

A community epidemic prevention and control system based on Vue

Bing Zhang

Department of Electronic Engineering, Tianjin University of Technology and Education, Tianjin, China

Email: 18439325891@163.com

How to cite this paper: Zhang, B. (2025). A community epidemic prevention and control system based on Vue. Journal of Computer Science and Frontier Technologies, 1(1), 63-79. ISSN Print: 3104-4204; ISSN Online: 3104-4212.

<https://doi.org/10.63313/JCSFT.9006>

Published: 2025-09-04

Copyright © 2025 by author(s) and Erytis Publishing Limited.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Abstract

The community is the basic unit of society. Since the outbreak of the novel coronavirus pneumonia, the community, as the front line of epidemic prevention and control, has been the main battlefield for preventing the spread of the virus internally and the importation of cases externally, and holds a fundamental position in the overall situation of epidemic prevention and control. Whether residents can be mobilized and communities organized to hold this line of defense well is related to the overall situation of defeating the epidemic. This paper combines modern Internet technology and the actual needs of community personnel, and designs and develops a community epidemic prevention and control management system by using Java technology, Vue framework and MySQL database. Compared with the traditional community system, the community epidemic prevention system can effectively record the information of community residents, strictly control the entry and exit of the community, ensure the health of residents, and thus achieve the effect of epidemic prevention. By using the functions of this system such as resident information management, visitor information management, daily entry and exit registration, health check-in, return home reporting, visitor registration, announcement management and domestic epidemic data statistics, it can further promote the participation of community residents in community governance, form a community governance pattern of co-construction, co-governance and sharing, effectively reduce the burden on community workers, and strengthen community epidemic prevention and control.

Keywords

Community management Epidemic prevention and control; Java; MySQL

1. Introduction

With the wide application of the Internet and the continuous development of information globalization, information technology is playing an increasingly important role in China's market economy and People's Daily lives. Nowadays, street communities need effective management. However, due to the expansion of China's market economy and the change of people's lifestyles, traditional community management

can no longer meet people's needs. However, some community offices have installed community system software and are staffed with dedicated managers. However, in practical applications, there are also some deficiencies, such as differences in basic information, data sharing, and incompatibility between systems, which lead to repetitive low-level management, high investment, low efficiency, and residents being unable to log in to the system to view their own information. All these have hindered the effective implementation of community management work and failed to enable the community management level to develop better and faster. The network-based community management system can change the previous community management model and achieve the informatization and efficiency of community management.

2. Research Context

2.1. The purpose and significance of the topic selection

With the popularization of the Internet and the improvement of information technology, community epidemic prevention has played a key role in the control of the novel coronavirus. To reduce the workload of community epidemic prevention personnel, it is necessary to utilize the technological power of the Internet to add a helping hand to community epidemic prevention and control and develop a complete community epidemic prevention system.

Meanwhile, the urban population is increasing, and the number of community residents is also growing. The traditional community resident information management system has encountered many new problems. Moreover, in the face of the outbreak of the epidemic, there is an urgent need to develop the latest community epidemic prevention and control management system. By analyzing the business of community residents' information management and combining the actual situation of community residents' information, a detailed and in-depth analysis of the requirements for the community management system is conducted. Drawing on the practices and experiences of promoting the informatization of community residents in various places, a preliminary design and discussion of the system are carried out, which is conducive to the construction of the community epidemic prevention system and facilitates the management of community residents' information. Only efficient community services can meet the needs of residents. Therefore, the scientific and information-based management of communities has been regarded as an important indicator of improving people's living standards. In the past, residents had to enter and exit the community office one after another. Now, due to the pandemic, the lifestyle and working methods of community residents have changed. Residents need increasingly high standards of community services, and the community management model and service quality also urgently need to be improved, which can be accomplished through modern information technology.

2.2. Current research status at home and abroad

In recent years, with the outbreak of the novel coronavirus both at home and abroad, community epidemic prevention and control management has become an important part of the country's epidemic prevention and control efforts. According to research and investigation, compared with foreign countries, China pays more attention to community epidemic prevention and control measures and actively builds community epidemic prevention and control management systems. Some large and medium-sized software companies in developed countries abroad have also started to develop community epidemic prevention and control systems. Their development technologies are more mature than those in China. They have made reasonable arrangements for the management information of community personnel and have created many grid-based epidemic prevention and control management systems based on communities, which are also highly favored.

From the current community epidemic prevention and control management system, there are still some problems, mainly including the following aspects. (1) Single function. The system's functions are not yet perfect. Despite the ongoing domestic epidemic, the demand has also increased accordingly. (2) Poor flexibility. The number of domestic communities is huge, but the system coverage is relatively weak and the functions are not comprehensive enough. In terms of the use of the software, such as community members purchasing daily necessities, daily health check-ins, and reporting of outsiders, the elderly and children have not been taken care of. (3) Low popularity. Most of the community epidemic prevention and control systems developed have not been widely applied to the management and control of community residents.

3. Development technology

3.1. Front-end and back-end separated architecture design

In the early days of the Internet, pages were relatively simple and most of the focus was on back-end technology. With the development of business and technology, the front-end has become increasingly complex and its functions have become more and more rich. Front-end and back-end separation means splitting a single application into two independent applications and conducting data interaction in Json format.

3.2. Java technology

Java is an object-oriented programming language that not only absorbs the advantages of the C++ language but also discards the difficult-to-understand concepts of C++. Therefore, the Java language has two major characteristics: powerful functionality and ease of use. Java can be used to write desktop applications, Web applications, distributed system and embedded system applications, etc.

3.3. Vue technology

The system design adopts the front-end and back-end separation technology, with the front-end using the Vue framework. Vue is a framework based on the MVVM

concept. It encapsulates JavaScript at the bottom, eliminating the complex operation of apis in JS. It is merely a core server based on JavaScript. Its core content is data-driven and componentization. Compared with the traditional model, Vue achieves two-way data binding by changing the model through listening. Componentization is the implementation of extending HTML elements and encapsulating usable code.

4. System analysis and design

4.1. System Requirements Analysis

Based on the business requirements of users, the functions of the community epidemic prevention and control system have been summarized as including resident information management, daily entry and exit registration, health check-in, return home reporting, visitor information management, visitor registration, announcement information management, and domestic epidemic data statistics, etc.

Non-functionality refers to the consideration of the system's security, reliability, ease of use, maintainability, portability and scalability, in addition to the user's required functions, so as to meet the various needs of customers.

4.2. System Outline Design

The epidemic prevention and control system of this community adopts the idea of gathering the big with the small, and takes structural hierarchical management as the design mode. It is developed and designed in modules, and the code of each module is written one by one to realize its functions. Its main functional modules are divided into three categories: the administrator module, the resident module and the visitor module. The main functions corresponding to the resident module include viewing resident information, daily entry and exit registration, health check-in, reporting for returning home, viewing announcements and checking the latest domestic epidemic information, etc. The main functions corresponding to the visitor module are visitor registration, viewing announcement information and checking the latest domestic epidemic information. The main functions of the administrator module are user management and the allocation of role permissions, etc. Its functional structure is shown in Figure 1.

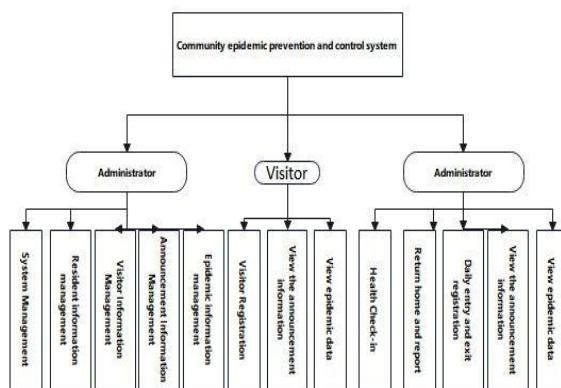


Figure 1. System Function structure diagram

4.3. System Outline Design

To meet the needs of residents, the design of this system has been modernized and redesigned compared with the traditional community management model. After a user logs into the system, the system first determines the user's login identity and role permissions. When the system determines that the currently logged-in user is an administrator, the administrator can modify the user's information, assign roles and permissions to the user, and also has functions such as resident management, visitor management, announcement issuance, and obtaining and displaying epidemic data. When the system determines that the current logged-in user is a resident, the resident has functions such as viewing resident information, daily entry and exit registration, health check-in, reporting for returning home, checking announcements and viewing epidemic data. When the system determines that the currently logged-in user is a visitor, the visitor has functions such as viewing visitor information, visiting registration, viewing announcements, and viewing epidemic data. The system divides functional modules based on the different roles of users.

5. Database design

5.1. Database Requirements Analysis

Based on the analysis of community management work under the current epidemic environment and in combination with our own research situation, determine the role types and corresponding permission functions required by the system. This includes user login and registration, allocation of user role permissions, resident information, daily entry and exit registration of residents, health check-ins, reporting for returning home, visitor information, visitor registration information, announcement information, and domestic epidemic information, etc. After conducting a survey on the needs of different residents in the community, a reasonable database design was carried out. The main data information is as follows:

Resident information: Mainly stores all the information of residents, including basic data items such as resident number, name, gender, age, mobile phone number, and health status.

Visitor information: Mainly stores the basic information of visitors, including visitor number, resident number, name, gender, mobile phone number, body temperature and other basic data items.

Community information: Mainly stores the information of the community, including basic data items such as community number, community name, and community address.

Building information: Mainly stores the building information within the community, including basic data items such as building number, community number, building

number, unit number, and room number.

Resident entry and exit information: Mainly stores the data information of residents' daily entry and exit in the community, including basic data items such as entry and exit numbers, resident numbers, names, genders, mobile phone numbers, body temperatures, entry and exit status, reasons for entry and exit, and entry and exit times.

Health check-in information: Mainly stores the health information of residents, including check-in number, resident number, name, gender, age, mobile phone number, body temperature, whether confirmed, whether visited medium and high-risk areas, and other basic data items.

Return home reporting information: Mainly stores residents' return home information, including basic data items such as reporting number, resident number, name, gender, mobile phone number, type of identification, identification number, health code, negative nucleic acid test result, picture path, place of origin, destination, means of transportation, and arrival time.

5.2. Conceptual structure design of database

After analyzing and designing the community epidemic prevention and control system, this system mainly consists of seven entities, namely "residents", "Visitors", "residential areas", "buildings", "Residents' Entry and Exit", "Health Check-in" and "Return Home Reporting". The relationships among these entities are as follows: One resident can be visited by multiple visitors, but one visitor can only visit one resident, so the relationship between "resident" and "visitor" is one-to-many. A residential community can accommodate multiple residents, and multiple residents can live together in the same community. Therefore, the relationship between "community" and "residents" is many-to-one. A residential community consists of multiple buildings, and multiple buildings are located within one community. Therefore, the relationship between "community" and "buildings" is also one-to-many. A resident can enter and exit the community multiple times a day. Each resident needs to fill out an entry and exit registration form when entering or leaving the community. Therefore, the relationship between "residents" and "residents' entry and exit" is one-to-many. Every resident in the community needs to complete a health check-in once a day, and the health check-in needs to be submitted by all residents every day. Therefore, the relationship between "residents" and "health check-in" is one-to-many. Every resident returning home from outside needs to log in to the system in advance to report. The reporting for returning home requires multiple residents to fill in the reporting information. Therefore, the relationship between "residents" and "reporting for returning home" is one-to-many.

5.3. Database logical structure design

Based on the drawn entity relationship diagram, the logical structure design of the community epidemic prevention and control management system can be derived:

Resident information (resident number, community number, name, gender, age, mobile phone number, health status, building number, unit number, room number)

Visitor information (visitor number, resident number, name, gender, mobile phone number, body temperature, visit time, visit reason, departure time)

Community information (community number, community name, community address)

Building information (building number, community number, building number, unit number, room number)

Residents' entry and exit (entry and exit number, resident number, name, gender, mobile phone number, body temperature, entry and exit status, reason for entry and exit, entry and exit time)

Health Check-in (Check-in number, resident number, name, gender, age, mobile phone number, body temperature, whether confirmed, whether visited medium and high-risk areas, others)

Return Home Report (Report Number, Resident Number, Name, Gender, Mobile Phone Number, Type of document, document Number, Health Code, Negative Nucleic acid Test result, Picture Path, Place of Origin, Destination, Means of transportation, Arrival Time)

5.4. Database physical structure design

By designing and analyzing the logical structure and transforming it into a table in the database, storing each attribute with an appropriate type and length, and setting the primary key and related constraints, the design of the database table can be completed. A total of ten tables were designed based on the system, including "User Table", "Role Table", "Menu Permission Table", "Resident Information Table", "Visitor Information Table", "Community Information Table", "Building Information Table", "Announcement Information Table", "Resident Entry and Exit Information Table", "Resident Health Check-in Information Table" and "Resident Return Home Reporting Information Table". The following mainly introduces the table structures of the "Resident Information Table" and the "Visitor Information Table".

The resident information table is mainly used to store the basic information of all resident users. Among them, the resident number serves as the primary key of the resident information table, and the community number is the foreign key. The main fields are shown in Table 1.

List	Data type	is empty ?	Primary key or not	Description
rid	<u>varchar</u> (100)	null	yes	Resident number
lid	<u>varchar</u> (100)	null	no	Community number
name	<u>varchar</u> (100)	null	no	name
sex	int(5)	null	no	gender
age	<u>varchar</u> (50)	null	no	Age
phone	<u>varchar</u> (50)	null	no	Telephone number
state	int(5)	not null	no	Health status
building	<u>varchar</u> (255)	null	no	Building number
element	<u>varchar</u> (255)	null	no	Unit number
room	<u>varchar</u> (255)	null	no	Room number

Table 1. Resident Information Form

The main function of the visitor information table is to store the information of visitors. Among them, the visitor number is the primary key of the visitor information table, and the resident number is the foreign key. The main fields are shown in Table 2.

list	Data type	Is empty?	Primary key or not	Description
<u>vid</u>	<u>varchar</u> (100)	null	Yes	Visitor Number
rid	<u>varchar</u> (100)	null	no	Resident number
<u>vname</u>	<u>varchar</u> (100)	null	no	name
<u>vsex</u>	int(5)	null	no	Gender
<u>vphone</u>	<u>varchar</u> (50)	null	no	number
<u>vtemp</u>	<u>varchar</u> (50)	not null	no	Body temperature
<u>vreason</u>	<u>varchar</u> (255)	null	no	Reasons for the visit
<u>vintime</u>	<u>datetime</u>	not null	no	Visiting time
<u>vouttime</u>	<u>datetime</u>	not null	no	Departure time

Table 2. Visitor Information Table

6. The specific implementation of the 5 system functional modules

This system mainly has three users: administrators, residents and visitors. The administrator module has system management functions, including login and registration, password modification, role and permission assignment for users, resident information management, visitor information management, announcement management, and collection and display of epidemic data, etc. The resident module has functions such as daily entry and exit registration, health check-in, return home reporting, announcement viewing, and viewing domestic epidemic data. The visitor module features functions such as visitor registration, viewing announcements, and

checking domestic epidemic data. The following takes the functional modules such as "Resident Entry and Exit Management Module", "Resident Health Check-in Module", "Resident Return Reporting Module", "Visitor Information Management Module", "Announcement Information Management Module", "Resident Information Management Module" and "Epidemic Information Management Module" as examples to explain their functions and implementation processes.

6.1. Resident access management module

(1) Function description

To strengthen the management of entry and exit, the community strictly enforces the control measures of the residential area, advocates that people do not go out unless necessary, and those entering the community need to scan the code for registration. Residents need to fill out the resident entry and exit registration form when entering or leaving the community, which records the daily entry and exit of people. Administrators can manage these entry and exit information and perform operations such as adding, deleting, modifying and querying.

(2) Function realization

When resident users log in to the system, they will see the daily entry and exit registration TAB in the resident management module. After residents fill in the entry and exit registration information and click the "Submit" button, the front desk will send a new entry request for all parameters using the `$axios.post` method through the click event. The mapping path will be annotated through the `@PostMapping` of the `ResidentComeController` in the background. After verifying the `PreAuthorize` permission, call the `save()` method to complete the registration operation. When the administrator clicks on the "Resident Access Information Management" Tab, the background automatically calls the `list()` method of `ResidentComeController` and performs pagination through the `IPage()` method, rendering the view and returning it to the foreground. When the "Modify" button is clicked, a modification prompt box will pop up on the front end. The selected information will be reflected in the prompt box. After the administrator modifies the information and clicks the "Submit" button, the `update()` method in the background will be called to perform the modification operation. When the "Delete" or "Batch Delete" button is clicked, the system will first determine whether the ID exists. When there are multiple ids, the front end will place these ids in an array and pass it to the back end. The back end takes the array as a parameter and calls the `removeById()` method to perform deletion or batch deletion operations.

6.2. Resident health check-in module

Function description

To ensure the prevention and control of the epidemic and the health and safety of the residents in the community, they need to fill in their basic information every day and report their health conditions. Administrators will back up and manage the in-

formation reported by residents. At the same time, they will automatically check the basic information status of residents. The system will determine the daily reported body temperature, health status, physical condition of co-residents and whether they have visited medium and high-risk areas. If any abnormal information occurs, the system will notify the management personnel to pay close attention to this group of people and take measures such as continuous nucleic acid testing or home isolation. (2) Function realization

Resident users need to enter the resident management module to complete the health check-in. After clicking on the "Health Check-in" Tab, the front desk will display the community resident health reporting form. Residents should fill in their real information completely. When residents click the "Submit" button, the front desk will send a new request for inclusion in the database via post and bring along the filled content. Go to the background PersonDayController to call the save() method, and then enter the database by calling the save() method of the personDayService. Administrators have the authority to query, modify and delete residents' health check-in information. When the "Health Check-in Information Management" TAB is clicked, the back end maps the path through the @GetMapping() annotation in the control layer, calls the list() method, uses the public Page() method for pagination query, and finally renders the view and returns it to the front end. When the administrator clicks the "Modify" button, the backend will determine whether the health check-in number exists. If it does, it can be modified; otherwise, it cannot be modified. After modifying the information, click "Submit", which will call the update() method on the back end. After the modification is completed, it will echo back to the front page. When the "Delete" or "Batch Delete" button is clicked, the system will first determine whether the ID exists. When there are multiple ids, the front end will place these ids in an array and pass it to the back end. The back end receives the array and calls the deletion method to perform deletion or batch deletion operations.

6.3. Resident return home Reporting module

Function description

Recently, the domestic epidemic situation has shown a trend of sporadic cases in multiple locations, and the number of provinces and cities with positive cases has gradually increased, which has brought considerable pressure to epidemic prevention and control in various regions. In order to effectively reduce the risk of epidemic spread, many places have adopted relatively strict epidemic prevention and control policies. It is required to report to the community for returning home three days in advance and prepare a nucleic acid test certificate within 48 hours. When community residents log in to the system, they will see the community announcement information, which requires residents to report three days in advance before returning home. Residents need to fill in the information for reporting their return to their hometowns in the menu under the resident management, and the administrator will manage this information.

(2) Function realization

After residents successfully log in to the system, they will see a rather prominent reminder on the system's home page, which reads, "In accordance with the requirements of epidemic prevention and control, report to the system three days in advance." Residents fill in the basic information, travel information and pictures of relevant nucleic acid test reports on the return home registration form and click the "Complete" button. The front desk transmits all the information to the background through the path via the post request of \$axios. The background maps the path through the @PostMapping() annotation. After the PreAuthorize permission verification is performed again, the save() method of the corresponding control layer is called to add new items to the database. Administrators manage the information reported for returning home and can modify, delete and query it. When the administrator clicks the "Modify" button, the backend will determine whether the return home reporting number exists. If it does, it can be modified; otherwise, it cannot be modified. After modifying the information, click "Submit", which will call the update() method in the background. After the modification is completed, it will echo back to the foreground page. When the "Delete" button is clicked, the system will first determine whether the ID exists. Only when it does can the removeById() method be called to perform the deletion operation. In the resident management drop-down menu, click on the "Return Home Reporting Information Management" Tab to directly view the list of return home reporting information. First, the front end will pass in the url using the \$axios.get() method, and the back end will parse the path through the @GetMapping() annotation and verify the permissions with @PreAuthorize. After verification is successful, the list() method will be executed for viewing.

6.4. Visitor Information Management module

Function description

Although visitors are not internal members of this community, they can enter and exit it. To prevent potential risks of the epidemic, the system will conduct statistics and management on the information of external personnel entering and leaving this community. For the convenience of future flow investigation and verification, visitors need to go through the visitor registration service before entering the community. Visitors need to fill in the registration form content and submit it before they can enter the community to visit friends. Administrators have the management operations to add, delete, modify and query visitors' information. Visitors themselves can also view the registered information, but they can only view it and have no permission to modify or delete.

(2) Function realization

The visitor first clicks on the "Visitor Registration" Tab. After filling out the visitor information registration form, they click "Submit". The front desk passes in the url through the \$axios.get() request method. After the server receives the request in-

struction, Based on the @GetMapping() of the visitor information management control layer VisitorInfoController, the path of the mapping file is queried. After the PreAuthorize permission verification is performed, the visitorInfoService.save() method is called. Since the VisitorInfoService layer directly inherits from the IService service, at this time, the server will go to the VisitorInfoController layer to look for and call the save() method within the system to perform the database addition operation. Complete the registration operation for visitors before they enter the community, ensure the personal information registration and temperature registration of people visiting the community, and add an extra layer of protection for epidemic prevention and control. Administrators can also add, modify, delete and query the information of visitors. When the "Add" button is clicked, the operation is the same as the method called for the registration form submitted by the visitor above for database entry. When you click "Modify", the backend will receive a visitor number that needs to be modified. Determine if it exists and then use the update() method to make the modification. When you click "Delete", the front end will first prompt whether you want to confirm the deletion. After clicking "OK", the back end will call the system's built-in removeById() method through the ID passed over to perform the deletion operation. The administrator can directly view the list of visitor registration information by clicking on the "Visitor Information" Tab in the visitor management drop-down menu. First, the front end will pass in the url using the \$axios.get() method, and the back end will parse the path through the @GetMapping() annotation in the ResidentComeController @PreAuthorize is used for permission verification. Once the verification is successful, the list() method will be executed, and then the public Page class will be used for pagination viewing based on the passed parameters.

6.5. Announcement information management module

Function description

In order to implement the relevant epidemic prevention policies stipulated by the national government, the system will issue announcements within the community. Announcement information is a notice issued by the community to residents and visitors, informing them of the latest regulations related to the epidemic. The administrator inputs the title of the announcement, the content of the announcement and the release time, and clicks the "Publish" button to complete the release of the announcement information. At the same time, they can also delete or modify the announcement in the announcement information list. Residents and visitors can browse the list of announcement information through the announcement information management module.

(2) Function realization

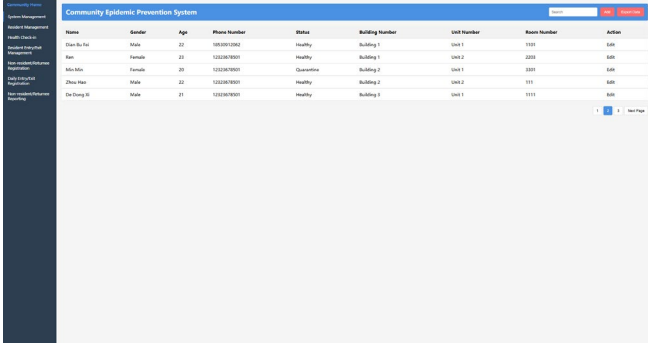
After the administrator fills in the announcement title, content and release time, they click the "Publish" button. The front-end transmits the received data information to the back-end by clicking the event using the post request method of \$axi-

os. When the url of the request instruction is received in the background, the path of the mapping file is queried based on the @PostMapping annotation of the NoticeController in the announcement information management control layer. After verifying the PreAuthorize permission, the saveOrUpdateNotice() method is called. Its addition and update use the same method. The program will first determine whether the announcement number is empty. If it is empty, it will receive the parameters sent from the front desk and add them to the database. If it is not empty, the modification method will be executed. The getById() method of noticeService will be used to query this record, update the received parameters, and finally return the result. Administrators delete announcements by clicking the "Delete" button. After the background receives the deletion request, it calls the delete() method of NoticeController and executes the removeById() method based on the passed announcement code to delete the corresponding number of announcements. Residents or visitors enter the announcement module, click on the "Announcement Information" Tab, and the system automatically calls the noticeList() method of the NoticeController. This method uses the noticeList() method of the noticeService to pass in the DTO parameter. The noticeList() method uses the conditional constructor QueryWrapper() method of Mybatis Plus to query the value of the corresponding entity class, and performs pagination processing through the system's built-in Page class to obtain the result set of the data. After rendering, it returns to the foreground page.

6.6. Resident information management module

Function description

The most important task of the community is to manage the residents' information, collect the basic information of all residents in the community, including local residents and tenants, and uniformly manage the basic information of the residents to understand their basic situations. The resident information management module is equipped with a resident information management function module, which manages the basic information of residents, their daily entry and exit information, and the reporting information for returning from other places. Administrators can perform operations such as adding, deleting, modifying and querying residents' information, while residents can only view it and are not allowed to add, modify or delete. The list of its resident information is shown in Figure 3.



Name	Gender	Age	Phone Number	Status	Building Number	Unit Number	Room Number	Action
Chen Bu-Fei	Male	27	18828012862	Healthy	Building 1	Unit 1	1101	edit
Ren	Female	28	13328078901	Healthy	Building 1	Unit 2	2202	edit
Wu Xue	Female	26	13328078901	Quarantined	Building 1	Unit 1	2201	edit
Zhou Xue	Male	22	13328078901	Healthy	Building 1	Unit 2	111	edit
Gu Chang-Bi	Male	27	13328078901	Healthy	Building 1	Unit 1	1111	edit

Figure 2. List of residents' information**(2) Function realization**

After successfully logging into the system, the administrator manages the resident information module. Administrators can perform operations such as adding, deleting, modifying and querying the list of residents' information. When the "Add" button on the page is clicked, a new input box will pop up. Fill in the corresponding information. After clicking the "Submit" button, the front end will call this.\$axios.post() to pass in the path corresponding to the new operation. The back end will parse the url through the @PostMapping() annotation before performing the corresponding new operation and entering it into the database. The modification operation is similar to adding. When the "Modify" button is clicked, the front end will first fill in the information that needs to be modified in the pop-up box. After modifying the content and clicking "Submit", the back end will determine whether it exists based on the ID. If it exists, the submitted form will be modified; if not, the addition operation will be executed. Both deletion and batch deletion operations are determined by the ID. When the "Delete" button is clicked, the front end will remind you whether you really want to delete. After clicking "OK", the front end will pass an ID to the back end. When nothing is selected, the "Batch Deletion" button is in an unclickable state. Only when one or more pieces of information are selected will the "batch Deletion" operation be triggered. After multiple ids are selected, the front end will pack these multiple ids into an array and pass it to the back end. The back end places one or more ids passed in an array and calls the removeByIds() method of basicResidentService to delete the passed ids, thus achieving the deletion operation. Residents can view the resident information within their own community. When residents click on the "Resident Information Management" Tab, a list of resident information is loaded. The front end calls the getResidentList() method and passes in the url call through the get request of \$axios. The back end first parses the url with the @GetMapping annotation, then verifies the PreAuthorize permission, and executes the list() method for query. Since BasicResidentService inherits from the IService service, queries are made through the QueryWrapper method of the MybatisPlus framework, and pagination is done through the page() method of the basicResidentService parent class. Ultimately, the list of resident information is displayed. Here, residents can only query and cannot add, modify or delete.

6.7. Epidemic information management module**Function description**

To ensure the health and safety of residents, this system will release the latest, most comprehensive and accurate domestic epidemic data, presenting the most comprehensive data in the most intuitive form. After residents or visitors log in to the system, they can grasp the current domestic epidemic information. It also supports zooming in or out and downloading functions. The epidemic data chart is shown in Figure 4.

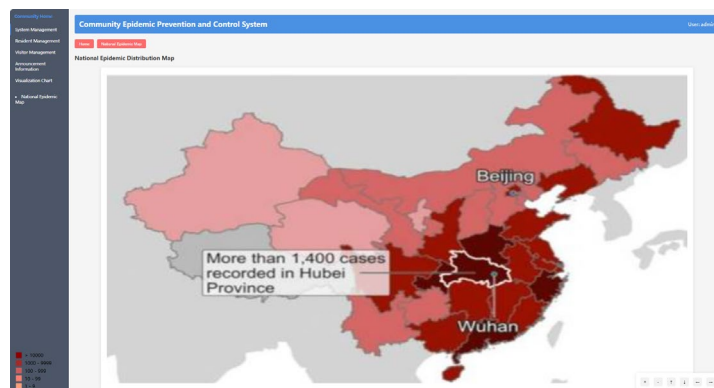


Figure 3. National Epidemic Data Chart

(2) Function realization

First, the front-end Vue framework needs to introduce the Echarts plugin and the Jsonp plugin. Use Echarts' `map:'china'` to obtain the map of China, control the Chinese characters corresponding to the region, control the font color of the region name, control the background color and font color after the mouse slides over, and change the color intensity according to the number of confirmed cases. Finally, Jsonp is used to parse Json and solve the cross-domain problem of browsers to obtain the latest epidemic Data statistics from Sina's official website. The obtained data is received using `Data[]` and transmitted to the front desk.

7. System Testing

7.1. Test Purpose

The purpose of system testing is to verify whether the complete system configuration can be correctly connected to the system through the operating environment combined with the system's requirements design, and to identify any inconsistencies or contradictions between the system and the design documents and software development regulations. Only by conducting thorough tests can the defects of the system be discovered, corrected in a timely manner, and the quality of the software be improved.

7.2. Test method

To ensure the normal operation of the system's functions, this system adopts a functional testing method, and the passed test cases are used for testing. The main task is to check whether the software functions meet the users' demands. It is necessary to take into account the details of each function and is the most time-consuming testing activity in the testing work. The following tests are conducted on several core functions by module, as shown in Table 3.

Test content	Testing process	Expected results	Actual result
Login function	Enter the user name: admin, password: 111111	Login successful	Login successful
	Enter username: 123, password: 123	Login failed	Login failed
Resident health check-in function	Enter the resident's health information and click Submit	Submitted successfully	Submitted successfully
Visitor registration function	Enter the registration information of the visiting personnel and click Submit	Registration successful	Registration successful
Announcement release function	Enter the title and content of the announcement, and click Submit	"Published successfully"	"Published successfully"

Table 3. System Function Test Cases

8. Conclusion

This paper designs and implements a community epidemic prevention and control system based on front-end and back-end separation according to the needs of different users.

The system has realized functions such as resident information management, visitor information management, health check-in, return home reporting, visitor registration, and announcement management. It has greatly reduced the work intensity of community workers, improved their work efficiency, and also ensured the health of residents and visitors. Although the system has completed the requirements and design, there are still some deficiencies in its development, such as how it manages abnormal body temperatures and confirmed cases in the community, and the interface is not aesthetically pleasing. In the future learning process, the system will be upgraded to make its functions more complete.

Acknowledgment

We are grateful to Zhang Bing and Zhang Zhixin. This work was partially funded and supported by Li Shixin.

References

- [1] Zhu Yi, Ye Feng The Design and Implementation of the "Home Health House" Epidemic Isolation Management System [J] Geospatial Information, 2020,18 (04) : 5-10+149.
- [2] Wu Baoyou The Design and Implementation of Community Grid-based Epidemic Prevention Information Service Platform [J] Beijing Surveying and Mapping, 2020,34 (06) : 735-740.
- [3] Zhao Yue Design of Community Emergency Management System for Public Health Emergencies: A Case Study of the Novel Coronavirus Pneumonia Incident [J] Beijing Surveying and Mapping, 2020,34 (06) : 731-734.
- [4] Wang Zhen Design and Implementation of Intelligent Data Application for Smart Com-

- munity Information System and Comprehensive Information Command System for Grassroots Governance [J] Computer Times, 2020, (08):82-84.
- [5] Xia Wan, Li Quan, Zhang Jingjing. Construction of Smart Community Management System under the Epidemic [J] Information Technology and Informationization, 2020, (08) : 67-70.
 - [6] Guan Saihui, Bao Junqi, Xie Ling, et al. Design and Implementation of Smart Community Platform for Epidemic Prevention and Environmental Protection [J] Electronic Technology and Software Engineering, 2021, (11) : 84-87.
 - [7] Liu Ping, Fan Rui. The Design and Implementation of Smart Community Service System [J] Electronic Components and Information Technology, 2021,5 (06) : 216-218.
 - [8] Lu Yuchen Discussion on the SpringBoot Framework in web Application Development [J] Science and Technology Innovation Herald, 2018,15 (08) : 168+173.
 - [9] Yang Jinshan, Wang Hui. The Design and Implementation of Project Information Management System Based on SpringBoot [J] Electronic Technology and Software Engineering, 2020, (09) : 38-39.
 - [10] Xia L ,Zhuona Z ,Keyang L , et al.Strengthening Community Defenses to Prevent and Control the Spread of COVID-19 in China.[J].China CDC weekly,2022,4(10):191-194.
 - [11] Xiong Yongping Analysis and Research on Application Development Technology Based on SpringBoot Framework [J] Computer Knowledge and Technology, 2019,15 (36) : 76-77.
 - [12] Xiong Yong Analysis of Background Service Program Design Based on Java Technology [J] Information Record Materials, 2022,23 (02) : 80-82.
 - [13] Dou Li. Query Optimization Technology Based on MySQL [J] Computer Knowledge and Technology, 2021,17 (15) : 35-36.
 - [14] Yan Zhang,Xinjie Jiang.Intelligent community system based on Internet of things design research analysis[J].The Frontiers of Society,Science and Technology,2019,1(3):6.
 - [15] Meng Hou.Analysis and Design of Community Management System based on Web[J].Frontiers in Economics and Management, 2021, 2 (2) : 149-158.
 - [16] Jian W ,Huiyan L ,Yafei W , et al.Research on the Construction of Local Grass-roots Community Emergency Management Systems Under the Normalization of Epidemic Prevention and Control[C]//[出版者不详],2022:
 - [17] Xie Q ,Hara R ,Kita H , et al.A real-time cooperative overvoltage prevention control by a community energy management system[J].Journal of International Council on Electrical Engineering,2016,6(1):224-230.
 - [18] Cardiovascular Diseases; Study Results from Centers for Disease Control and Prevention Broaden Understanding of Cardiovascular Diseases (Cost and economic benefit of clinical decision support systems for cardiovascular disease prevention: a community guide systematic ...) [J].Information Technology Newsweekly,2017,357-.
 - [19] Xiaoyan Z ,Shiyu Y ."A community system": A critical foundation for the epidemic prevention and control of SARS-CoV-2.[J].The International journal of health planning and management,2020,35(5):1246-1249.
 - [20] Hao J ,Wang P ,Jiao D , et al.Design and implementation of java-based community epidemic management system[J].The Frontiers of Society, Science and Technology,2022,4.0(5.0).