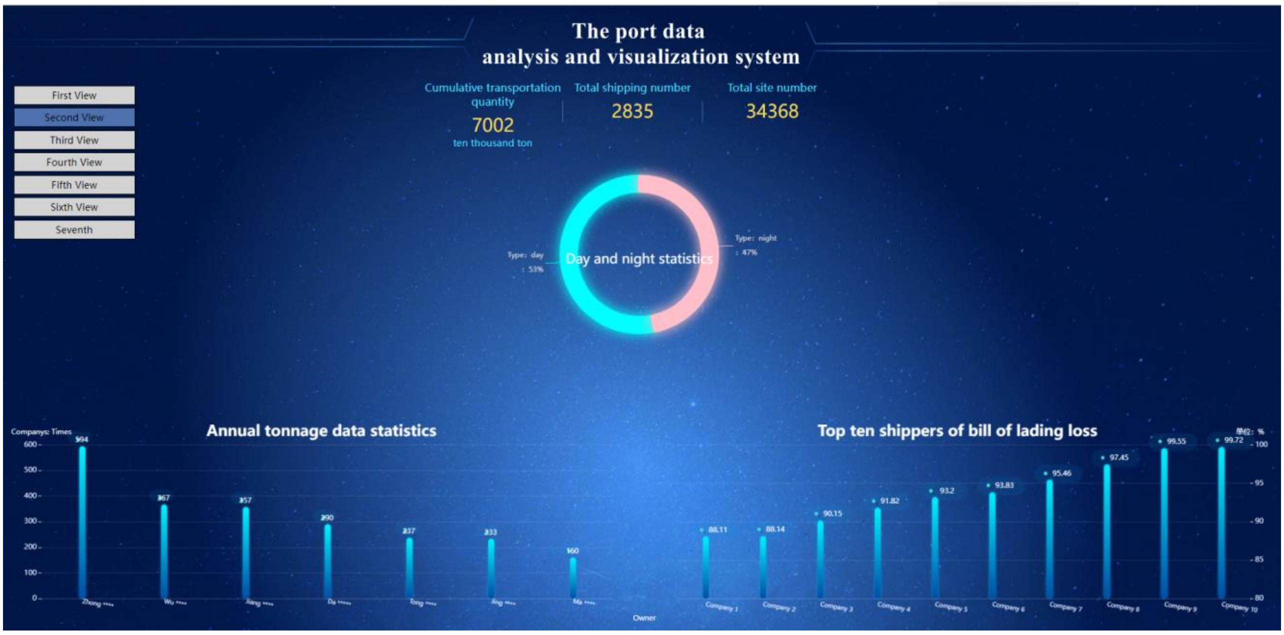

Figure 5. The second view

4. Conclusion

The system uses Flask framework technology and ECharts open-source visualization library to complete the port data analysis and visualization. The purpose of the system development is to better display the port resource transportation situation, and to visually analyze the port cargo classification. The system mainly displays the main page diagram of port data, the second page diagram of port data, the comparison diagram of day and night shifts, and the statistics diagram of the number of picking times of each cargo owner unit.

Acknowledgments

This work was financially supported by the Excellent Scientific and Technological innovation Team of the Jiangsu Higher Education Institutions of China (Maritime big data team), the Young Academic Leaders for QingLan Project of the Jiangsu Higher Education Institutions of China, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (21KJB580007).

References

[1] Panahi, Roozbeh, et al. "Developing a resilience assessment model for critical infrastructures: The case of port in tackling the impacts posed by the Covid-19 pandemic." Ocean & Coastal Management 226 (2022): 106240.

[2] Vogel, Patrick, et al. "A low-effort analytics platform for visualizing evolving Flask-based Python web services." 2017 IEEE Working Conference on Software Visualization (VISSOFT). IEEE, 2017: 109-113.

[3] Lemenkova, Polina. "Processing oceanographic data by Python libraries NumPy, SciPy and Pandas." Aquatic Research 2.2 (2019): 73-91.

[4] Li, Deqing, et al. "ECharts: a declarative framework for rapid construction of web-based visualization." Visual Informatics 2.2 (2018): 136-146.

[5] Wu, Mingxuan, et al. "Epidemic Data Visualization Surveillance Based on Flask." 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA). IEEE, 2022:1147-1151.

[6] Novac, Ovidiu Constantin, et al. "Comparative study of some applications made in the Angular and Vue. js frameworks." 2021 16th International Conference on Engineering of Modern Electric Systems (EMES). IEEE, 2021: 1-4.