
Design and Implementation of Medical Service System for Sub-Healthy Population

Yingqing Zeng*

Guangzhou Huashang College, Guangzhou 511320, Guangdong, China

**Author to whom correspondence should be addressed.*

Copyright: © 2026 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: Traditional medical service systems often focus on disease treatment, while prevention and intervention for sub-healthy conditions are relatively weak. The back-end of the medical service system for sub-healthy individuals is built using a distributed microservice architecture based on the Spring Boot framework, while the front-end employs the Vue framework to design a responsive, interactive interface, paired with MySQL as the database management system. The aim is to create a comprehensive and easily scalable medical service system for sub-healthy individuals, providing them with more convenient and efficient health management services.

Keywords: Sub-healthy population; Health services; Medical management

Online publication: March 16, 2026

1. Introduction

According to the latest report from the World Health Organization, only 5% of the global population is in a fully healthy state, and the situation in China is even more severe, with the proportion of sub-healthy individuals reaching up to 75%^[1]. Based on the urgent demand for sub-health medical services in the current market, this system aims to lower the threshold of sub-health medical services, promoting the rational utilization of medical resources, and improving the user's health management experience as the research objects^[2]. It proposes an optimization scheme for an intelligent sub-health medical service system based on Spring Boot. This paper employs literature research and functional analysis methods to conduct a comprehensive and in-depth investigation and analysis of the needs of different user groups (including sub-health individuals, medical staff, and medical management personnel) in current sub-health medical services, ensuring the pertinence and comprehensiveness of the research. By clarifying the research tasks and overall plan, the system design is highly aligned with the actual needs of users^[3]. This system aims to establish a high-concurrency, efficient, and user-friendly sub-health medical service platform. The platform can provide users with more convenient and accurate health management services, including personalized health assessments, customized intervention plan recommendations, real-time health data monitoring, and convenient doctor-patient communication channels, helping users better manage their own health and improve their quality of life.

2. System analysis

2.1. System architecture analysis

The medical service system for sub-healthy individuals adopts a front-end and back-end separation architecture, primarily consisting of three major components: the user module, the doctor module, and the back-end administrator module. Each module encompasses multiple functional sub-modules. The structure of the user module encompasses functions such as user registration and login, consulting outpatient doctors, psychological testing, and browsing information. New users must complete registration before using the system for the first time, while registered users can log in directly using their username and password. After logging in, users can access the main interface to operate various functional modules. System operations are not available in the unlogged-in state. The doctor-side module includes functions such as personal center, registration record management, psychological test management, and medical record information management. In the backend administrator system, administrators need to log in successfully to the backend management system before they can access the backend homepage. If an account is not required, one can log out from the backend. Additionally, management of department information, patients/doctors, registration records, psychological tests, medical record information, etc., can be conducted in the backend.

2.2. Key technologies

2.2.1. Spring Boot

Spring Boot, as an open-source framework, plays a pivotal role in the rapid development of applications based on the Spring framework. Its core objective is to significantly simplify the development process, leveraging the powerful features of default configurations and automatic configuration, thereby significantly reducing the time and effort developers invest in configuration tasks^[4]. At the same time, Spring Boot possesses excellent integration capabilities, integrating a large number of third-party libraries and frameworks, making it easier for developers to integrate other technology stacks and achieve collaborative work between different technologies^[5].

2.2.2. Vue

Vue.js, as a popular JavaScript front-end framework, demonstrates exceptional performance in building interactive user interfaces and single-page applications. Its design follows the concept of simplicity, flexibility, and ease of use, providing developers with a user-friendly entry experience^[6].

3. System design

This system is developed in two parts: the web front end and the back-end management system, utilizing a B/S architecture. The web front end is designed for users and doctors, offering functionalities such as user registration and login, personal center, appointment registration, registration information management, and sub-health self-assessment. The back-end system includes modules for patient management, department management, appointment registration management, registration information management and a patient medical record management module. This system is developed using the Java language, following a three-tier architecture consisting of the controller layer, business logic layer, and data persistence layer. The controller layer is responsible for front-end and back-end data interaction, receiving requests from the browser and forwarding them to the business logic layer. In addition to providing services to the controller layer and executing specific business processes, the business logic layer can also interact with the data persistence layer^[7]. The data persistence layer, implemented by MyBatis-Plus, is primarily responsible for implementing synchronous operations between data and the database.

3.1. Overall system architecture design

The design and development of the medical service system for sub-healthy individuals is based on the Spring Boot

framework. The core lies in meticulously planning each functional module to ensure that each one aligns with user needs. After the system is fully developed, comprehensive functional testing is immediately implemented, aiming to reveal and fix potential issues, thereby ensuring the system can operate smoothly and reliably. The flowchart for adding login information in the system is shown in **Figures 1–2**.

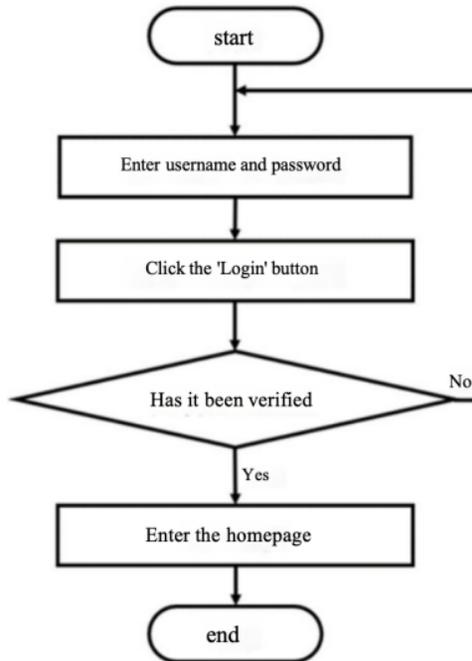


Figure 1. Login Flowchart.

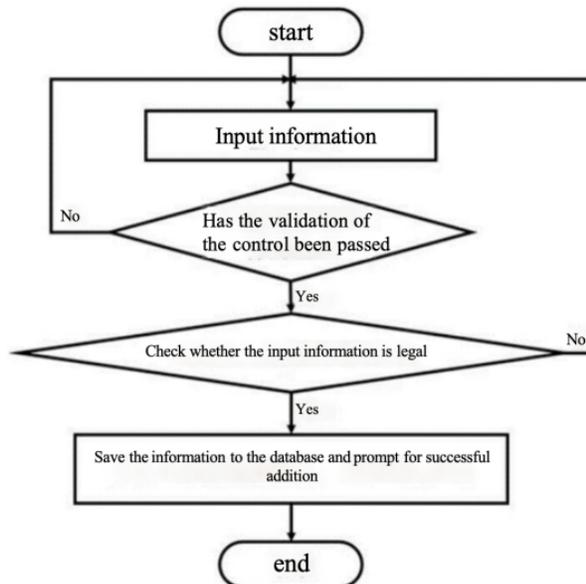


Figure 2. Add Information Flowchart.

3.2. Database design

3.2.1. Physical structure design

Based on business functional requirements, the system has been designed with user tables, token tables, department information tables, outpatient doctor tables, etc. The sample data table is shown in **Table 1**.

Table 1. User table

Field name	Type	Field description	Primary key	Default value
Id	Bigint	Primary key	Primary key	
Username	Varchar	Username		
Password	Varchar	password		
Image	Varchar	Avatar		
Role	Varchar	Role		Administrator
Addtime	Timestamp	New time		CURRENT_TIMESTAMP

4. System implementation and testing

4.1. Design and implementation of functions

4.1.1. Patient module implementation

(1) System homepage implementation

After opening the website, the patient first sees the homepage interface. On this page, the user can see the navigation bar and carousel of the medical service system for sub-healthy people based on Spring Boot, as shown in **Figure 3**.

(2) Cancel registration implementation

Fill in the name and “Cancel registration” in the search bar on the cancellation registration page to perform a search. Once you enter the cancellation registration page, you can view its detailed information as shown in **Figure 4**.



Figure 3. System homepage interface.

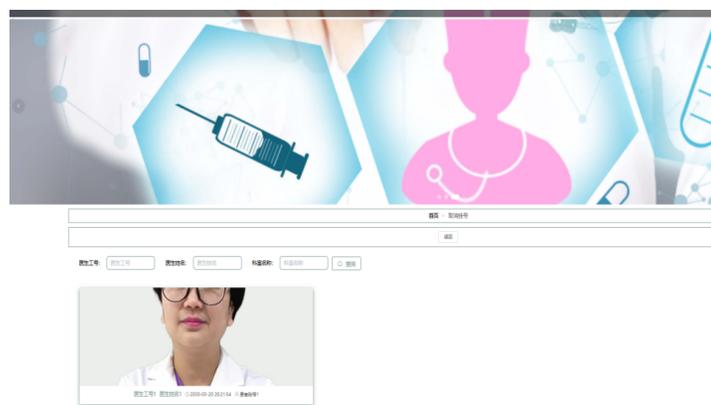


Figure 4. Cancel registration page.

(3) Realization of personal center

Patients can update their personal details, change their passwords, and perform detailed operations on the favorites page from their personal center page. The page of the personal center is as shown in **Figure 5**.

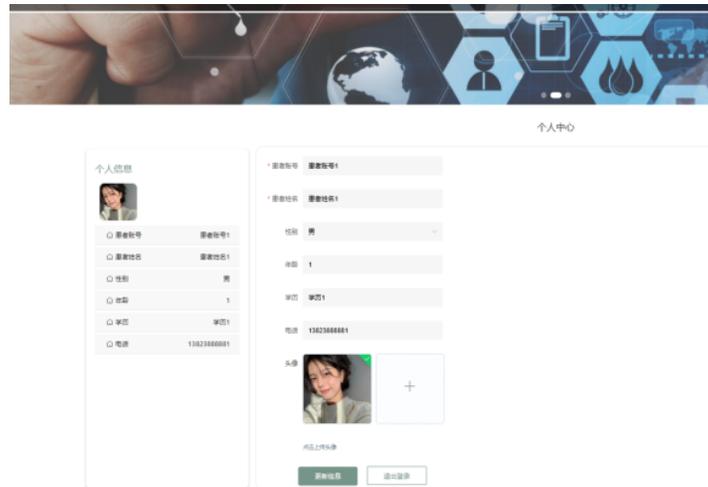


Figure 5. Personal center page.

4.1.2. Administrator module implementation

(1) Login function implementation

Before entering the backend login page, the administrator needs to first select the login role and accurately enter the username and password information. Once the verification is successful, the administrator can log in to the system without any barriers and perform the required operations. The backend login interface is shown in **Figure 6**.



Figure 6. Back-end login page.

(2) Implementation of the main administrator interface

After logging into the system, the administrator can access the fully functional main page. This page encompasses core modules such as the system homepage, personal center, outpatient doctor and patient management, cancellation registration management, message board feedback, and system management, allowing the administrator to perform relevant operations within these areas (**Figure 7**).

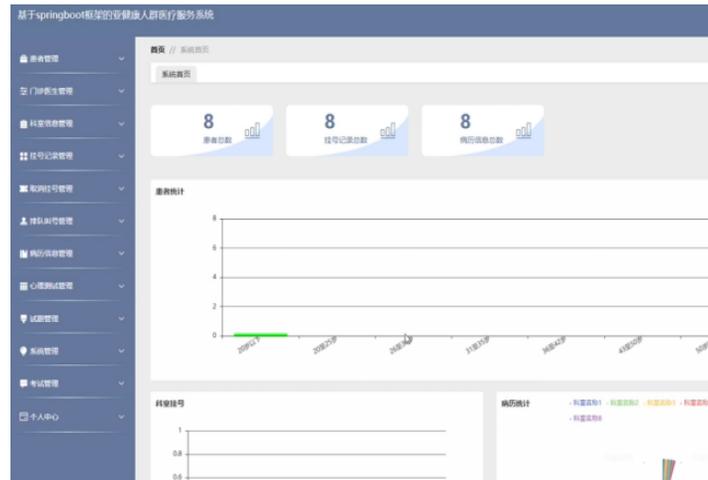


Figure 7. Main interface of the administrator.

(3) Implementation of doctor management

The administrator clicks on “Doctor Management” to enter the outpatient doctor page. By entering the outpatient doctor’s account number, name, and other relevant personal information, the administrator can perform queries, select whether to add or delete outpatient doctor information, view details, and perform modifications and deletions.

(4) Cancel registration management implementation

When the administrator accesses the cancellation registration management page, they can view the existing cancellation registrations and choose whether to delete them based on their needs. They can also click “Modify” to change their details. If they want to add a new cancellation registration, they simply need to click the “Add” button in the top right corner to start adding a new cancellation registration.

4.1.3. Implementation of outpatient doctor module

(1) Implementation of the main interface for doctors

Doctors can perform various tasks through the main page, including multiple functions such as the management system homepage, personal center, and cancellation of registration management.

(2) Cancel registration management implementation

After clicking on “Cancel Registration Management”, the outpatient doctor enters the cancel registration management page, where they can search and query based on department name and cancellation registration. When the desired information is found, the outpatient doctor can perform operations such as viewing, modifying, and deleting (Figure 8).

序号	医生工号	医生姓名	科室名称	联系电话	注册日期	头像	点击次数	收藏
1	医生工号 8 8 8	医生姓名 张德胜	科室名称 主任内科	1382188 8888	8 8	张德胜	8	8
2	医生工号 7 7 7	医生姓名 李德胜	科室名称 主任内科	1382188 8887	7 7	李德胜	7	7
3	医生工号 6 6 6	医生姓名 王德胜	科室名称 主任内科	1382188 8886	6 6	王德胜	6	6
4	医生工号 5 5 5	医生姓名 赵德胜	科室名称 主任内科	1382188 8885	5 5	赵德胜	5	5
5	医生工号 4 4 4	医生姓名 孙德胜	科室名称 主任内科	1382188 8884	4 4	孙德胜	4	4

Figure 8. Cancel registration management page.

5. Conclusion

The medical service system for sub-healthy individuals holds broad application prospects in today's society, offering individuals a convenient and efficient health management tool. The backend of this system employs the high-performance Spring Boot framework. Leveraging powerful development tools and excellent optimization capabilities, this framework fully meets the system's requirements for stability and scalability. When handling large volumes of medical data and high-concurrency user requests, Spring Boot ensures stable system operation and allows for easy expansion of functionality in accordance with business development. The front-end is built using the Vue.js framework, which boasts strong flexibility. Its responsive design and component-based development model provide users with a smooth operating experience, while also facilitating system expansion and maintenance. Through concise and intuitive interface design and friendly interaction processes, users can easily access various functions of the system, enhancing user satisfaction. In terms of data storage, it is combined with the relational database MySQL. MySQL is used to store structured medical data, such as basic user information, medical records, examination reports, etc., ensuring data integrity and consistency. By integrating these advanced technologies and tools, the aim is to create a high-concurrency, efficient, and user-friendly sub-health medical service platform. This platform can provide users with more convenient and accurate health management services, including personalized health assessments, customized intervention plan recommendations, real-time health data monitoring, and convenient doctor-patient communication channels, helping users better manage their own health and improve their quality of life.

Funding

2023 youth academic research project of Guangzhou Huashang College, "Research on Family Private Medical Smart Health Management" (Project No.: 2023HSQX047)

Disclosure statement

The author declares no conflict of interest.

References

- [1] Liu S, 2021, Explore Java Language and Android Mobile Software Development. *International Journal of Frontiers in Engineering Technology*, 3(2): 5–9.
- [2] Liang C, 2023, School Vehicle Management System Based on JAVA Language. *Academic Journal of Computing & Information Science*, 6(9): 126–131.
- [3] Scott VG, LJC, 2021, *Computer Graphics Programming in OpenGL with Java*. Mercury Learning and Information, 2021.
- [4] Abandansari TEM, Mehrsoroush S, Sirizi JM, et al., 2025, Survey of the Situation of the Prehospital Emergency Medical Services System in Iran. *BMC Emergency Medicine*, 25(1): 219.
- [5] Păcuraru MI, Năstac A, Zamfir A, et al., 2025, Digital Transformation of Medical Services in Romania: Does the Healthcare System Meet the Current Needs of Patients?. *Healthcare*, 13(20): 2549–2549.
- [6] Sun Y, Zou Y, Geng A, et al., 2025, Design and Implementation of Sports Social Information Management System Based on Spring Boot. *Journal of Electronic Research and Application*, 9(3): 52–57.
- [7] Guo T, Xue Z, 2024, Design and Implementation of Interactive Platform for Sharing Travel Guide Based on Spring Boot. *Academic Journal of Computing & Information Science*, 7(7): 1–8.

Publisher's note

Whoice Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.