

Epidemic Site Code Management System

Document



Candidate Name: Zhao Shirui

Candidate SCN: 227006951

Candidate Group: 20SD

Submission Date: 25, 4, 2023

Content

1. Project Proposal	1
2. Analysis and Design Stage	4
2.1 Basic analysis of the project	4
2.1.1 Questionnaire investigation	4
2.1.2 Similar system observation	11
2.1.3 Action Minutes	13
2.2 Feasibility study of a project	14
2.2.1 Technical aspects	14
2.2.2 Temporal aspects	15
2.3 Demand determination	15
2.3.1 Functional structure diagram	15
2.3.2 Functional requirements	16
2.3.3 Non-Functional requirements	16
2.3.4 UI requirements	16
2.4. Detailed project plan	20
2.4.1 Schedule	20
2.4.2 Gantt Chart	21
2.5 Systems analysis	21
2.5.1 Selection of development software, language and tools	21
2.5.2 Analysis of objects	22
2.5.3 Analysis of class	22
2.5.4 Input, output, storage and analysis of data	22
2.5.5 Basic process analysis	23
2.5.6 UML design	23
2.6 Systems design	28
2.6.1 UI Design	28
2.6.2 data dictionary	34
2.6.3 E-R Model	36
2.6.4 Functional design	36
3. Program Code	39
3.1 Overview	39
3.2 Index App	39
3.2.1 The Index Page for The User	39
3.2.1.1 The static page code is as follows:	40
3.2.1.2 The url path is as follows	44
3.2.1.3 The view function is as follows	44
3.2.2 The Index Page for The Admin	45
3.2.2.1 The static page code is as follows:	45
3.2.2.2 The url path is as follows	50
3.2.2.3 The view function is as follows	50
3.3 User App	50
3.3.1 The Register Page	50
3.3.1.1 Front End Display Interface Code	51
3.3.1.2 The url path is as follows	53

3.3.1.3 The view function is as follows	53
3.3.2 The Login Page	54
3.3.2.1 Front End Display Interface Code	54
3.3.2.2 The url path is as follows.....	56
3.3.2.3 The view function is as follows	56
3.3.3 The Info Page	59
3.3.3.1 Front End Display Interface Code	59
3.3.3.2 The url path is as follows.....	63
3.3.3.3 The view function is as follows	63
3.3.4 The Site Register Page	63
3.3.4.1 Front End Display Interface Code	63
3.3.4.2 The url path is as follows.....	65
3.3.4.3 The view function is as follows	65
3.3.5 The My Site Page	66
3.3.5.1 Front End Display Interface Code	67
3.3.5.2 The url path is as follows.....	69
3.3.5.3 The view function is as follows	69
3.3.6 The User-Site Page	69
3.3.6.1 Front End Display Interface Code	70
3.3.6.2 The url path is as follows.....	72
3.3.6.3 The view function is as follows	72
3.3.7 The Logout Function	73
3.3.7.1 Front End Style	73
3.3.7.2 The url path is as follows.....	73
3.3.7.3 The view function is as follows	74
3.3.8 Verification Code Function	74
3.3.8.1 Front End Style	74
3.3.8.2 The url path is as follows.....	74
3.3.8.3 The view function is as follows	74
3.4 QR code App	76
3.4.1 The Scan Page	76
3.4.1.1 Front End Display Interface Code	77
3.4.1.2 The url path is as follows.....	79
3.4.1.3 The view function is as follows	79
3.4.2 The Scan Function	80
3.4.2.1 Front End Display Interface Code	81
3.4.2.2 The url path is as follows.....	83
3.4.2.3 The view function is as follows	84
3.4.3 QR Code Generation Function	84
3.4.3.1 Front End Display Interface Code	85
3.4.3.2 The url path is as follows.....	85
3.4.3.3 The view function is as follows	85
3.5 Supervise App	86
3.5.1 The User-Status Page	86
3.5.1.1 Front End Display Interface Code	86
3.5.1.2 The url path is as follows.....	89

3.5.1.3 The view function is as follows	89
3.5.2 The User-Status Modify Page	90
3.5.2.1 Front End Display Interface Code	90
3.5.2.2 The url path is as follows.....	93
3.5.2.3 The view function is as follows	93
3.5.3 The Site-Status Page	93
3.5.3.1 Front End Display Interface Code	94
3.5.3.2 The url path is as follows.....	96
3.5.3.3 The view function is as follows	96
3.5.4 The Site-Status Modify Page	97
3.5.4.1 Front End Display Interface Code	97
3.5.4.2 The url path is as follows.....	100
3.5.4.3 The view function is as follows	101
5.5 The User Visit Page	101
3.5.5.1 Front End Display Interface Code	101
3.5.5.2 The url path is as follows.....	104
3.5.5.3 The view function is as follows	104
3.5.6 The Application Verification Page	105
3.5.6.1 Front End Display Interface Code	105
3.5.6.2 The url path is as follows.....	107
3.5.6.3 The view function is as follows	107
3.6. Plug-in Components	107
3.6.1 Pagination	107
3.6.1.1 Pagination code	108
3.6.2 Middleware	110
3.6.2.1 Middleware code	110
3.6.3 Verification Code Function	111
3.6.3.1 Verification Code	111
4. Program Test	113
4.1 App Test	113
4.1.1 Index App	113
4.1.1.1 User Main Interface	113
4.1.1.2 Admin Main Interface	114
4.1.2 User App	115
4.1.2.1 Register Interface	115
4.1.2.2 Register	115
4.1.2.3 Login Interface	118
4.1.2.4 Login	119
4.1.2.5 View User Information	120
4.1.2.6 Site code application	121
4.1.2.7 My Site Interface	122
4.1.3 QR Code App	123
4.1.3.1 Scan Interface	123
4.1.3.2 Scan QR code Interface	123
4.1.3.3 Generate QR code Interface	125
4.1.4 Supervise App	126

4.1.4.1 Login Admin	126
4.1.4.2 Verify Site	128
4.1.4.3 User Status modify function	128
4.1.4.4 User Status function	129
4.1.4.5 Site Status function	131
4.1.4.6 Site Status modify function	132
4.1.4.7 User-Site Record Function	133
4.2 Additional Tests	135
4.3. Debug	137
4.3.1 Scan Interface	137
4.3.2 User Status function	138
4.3.3 User Status Modify function	139
4.3.4 User Register function	140
4.3.5 Optimize the scanning interface	142
4.4 Conclusion	143
4.4.1 Purpose of testing:	143
4.4.2 Test scope:	143
4.4.3 Testing methods:	144
4.4.4 Test results:	144
4.4.5Test conclusion:	144
5. Evaluating Stage	144
5.1 Assignment Outline And Degree of Development	144
5.1.1 Ordinary users	144
5.1.2 Administrator	144
5.1.3 The Degree of Project Completion	144
5.2 Project Advantages and Disadvantages Assessment	145
5.2.1 Project Advantages	145
5.2.2 Project Disadvantages	145
5.3 Future development	146
5.3.1 Solutions	146
5.3.2 Reasons	146
5.4 Modifications	146
5.4.1 Modification content	146
5.4.2 Impact of Modification	146
5.5 Identification of Any Knowledge And Skills	147
5.5.1 Personal gains	147
5.5.2 Improvement project	147
6. Documentation	149
6.1 Normal users	149
6.2 Administrator	149
7. References	150
8. History Version	151
9. Appendix	152

1. Project Proposal

Title	Epidemic prevention and control site code management system
Project leader	Zhao Shirui
Aim	This system is mainly for merchants and customers. The merchant can complete the application for place code, and the customer can complete the registration of the system and scan the place code applied by the merchant. The customer's information will be entered into the database for retention after the customer has scanned the code.
Objectives	<p>Description of the role of the service:</p> <p>Customers, Business Operators, System First Level Operators, System Second Level Operators</p> <p>Functions that can be realized by each role:</p> <p>Customers:1. Be able to register user information. 2. Be able to scan the site code.</p> <p>Business Operators:1. Be able to register merchant information. 2. Be able to apply for the location code of the merchant. 3. Be able to independently apply for destroying the site code which has been applied for.</p> <p>System Second Level Operators:1. Be able to view the merchants who have applied for the pass code. 2. Be able to allow or reject the merchant's application. 3. Be able to view all customer information. 4. Be able to check the places where customers are.</p> <p>System First Level Operators:1. Be able to view the merchants who have applied for the pass code. 2. Be able to allow or reject the merchant's application. 3. Be able to view all customer information. 4. Be able to check the places where customers are. 5. Be able to clear all data.</p>
Content	<p>1. Project background introduction.</p> <p>Company name: Dawn Tec.</p> <p>Company location: Room 301, Block A, No. 3 Real Estate, No. 1A, Shuguang Xili, Sanyuanqiao, Beisanhuan East Road, Chaoyang District, Beijing (100028).</p> <p>Company size: a microenterprise with seven employees, including company boss, company manager, project development and testing people, and graphic designers.</p>

	<p>Types of project customers: management, ordinary users, merchants, etc.</p> <p>The project starts and ends: December 16, 2022 to April 9, 2023.</p> <p>Significance of the project development: Due to the impact of the epidemic situation, there are serious problems in the normal operation of merchants and the travel of customers. The development of this project is mainly to know the travel route of a user in case of abnormal nucleic acid detection, so as to prevent the spread of the epidemic in a large way. The development of this project is conducive to the business of normal merchants and the travel of normal customers.</p>
	<p>2. Project development environment.</p> <p>2.1 Hardware:</p> <ul style="list-style-type: none"> 1) Personal computer: used to store and develop projects. 2) Personal tablet: used to design UI interface. 3) ECS: used for deployment and testing projects. <p>2.2 Software:</p> <p>2.2.1 Develop Tools.</p> <ul style="list-style-type: none"> 1) PyCharm, Version: 2022.1.3(Professional Edition) 2) Apache, Version: 2.4.54 3) MySQL, Version:8.0.31 <p>Alternative 1</p> <ul style="list-style-type: none"> 1) PhpStorm, Version: 2022.1.3 2) Apache, Version: 2.4.54 3) MySQL, Version:8.0.31 <p>Alternative 2</p> <ul style="list-style-type: none"> 1) IntelliJ IDEA, Version: 2020.1 (Ultimate Edition) 2) Apache Tomcat, Version: 9.0.69 3) MySQL, Version:8.0.31 <p>2.2.2 Develop Language.</p> <ul style="list-style-type: none"> 1) Python Version: 3.10.3 <p>Alternative 1</p> <p>PHP-Version 8.1.10).</p> <p>Alternative 2</p> <p>Java, Version 1.8</p> <p>JSP, Version 2.3</p> <p>Servlet, Version 4.0</p>
Methodology	<p>1. Skills Learned</p> <p>I have learned 1. Dynamic website development, 2. Project management, 3. c++,</p>

	<p>4. Python, 5. Research Skill and other courses. Through these courses, I can complete some basic functions and help the implementation of the project.</p> <p>I think I still need to go online for information on the generation of place codes applied by merchants and the interaction of place codes scanned by customers. Although I have learned Python, I only need to continue to study some basic methods of using Python and some powerful library operations. I plan to use the Django framework to build this project, so I need to learn about the use of Django online.</p>
	<p>2. Conduct Research.</p> <ol style="list-style-type: none"> 1. For ordinary citizens, the merchant conducts a questionnaire to check whether the system is needed and the general direction of use. 2. For management personnel, conduct in-depth interviews and exchanges, send invitations to invite them to visit, and deeply discuss the needs and function realization of this project.
	<p>3. Personal Direction</p> <ol style="list-style-type: none"> 1. I first need to go to bilibili to check the courses on Django framework construction, and guarantee 4 hours of effective learning time every day to complete Django learning. 2. I want to learn the QR code generation, scanning results and subsequent operations on Moke. 3. Try to learn the spring framework to build the project on the premise of ensuring the quality. 4. I have learned the courses related to object-oriented design, but I think that the object-oriented design I have learned is not very specific and can not be put into real use, so I bought the book System Analysis and Design Methods to help me complete the system analysis and design of this project.
Proposal Submitted to	Yang Shuchun
Date Submitted	2022/12/19
Proposal Result	Pass/Fail
Supervisor Signature:	

2. Analysis and Design Stage

2.1 Basic analysis of the project

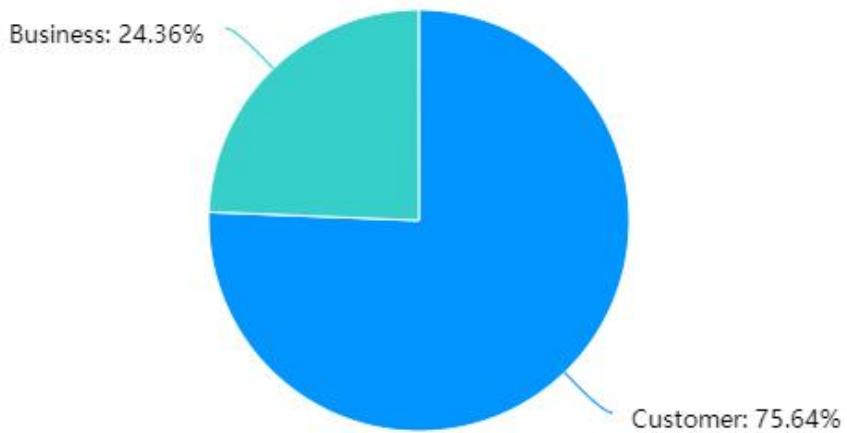
2.1.1 Questionnaire investigation

For the safety of the people, it is now necessary to develop an epidemic site code management system. Through this questionnaire, I hope to get some views and needs of the public (customers, site managers) on the system to help the development of the system. A total of 85 questionnaires were distributed through QR code scanning and sharing, of which 73 were valid.

This questionnaire is only for academic purposes, not for commercial purposes, and any information collected in it will not be disclosed.

Figure1:

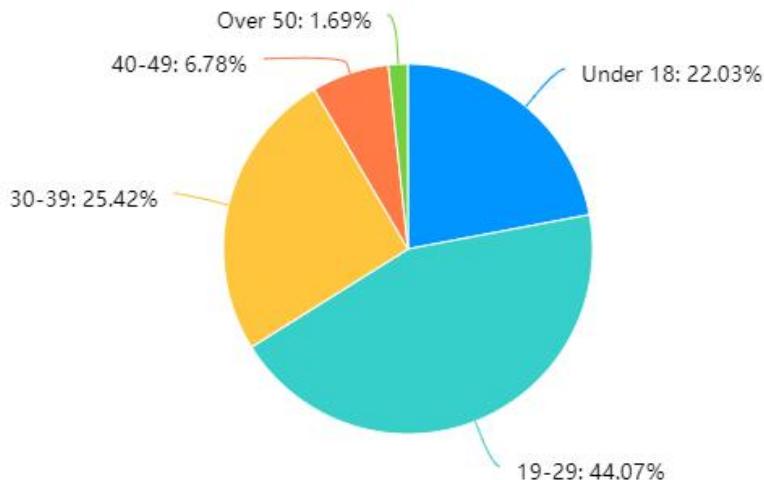
What is your identity?



It can be seen from the figure that about three quarters of the surveyed people are consumers (75.64%) and one quarter are merchants (24.36%).

Figure2:

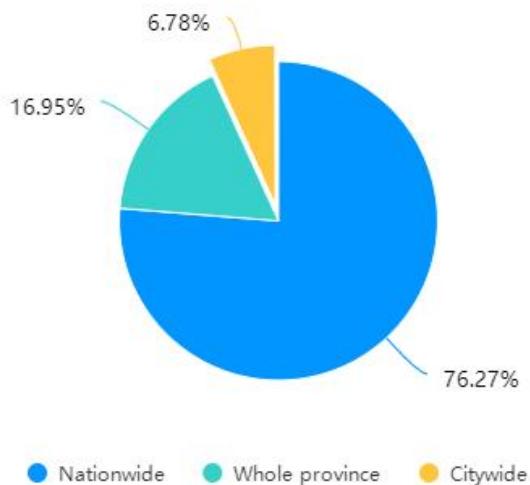
What is your age?



It can be seen from the figure that 22.03% of the surveyed consumers are minors, the population aged 19-29 accounts for about half (44.07%), 25.42% of the consumers are in the 30-39 age group, nearly a quarter, 6.78% of the consumers are in the 40-49 age group, and only a small part of the population is over 50 years old (1.69%).

Figure3:

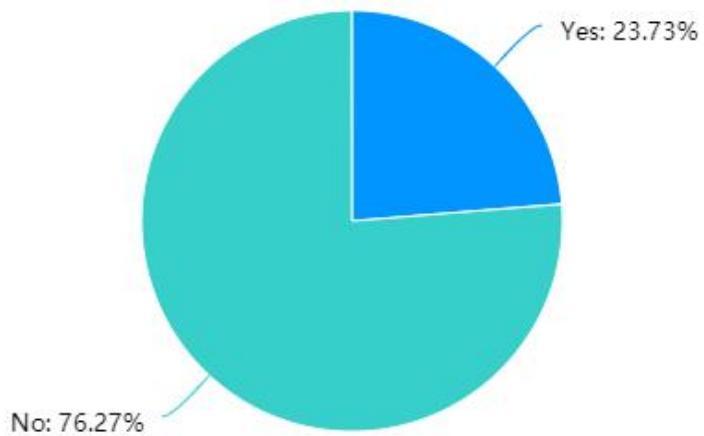
Where can the system be used?



It can be seen from the figure that nearly three quarters of the consumers surveyed believed that the system should be used nationwide (76.27%), 16.95% believed that it should be used throughout the province, and only 6.78% believed that it should be used throughout the city.

Figure4:

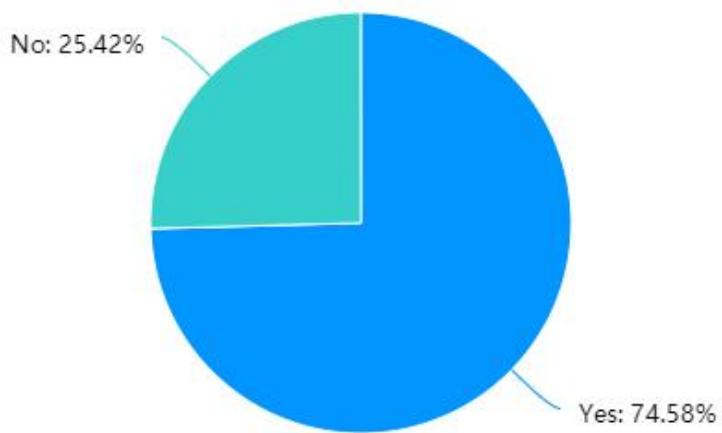
Can a user apply for site qualification for family members?



It can be seen from the figure that nearly three quarters of the surveyed consumers think that a user should not be allowed to apply for the qualification instead of his family (76.27%), while the rest think that this function is necessary (23.73%).

Figure5:

Do you think it is necessary to judge the status of the site through the system?



It can be seen from the figure that nearly three quarters of the surveyed consumers think that it is necessary to let the system display the status of this place (74.58%), while the rest think that this function is unnecessary (25.42%).

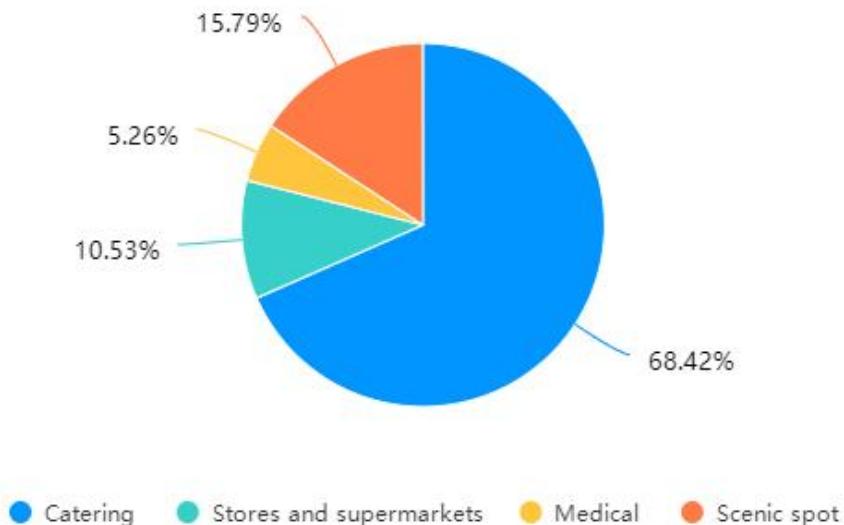
Figure6:

What do you think this system can help customers understand?



It can be seen from the figure that 79.66% of the surveyed consumers think that the system needs to display the current health status of customers, 72.88% think that the system needs to display whether the current place is safe, 37.29% think that the system can query whether it has been traveling with abnormal people synchronously, and 25.42% think that it needs to display whether customers have been vaccinated.

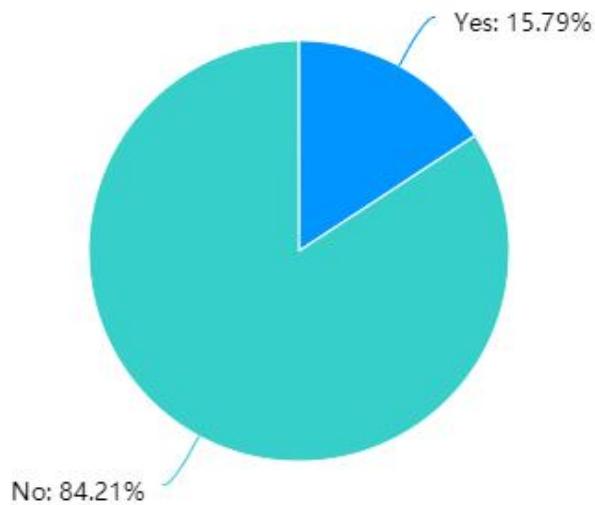
Figure7:
What kind of enterprise do you operate?



It can be seen from the figure that 68.42% of the places surveyed are catering, 10.53% are supermarkets, 5.26% are medical and 15.79% are tourist attractions.

Figure8:
Whether online help can be provided on the interface to solve the difficulties encountered by

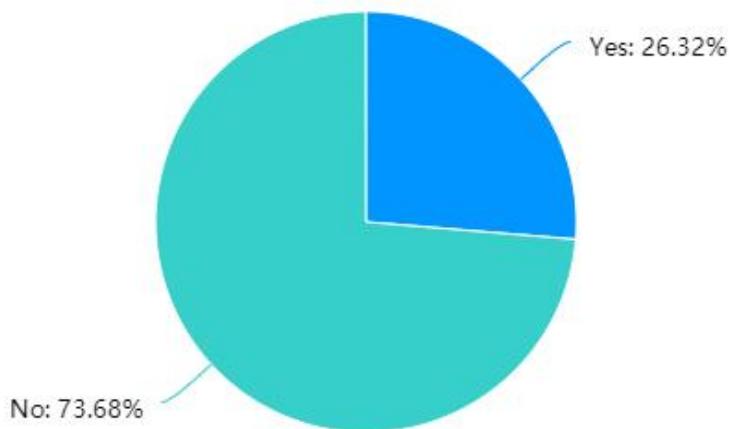
users during use?



It can be seen from the figure that 84.21% of the site administrators in the surveyed places think that there is no need to add an online help function to the system, while another part (15.79%) think that this function is needed.

Figure9:

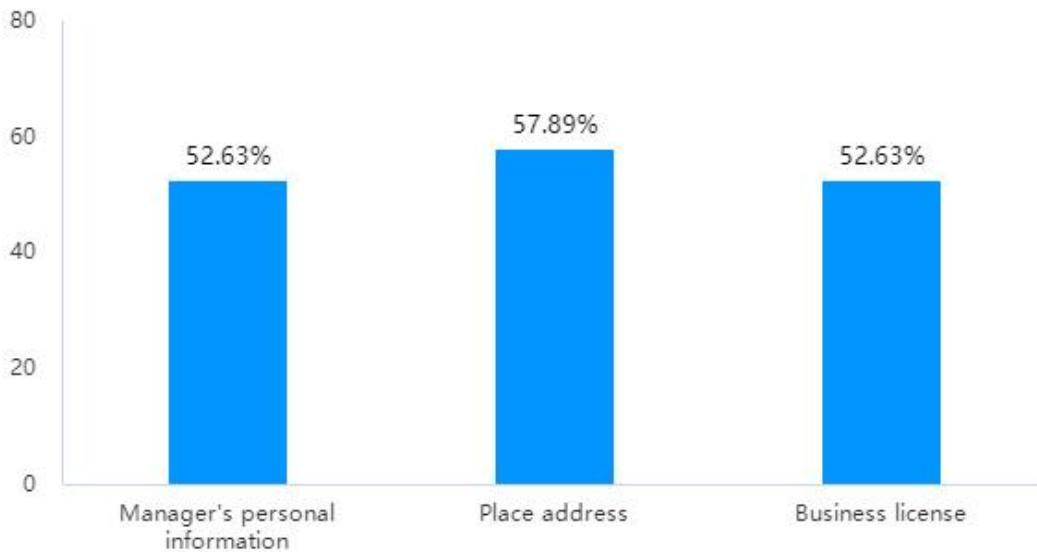
Do you think the site operator has the right to notify and warn customers ?



It can be seen from the figure that 73.68% of the investigated places believe that they have no right to inform customers of the people who have been in and out of the place with abnormal health. And 26.32% of the places think they have this right.

Figure10:

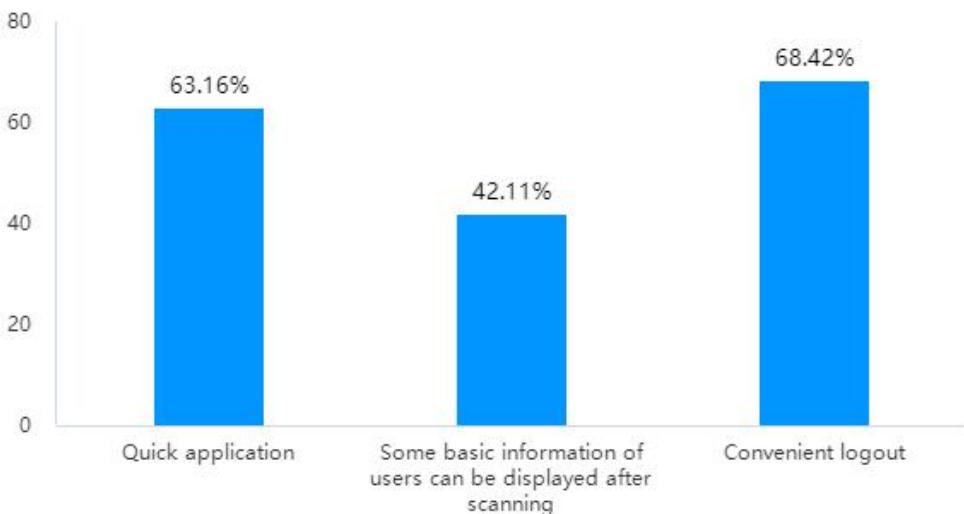
What conditions do you think are needed to apply for the site code?



It can be seen from the figure that 52.63% of the surveyed places think it is necessary to fill in the manager's personal information, 57.89% of the places think it is necessary to fill in the place address, and 52.63% of the places think it is necessary to provide the business license.

Figure11:

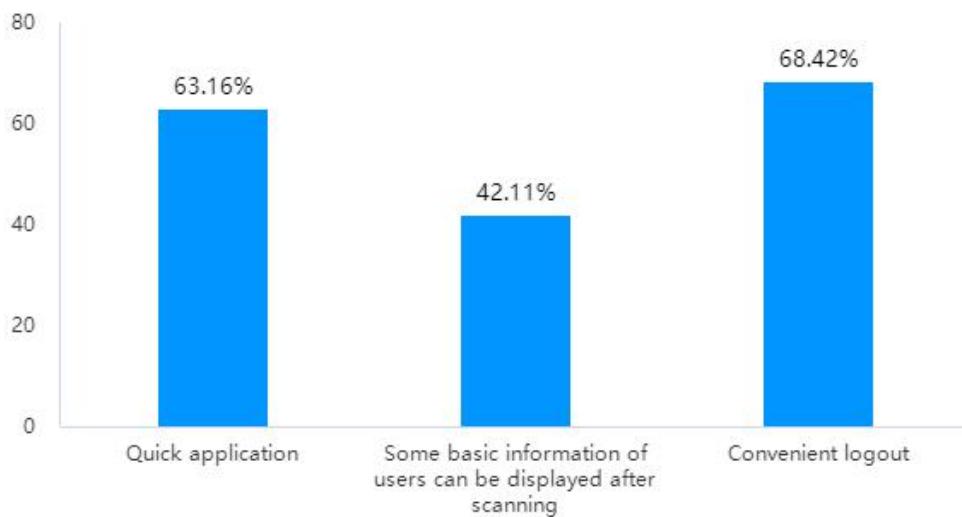
If this system is put into use, what functions do you think the site code needs?



It can be seen from the figure that 63.16% of the sites surveyed think that the application needs to be fast, 68.42% of the sites think that the application site code can be canceled at any time, and 42.11% of the sites think that part of the customer's personal information needs to be displayed after scanning.

Figure12:

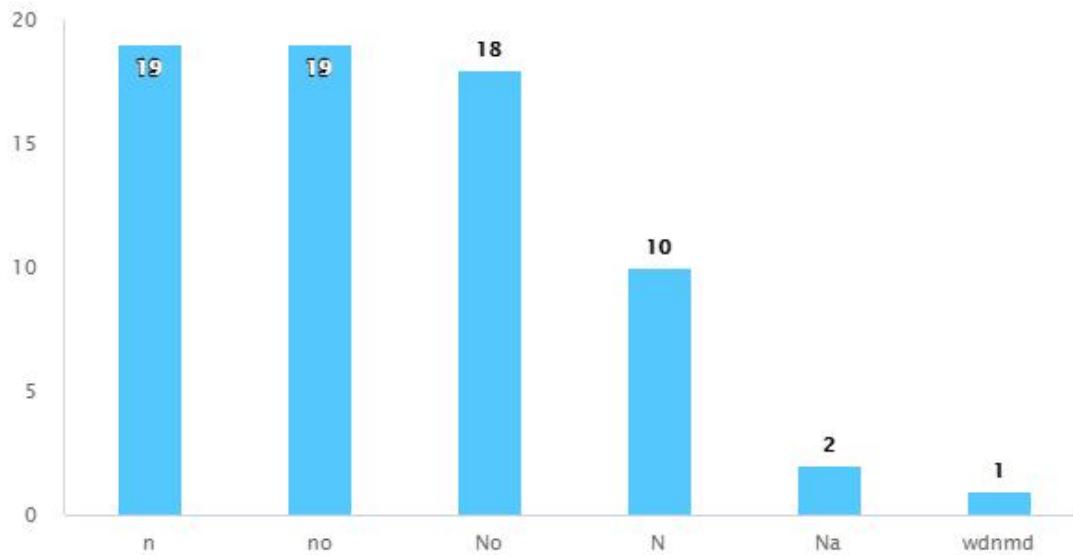
What do you think this system can help the site understand?



It can be seen from the figure that 57.89% of the surveyed places think that they need to know whether the customer is healthy, 63.16% of the places think that they need to warn customers who are not healthy, and 31.58% of the places think that they need to inform normal customers that they have contact with abnormal customers.

Figure13:

Do you think this system needs other functions? Please list.



It can be seen from the figure that the vast majority of the respondents did not provide the need to improve the system.

Conclusion

Through this survey, I can basically identify several basic needs:

For customers

1. Be able to display personal health status.
2. Know whether the scanning place is safe.
3. The scope of application is applicable nationwide.

For the place

1. The manager information, address and business license are required to apply for the site code.

2. The place code applied can be returned quickly and can be canceled in time.
3. This system can prompt all customers with abnormal health.

2.1.2 Similar system observation

This is the schematic diagram of Shandong Electronic Health Pass, which is used as a reference basis for reference.

This is the main interface of mobile terminal.



This is the place where individual users need to register for the first time. When users meet the required conditions, they can log in.

个人信息

姓名	
证件号码	
手机号码	
国籍(地区)	
当前居住地址	

This is the site code application interface, which includes various conditions for the site code application and requires various information of the applicant.

场所码申请

一码在手 全国通行

场所类型 社区（村）车库入口

场所名称 如xx餐厅、xx超市

行政区划 请选择行政区划

详细地址 请输入详细地址

联系人 请输入联系人

联系电话 请输入联系电话

本人对场所信息的真实性和完整性负责。如若伪造场所信息，本人愿承担相应法律责任。

注册

我的场所

This is the site code management interface. Once again, you can manage the site codes that have been applied for.



Conclusion

Through this observation, I understand the basic design direction of the system and how to design the basic interface of the system, as well as the conditions that users need to meet when registering and the matters needing attention when applying for the site code.

2.1.3 Action Minutes

Recorder: Zhao Shirui

Date of meeting	23/1/20	Time	3pm-4:30pm
Location	We-Chat	Minute taker	Zhao Shirui
Present	Zhao Shirui, An Moli, Dawn Jack, Zhao Xin, Li Sijia.		
Apologies	None		

Item	Conference Organizers	Timescale
1. The main functions of the administrator.	Zhao Shirui, An Moli, Dawn Jack, Zhao Xin, Li Sijia.	1.5 hours

Discussion/Action:

Zhao Shirui: So, do you think there is any need for additional information?

An Moli: I think the needs of the main users, such as customers and venue personnel, have been improved, as well as the management personnel. User registration and site code application have been mentioned. How can managers judge whether they can apply for site code?

Zhao Shirui: The administrator needs to check his/her business license, site address and manager information, and can issue the electronic health code after the review is completed.

An Moli: That would be great.

Zhao Shirui: Do others have any ideas?

Dawn Jack: I have any idea, that is, if the administrator finds that there have been abnormal cases in

this place, whether the administrator needs to adjust the state of this place?

Zhao Shirui: After verification, as long as there is an abnormality, it will be displayed as an abnormal place?

Li Sijia: The administrator just needs to check the application of the site code submitted? Don't you need to check the user's registration information? In case any user uses illegal address, words, etc., we also need to bear the responsibility of this part.

Zhao Shirui: If there are many registered users and the usage is large, the administrator needs to carry out repeated work management, which is a very heavy task. Checking users can verify and review by restricting the registration of users, which can save a lot of labor.

Li Sijia: When the machine can't recognize it, it can make improper input in a different way.

Zhao Xin: No, we just need to restrict users to input only numbers or letters where they input. It is necessary to save labor, otherwise it will be very expensive.

Li Sijia: Indeed, labor costs are also high!

Dawn Jack: How do administrators register? How to determine who is the administrator and who is the ordinary user?

Zhao Shirui: The administrator is directly determined by the background, and can't be used after registering like ordinary users, which is illegal. However, login can be set to login as usual, which is safer and can also simplify some development.

Dawn Jack: OK, I see.

Zhao Shirui: Do you have anything to add or ask?

Everyone shook their heads.

Zhao Shirui: Ok, that's all for our discussion.

Conclusion

Through this meeting, I clarified the task of the administrator. The administrator needs to review the site code applied for. If he has any questions about the information submitted, he can refuse to pass the application. If there are abnormal conditions in the place, such as the abnormal personnel entering and leaving the place, after verification, the place will be determined as an abnormal place, and the abnormal display will be made.

2.2 Feasibility study of a project

2.2.1 Technical aspects

I have learned 1. Dynamic website development, 2. Project management, 3. Python, 4. Research Skill

and other courses. Through these courses, I can complete some basic functions and help the implementation of the project.

I think I still need to go online for information on the generation of place codes applied by merchants and the interaction of place codes scanned by customers. Although I have learned Python, I only need to continue to study some basic methods of using Python and some powerful library operations. I plan to use the Django framework to build this project, so I need to learn about the use of Django online. In addition, I also need to understand the operation of QR codes, such as how to generate QR codes, how users scan QR codes, and how to express existing information in QR codes.

2.2.2 Temporal aspects

- 1) Planning stage: 2022.12.12-2022.12.16
- 2) Analysis stage: 2022.12.17-2022.12.31
- 3) Design stage: 2023.1.1-2023.1.28
- 4) Programming stage: 2023.1.29-2023.2.25
- 5) Test stage: 2023.2.26-2023.3.5
- 6) Evaluation stage: 2023.3.6-2023.4.10

In the programming stage, it can be divided into the following small stages

- 1. Front-end interface development stage: 2023.1.29-2023.2.4
- 2. Registration and login function development stage: 2023.2.5-2023.2.7
- 3. Application function development stage: 2023.2.8-2023.2.10
- 4. Code scanning function development stage: 2023.2.11-2023.20
- 5. Audit function development stage: 2023.2.21-2023.2.25

2.3 Demand determination

2.3.1 Functional structure diagram

Basic functions of epidemic prevention and control place code management system		
Epidemic prevention and control place code management system	Ordinary users	User registration
		User login
		View personal information
		Scan site code
		Apply for the site code
		Cancellation of site code
	Administrator user	Administrator login
		Approval site code
		Check site information
		Determine site status
	View the result of user scanning.	

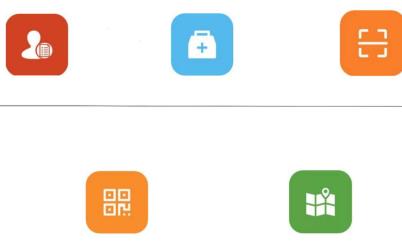
2.3.2 Functional requirements

1. Users can register and log in to this system.
2. Users can view personal information.
3. The user can view the status of the site code after scanning the site code.
4. Users can apply for site code.
5. The user can cancel the site code.
6. Administrators can log in to the system
7. The administrator can review the application for site code.
8. The administrator can select the status of the site code after verification.
9. The administrator can check the site information.
10. The administrator can view the result of user scanning.

2.3.3 Non-Functional requirements

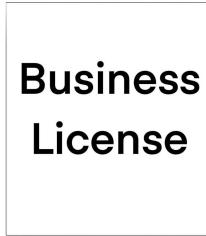
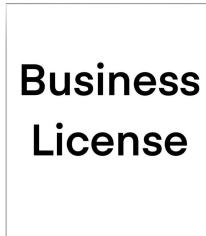
1. Development on the PC side.
2. Use python, mysql and Apache for supporting development.
3. Use the Django framework for development.
4. The development interface is beautiful and practical.
5. Immediate response.
6. Wide range of use (can be used nationwide).

2.3.4 UI requirements

UI Design	Comment
<p style="text-align: center;">Main Interface</p> 	This is the UI of the main interface, where from left to right are the functions of personal information registration/login, personal information query, scanning site code, site code application, site code view, etc.

<p>Personal Information</p> <p>Name <input type="text"/></p> <p>Phone <input type="text"/></p> <p>Password <input type="text"/></p> <p style="text-align: center;">Submit</p>	<p>This is the UI of the registration interface, where users can register accounts by submitting corresponding data, such as mobile phone number, name, password, etc.</p>		
<p>Login</p> <p>Phone <input type="text"/></p> <p>Password <input type="text"/></p> <p style="text-align: center;">Admin <input type="radio"/> User <input checked="" type="radio"/></p> <p style="text-align: center;">Submit</p>	<p>This is the UI of the login interface. After entering the correct account and password, users can log in according to the corresponding role.</p>		
<p>Personal Status</p>  <p>Name: _____</p> <p>Phone: _____</p> <hr/> <p>January 21, 10:21:20</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Status</td> <td style="width: 70%; color: green; font-weight: bold;">Health</td> </tr> </table>	Status	Health	<p>This is the display interface of general user's personal information, which displays the user's personal information and the user's current health status.</p>
Status	Health		

<p>Site type <input type="text"/></p> <p>Place name <input type="text"/></p> <p>Address <input type="text"/></p> <p>Contacts <input type="text"/></p> <p>Contact No. <input type="text"/></p> <p>Submit</p> <p>My application</p>	<p>This is the site code application submission interface. When the application information is completed, it can be submitted to the management end.</p>
<p>< My Application</p> <p>Background</p>  <p>No Data</p>	<p>This is the common user site code management interface, where users can view the site code that has been applied for, or cancel the site code.</p>
<h2>Site Code Management</h2> <hr/> <p>Business License</p>	<p>This is the administrator view application interface, where the administrator can roughly view all applications.</p>

<h2>Site Code Management</h2>  <p>Site type _____</p> <p>Place name _____</p> <p>Address _____</p> <p>Contacts _____</p> <p>Contact No. _____</p> <p>Pass Fail</p>	<p>This is the administrator view application details interface. The administrator can view various information of a single application and decide whether to pass or not.</p>
<h2>Site Code Management</h2>  <p>Status Health</p> <p>Place name _____</p> <p>Address _____</p> <p>Update</p>	<p>This is the administrator's view site status interface. If the administrator checks that the current site is abnormal, he can modify its status.</p>

Site Code Management

Name	Phone	Time	Site No

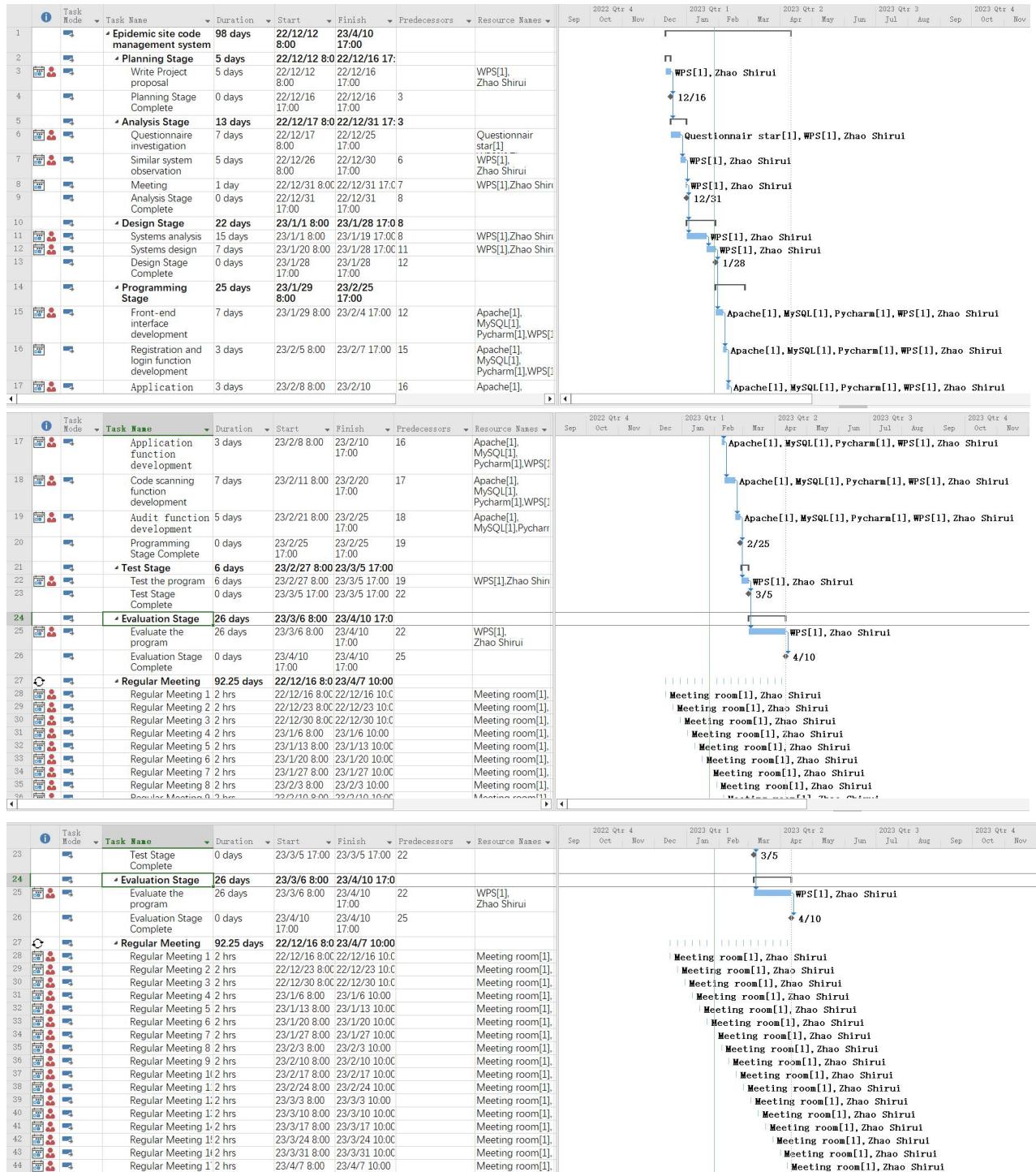
This is the interface for the administrator to view the user's visited location. After the user has scanned the code, part of the user's information will be recorded, and the user can see when and where he has visited.

2.4. Detailed project plan

2.4.1 Schedule

NO. Day Action	5	15	28	7	3	3	10	5	8	35
Planning the Project										
Analysis the Project										
Design the Project										
Front-end interface development										
Registration and login function development										
Application function development										
Code scanning function development										
Audit function development										
Test the Project										
Evaluation the Project										

2.4.2 Gantt Chart



2.5 Systems analysis

2.5.1 Selection of development software, language and tools

The development software uses pycharm (Version: 2022.1.3 (Professional Edition)), the database selects mysql (Version: 8.0.31), and the server selects Apache (Version: 2.4.54). Using python to develop web services can not only show the technology i have learned, but also show my

self-learning ability. Not only that, python also has rich third-party library packages that can help me develop complex programs.

2.5.2 Analysis of objects

1. User objects can perform simple registration, login, personal information query and other functions.
2. Ordinary users, inherited from users, can apply for and scan the site code.
3. The administrator user, inherited from the user, can review the application of the site code, modify the status of the site code, and view the user's access information.
4. The QR code generation object can generate a QR code from the location information.
5. QR code scanning object, capable of scanning QR code, reading and writing data.

2.5.3 Analysis of class

Class Name	Attribute	Method
User	-name: String -phone: String -password: String	+registration(): int +login(): int +query(): void
Ordinary_user	-address: String -pictures: Image -street: String -sex: String -status: String -site_status: String	+apply(): void +cancellation(): void +use_scan(): void
Admin_user		+approval(): int +modify(): int +view(): int
QR_generation	-address: String	+generation(): void
QR_scan		+scan(): void

2.5.4 Input, output, storage and analysis of data.

- 1.The name, mobile number, password, address and other information of ordinary users are stored in the relational database. When the user registers the account successfully, insert a new data into the database and prompt that the registration is successful. When users log in, they can log in after entering the correct account and password, which can match the data in the database.
- 2.The administrator judges the user's application. When the user's application is successful, the application address filled in by the user can be converted into a QR code.
- 3.The administrator modifies the status of the user's site. When it is confirmed to be modified, the

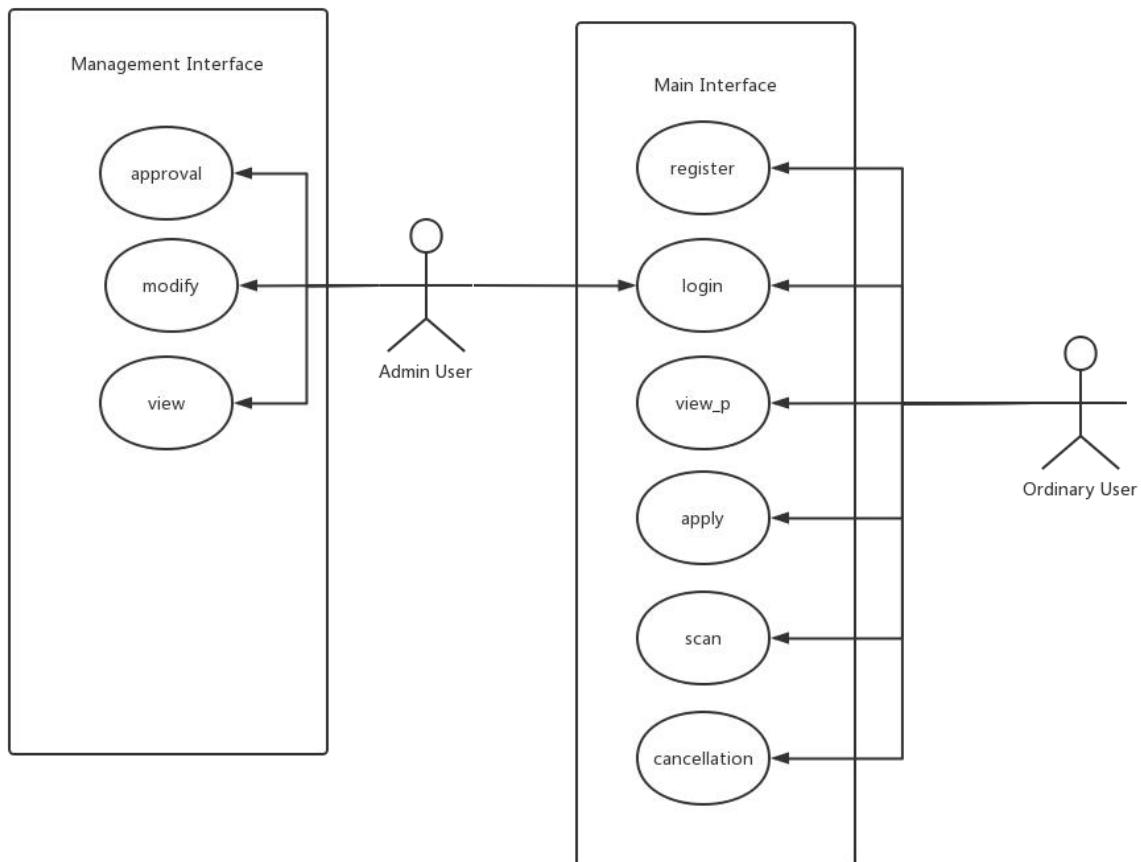
data will be updated in the database,

2.5.5 Basic process analysis

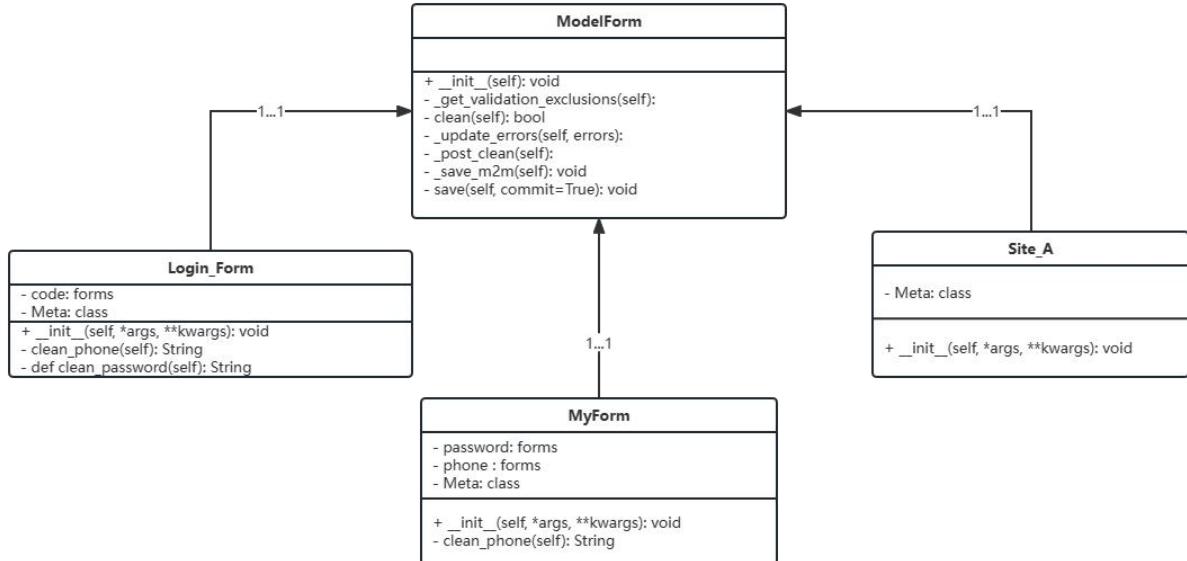
1. Ordinary users can register and log in to the system by submitting personal information, view personal information, and submit corresponding materials to apply for a place code for their own place.
2. The administrator user can approve the application of ordinary users for the site code, modify the current status of the site, and view the visits of all users.

2.5.6 UML design

1. Use Case diagram

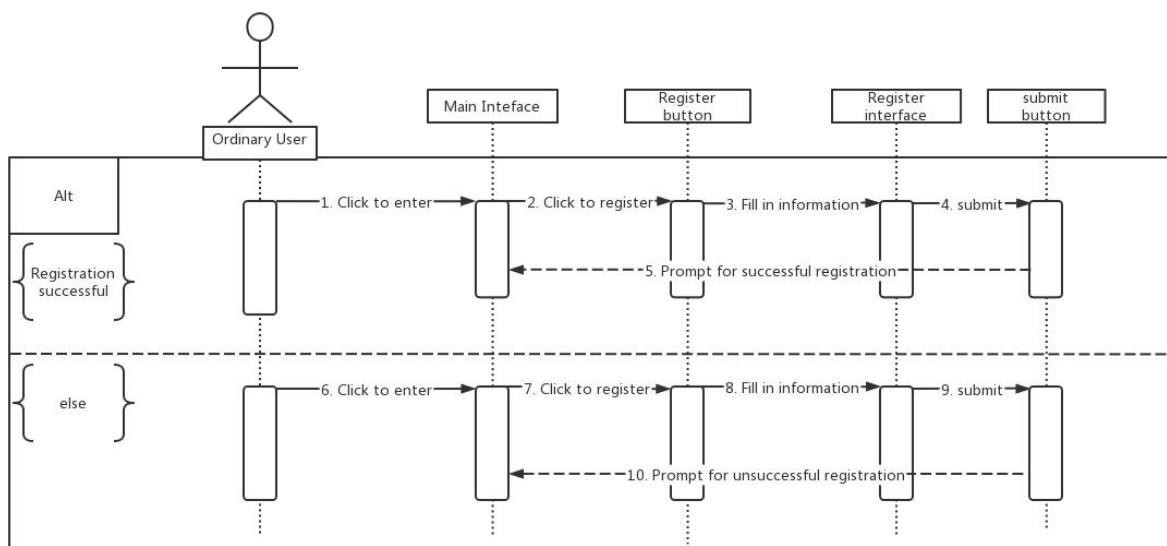


2. Class diagram

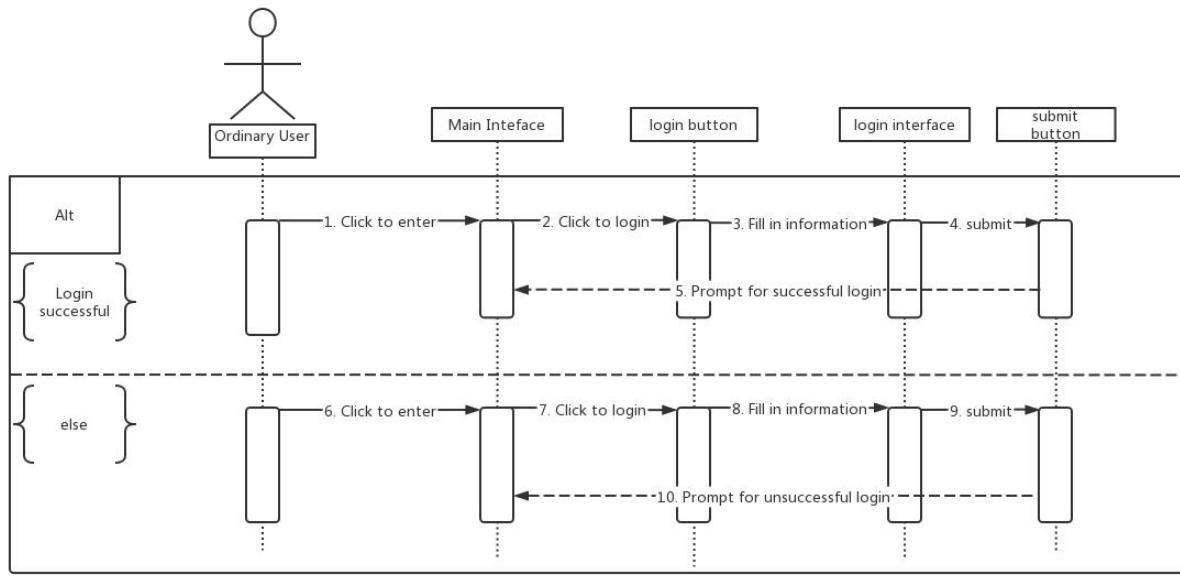


3. Sequence diagram

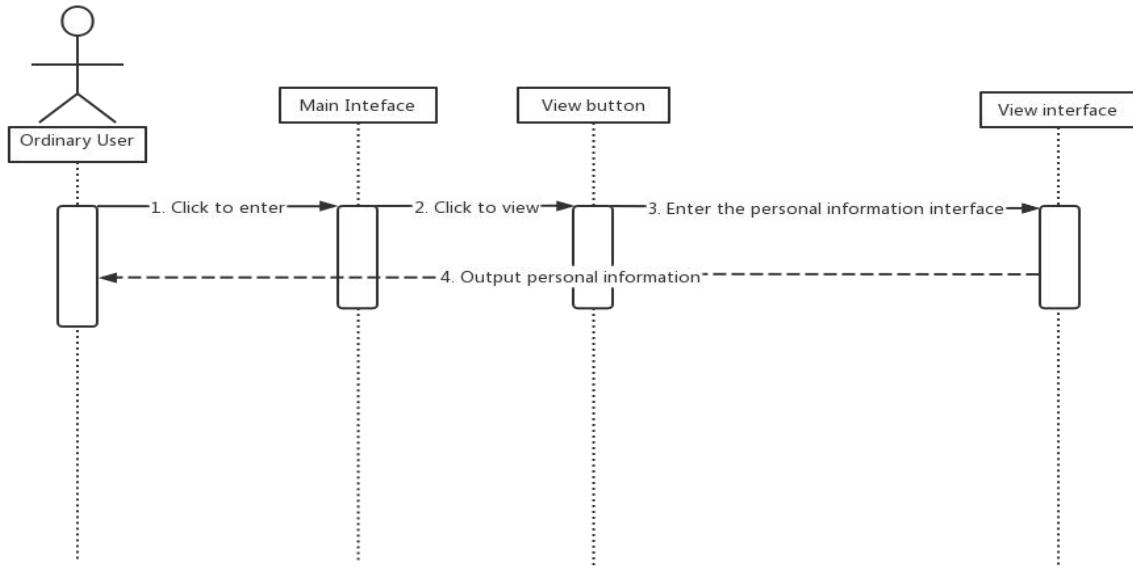
1) Registration



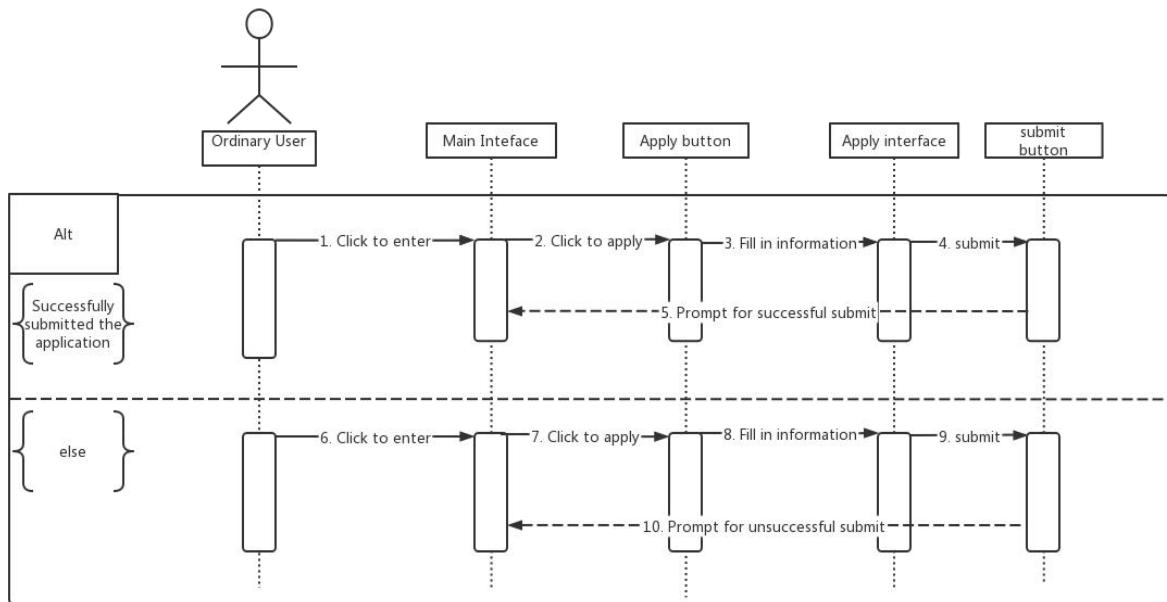
2) Login



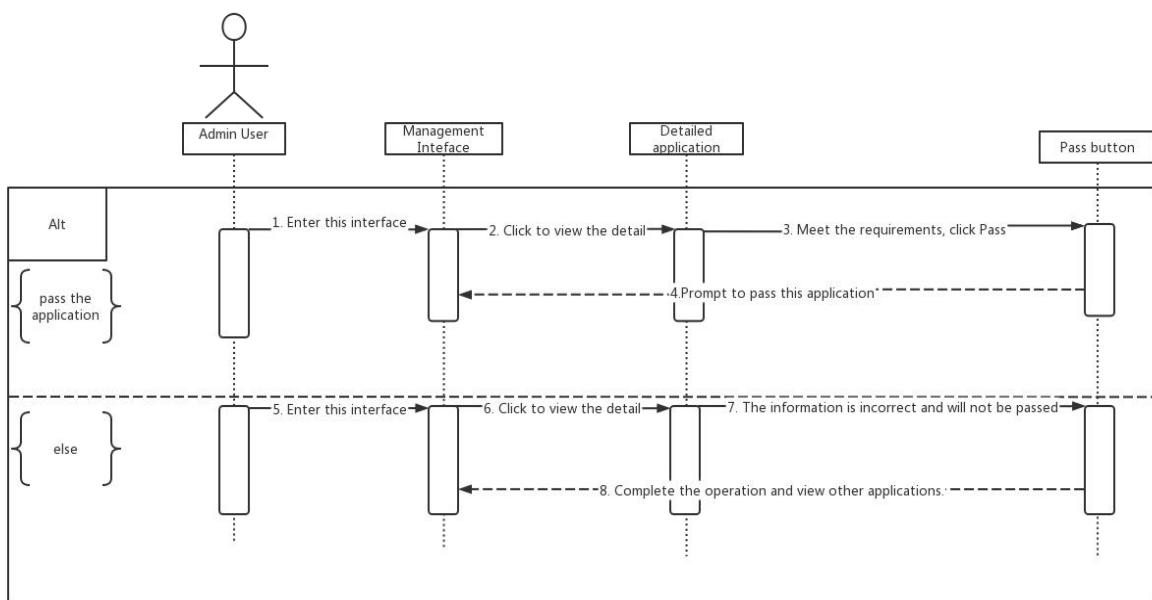
3) View Information



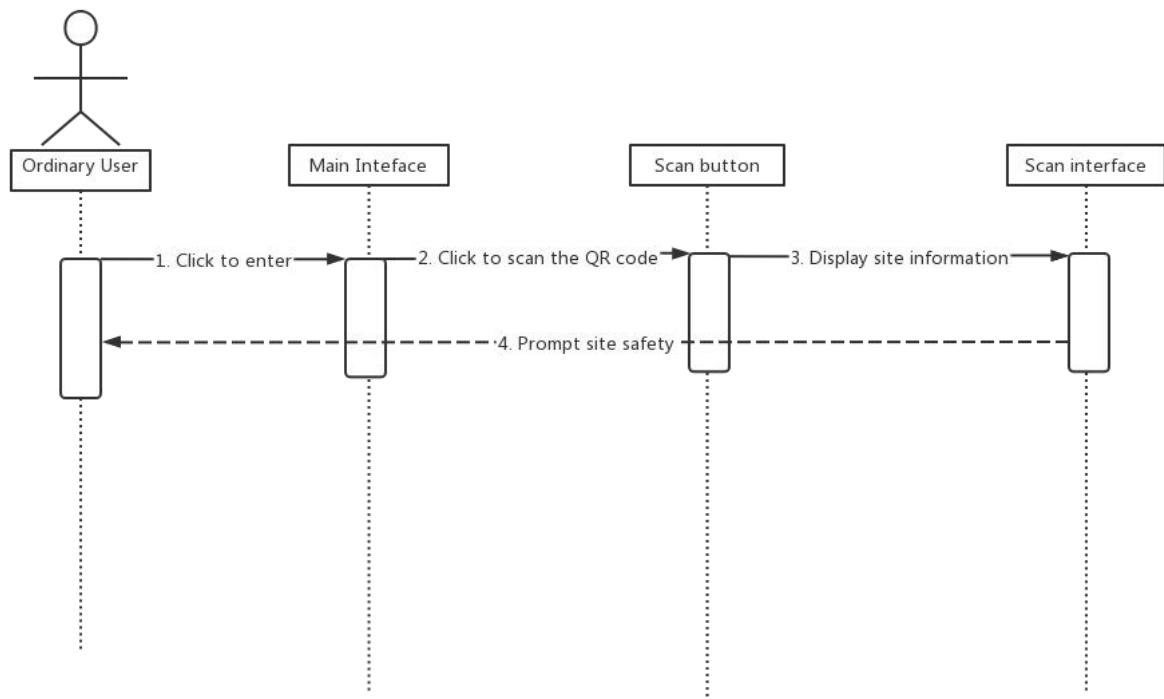
4) Apply for site code



5) Approval application



6) Scan



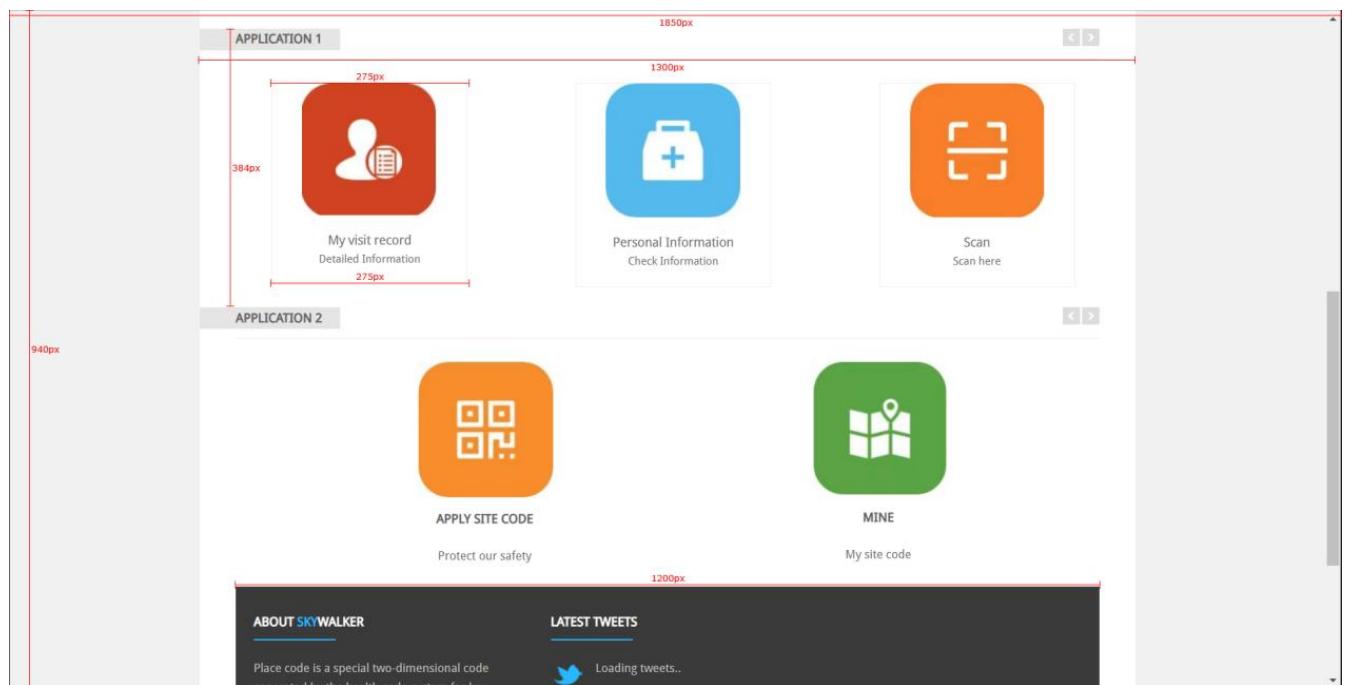
2.6 Systems design

2.6.1 UI Design

1) login interface

The login interface for the Epidemic Site Code Management System is a rectangular form with a white background. At the top center, it displays the system's name: "Epidemic Site Code Management System" in a dark blue font, with a red "1920px" dimension line above it. To the right of the title is a green "Register" button. Below the title, there is a logo for "Login to US". The main area contains three input fields: "Phone" (with placeholder "Enter your Phone for login..."), "Password" (with placeholder "Enter your Password for login..."), and "Verification code" (with placeholder "Enter your Verification code for login..."). Below these fields is a radio button group for "Admin" and "User", with "User" being selected. At the bottom is a green "Login" button with a white border and the text "318px" to its right. The entire form is enclosed in a red border. On the left side, there is a vertical dimension line labeled "941px". At the bottom center, a small copyright notice reads "Copyright © Epidemic Site Code Management System All rights reserved."

2) Main Interface



3) Registration Interface

1949px

941px

645px

Epidemic Site Code Management System.

[Login](#)

REGISTER TO US

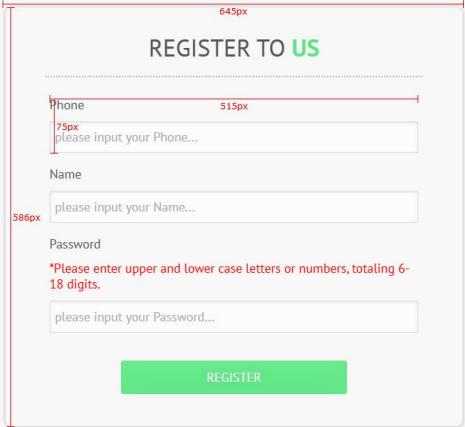
Phone 515px
75px
please input your Phone...

Name 586px
please input your Name...

Password
*Please enter upper and lower case letters or numbers, totaling 6-18 digits.
please input your Password...

REGISTER

Copyright © Epidemic Site Code Management System All rights reserved.



This screenshot shows the registration page of the Epidemic Site Code Management System. The page has a header with the system name and a 'Login' link. Below the header is a large registration form titled 'REGISTER TO US'. The form includes fields for 'Phone' (with placeholder 'please input your Phone...'), 'Name' (with placeholder 'please input your Name...'), and 'Password' (with placeholder 'please input your Password...'). A note above the password field specifies that it must be between 6 and 18 characters long and contain both uppercase and lowercase letters or numbers. A green 'REGISTER' button is at the bottom of the form. The entire registration area is enclosed in a light gray box with a red border. The overall width of the page is 1949px, and the registration box is 645px wide. The height of the registration box is 586px, and the total height of the page is 941px.

4) My Application Interface

1949px

940px

485px

720px

Epidemic Site Code Management System.

[Home](#)

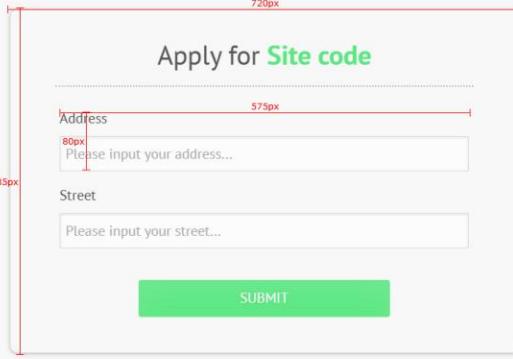
Apply for Site code

Address 575px
80px
Please input your address...

Street
Please input your street...

SUBMIT

Copyright © Epidemic Site Code Management System All rights reserved.



This screenshot shows the site code application page of the Epidemic Site Code Management System. The page has a header with the system name and a 'Home' link. Below the header is a form titled 'Apply for Site code'. It contains two input fields: 'Address' (placeholder 'Please input your address...') and 'Street' (placeholder 'Please input your street...'). A green 'SUBMIT' button is at the bottom of the form. The entire application area is enclosed in a light gray box with a red border. The overall width of the page is 1949px, and the application box is 720px wide. The height of the application box is 485px, and the total height of the page is 940px.

5) Application Interface

The screenshot shows a web-based application titled "Epidemic Site Code Management System". At the top, there is a search bar with a placeholder "Search" and a magnifying glass icon. To the right of the search bar is a user profile icon. The main content area is titled "Site verify" and contains a table with the following data:

Site ID	User ID	Address	Street	Operation
1	User object (19)	sdszcs	fsnl10086	<button>verify</button>

6) Personal Information Interface

The screenshot shows a personal information interface titled "Home". In the top right corner, there is a "Logout" link. The main content area features a large, stylized profile picture placeholder. Below the placeholder, the text "UserNo. : 24" is displayed in bold. A bulleted list provides the following information:

- Name: zhaoshirui
- Phone/Username: 18266850778
- Status: abnormal

At the bottom of the page, there are social sharing icons for Facebook, Twitter, LinkedIn, Google+, Baidu, and Weibo. A copyright notice at the very bottom reads: "© Production is only for operation by Zhao Shirui".

7) User Modify Interface

The screenshot shows a user modification interface for a management system. At the top, there is a header bar with the text "Epidemic Site Code Management System" and a search bar. Below the header is a table titled "User" with columns: User ID, Phone, Name, Status, and Modify. The table contains 11 rows of data. At the bottom of the table is a navigation footer with buttons for Prev, page... (set to 1), Next, and Skip.

User				
User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<button>modify</button>
2	1111111111	zsr	abnormal	<button>modify</button>
3	2222222222	123	health	<button>modify</button>
14	11111	name		<button>modify</button>
16	321321	123		<button>modify</button>
17	0	1	health	<button>modify</button>
18	3213213	123	health	<button>modify</button>
19	1111111	1	health	<button>modify</button>
20	8008208820	An Moli	health	<button>modify</button>
22	1234567890	z1	health	<button>modify</button>

375px

Prev 1 2 Next page... Skip

8) Site Modify Interface

The screenshot shows a site modification interface for a management system. At the top, there is a header bar with the text "Epidemic Site Code Management System" and a search bar. Below the header is a table titled "User" with columns: Site ID, User ID, Admin ID, Address, Street, Status, and Modify. The table contains 10 rows of data. At the bottom of the table is a navigation footer with buttons for Prev, page... (set to 1), Next, and Skip.

User						
Site ID	User ID	Admin ID	Address	Street	Status	Modify
1	User object (19)	Admin object (1)	sdszcs	qqjd	abnormal	<button>modify</button>
2	User object (19)	Admin object (1)	bjshdq	hyldj	health	<button>modify</button>
3	User object (19)	Admin object (1)	wadadaw	dwadawd	health	<button>modify</button>
4	User object (19)	Admin object (1)	fniqunf	1243u890	health	<button>modify</button>
5	User object (19)	Admin object (1)	18ceng	123	health	<button>modify</button>
6	User object (19)	Admin object (1)	18ceng1	1234	health	<button>modify</button>
7	User object (24)	Admin object (1)	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	<button>modify</button>
8	User object (24)	Admin object (1)	bjshdq	huayuanlijuedao	health	<button>modify</button>
9	User object (19)	Admin object (1)	18ceng23	12345	health	<button>modify</button>
10	User object (19)	Admin object (1)	1111111111111111	1111111111111111	abnormal	<button>modify</button>

308px

835px

Prev 1 2 Next page... Skip

9) User Site Interface

Epidemic Site Code Management System

Search Submit

zhaoshirui

My Site			
Site ID	Address	Street	Status
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health
8	bjshdq	huayuanlujiedao	health

187px 1425px 1849px 941px

10) QR Code

Epidemic Site Code Management System

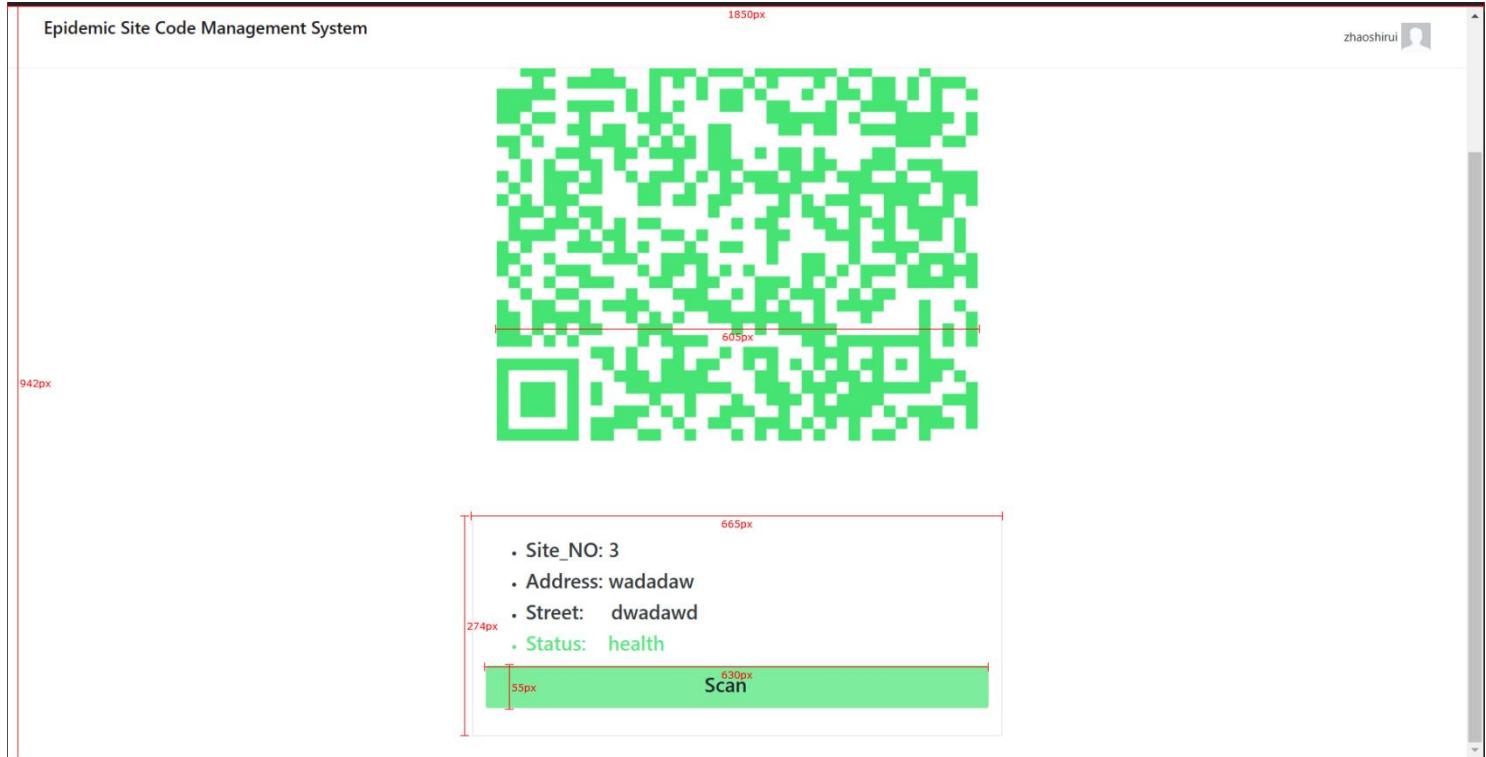
Search 305px Q 1849px zhaoshirui

Scan Site				
Site ID	Address	Street	Operation	
2	bjshdq	hyld	Scan	
3	wadadaw	dwadawd	Scan	
4	fniquqf	1243u890	Scan	
5	18ceng	123	Scan	
6	18ceng1	1234	Scan	
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	Scan	
8	bjshdq	huayuanlujiedao	Scan	
9	18ceng23	12345	Scan	

480px 1425px 1849px 940px

Prev 1 Next 340px page... Skip

11) QR Code Scan



11) Code scanning recording interface

A screenshot of a web-based application titled "Epidemic Site Code Management System". The header includes a search bar with placeholder text "Search time..." and a magnifying glass icon. A user profile icon is at the top right. The main content area is titled "User Visit Record" and contains a table with the following data:

Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
2	Site object (4)	User object (19)	March 12, 2023, 12:04 p.m.
3	Site object (1)	User object (19)	March 12, 2023, 8:10 p.m.
4	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
5	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
6	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
8	Site object (8)	User object (19)	March 14, 2023, 10:27 p.m.

At the bottom left, there are navigation buttons: "Prev" (45px), a blue "1" button (345px), "Next", "page...", and "Skip". The total width of the table is 1426px and its height is 465px.

2.6.2 data dictionary

There are 5 tables in this database: admin, site, sitead, user, us, where admin is the administrator table, site is the table of reviewed site codes, sitead is the table of user submitted applications, user is the table of user information records, and us is the table of user code access records.

1)Admin

The administrator data table is mainly used for the administrator to log in the account. There are three fields in the table, idadmin, adminno and password, with data types int, varchar (45) and varchar (200) respectively, which are used to record specific information about each administrator.

Result Grid							
	COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	EXTRA	COLUMN_COMMENT	COLUMN_DEFAULT	COLUMN_KEY
▶	idadmin	int	NO	auto_increment		NULL	PRI
	admin_no	varchar(45)	NO			NULL	
	password	varchar(200)	NO			NULL	

2)User

The common user table is mainly used for user login and can determine the user's health status. There are 5 fields in the table, iduser, phone, name, password, and status, and the data types are int, varchar(11), varchar(50), varchar(200), and varchar(45), which are used to record information about each user.

Result Grid							
	COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	EXTRA	COLUMN_COMMENT	COLUMN_DEFAULT	COLUMN_KEY
▶	iduser	int	NO	auto_increment		NULL	PRI
	phone	varchar(11)	NO			NULL	
	name	varchar(50)	NO			NULL	
	password	varchar(200)	NO			NULL	
	status	varchar(45)	NO		health		

3)Site

The site list is based on which user opened the site and which administrator passed the application and made records, including the detailed address, site status, etc. The data types are int, int, varchar(100), varchar(100), varchar(200), varchar(20), int, varchar(100). This table is used to record information about each site and the address of the site code.

Result Grid							
	COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	EXTRA	COLUMN_COMMENT	COLUMN_DEFAULT	COLUMN_KEY
▶	idsite	int	NO	auto_increment		NULL	PRI
	iduser	int	NO			NULL	MUL
	address	varchar(100)	NO			NULL	
	street	varchar(100)	NO			NULL	
	status	varchar(20)	NO		health		
	idadmin	int	NO			NULL	MUL
	path	varchar(100)	YES			NULL	

Foreign Key Name	Referenced Table	Column	Referenced Column
fk_admin	'epidemic_site_code_management_system': ...	<input type="checkbox"/> idsite	
fk_user	'epidemic_site_code_management_system': ...	<input type="checkbox"/> iduser	
		<input type="checkbox"/> address	
		<input type="checkbox"/> street	
		<input type="checkbox"/> status	
		<input checked="" type="checkbox"/> idadmin	idadmin

Foreign Key Name	Referenced Table	Column	Referenced Column
fk_admin	'epidemic_site_code_management_system': ...	<input type="checkbox"/> idsite	
fk_user	'epidemic_site_code_management_system': ...	<input checked="" type="checkbox"/> iduser	iduser
		<input type="checkbox"/> address	
		<input type="checkbox"/> street	
		<input type="checkbox"/> status	
		<input type="checkbox"/> idadmin	

4)US

When a user enters a place table (User_Site), when a user enters a place, it will be recorded in this table. The data types are int, int, int, datetime. this table is used to record the information of the user's scan and the administrator or user can view the corresponding scan record.

Result Grid							
	COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	EXTRA	COLUMN_COMMENT	COLUMN_DEFAULT	COLUMN_KEY
▶	idus	int	NO	auto_increment		NULL	PRI
	idsite	int	NO			NULL	MUL
	iduser	int	NO			NULL	MUL
	e_time	datetime	NO			NULL	

Foreign Key Name	Referenced Table	Column	Referenced Column
dk_user	'epidemic_site_code_management_system': ...	<input type="checkbox"/> idus	
fk_uss	'epidemic_site_code_management_system': ...	<input type="checkbox"/> idsite	
		<input checked="" type="checkbox"/> iduser	iduser
		<input type="checkbox"/> e_time	

Foreign Key Name	Referenced Table	Column	Referenced Column
dk_user	'epidemic_site_code_management_system': ...	<input type="checkbox"/> idus	
fk_uss	'epidemic_site_code_management_system': ...	<input checked="" type="checkbox"/> idsite	idsite
		<input type="checkbox"/> iduser	
		<input type="checkbox"/> e_time	

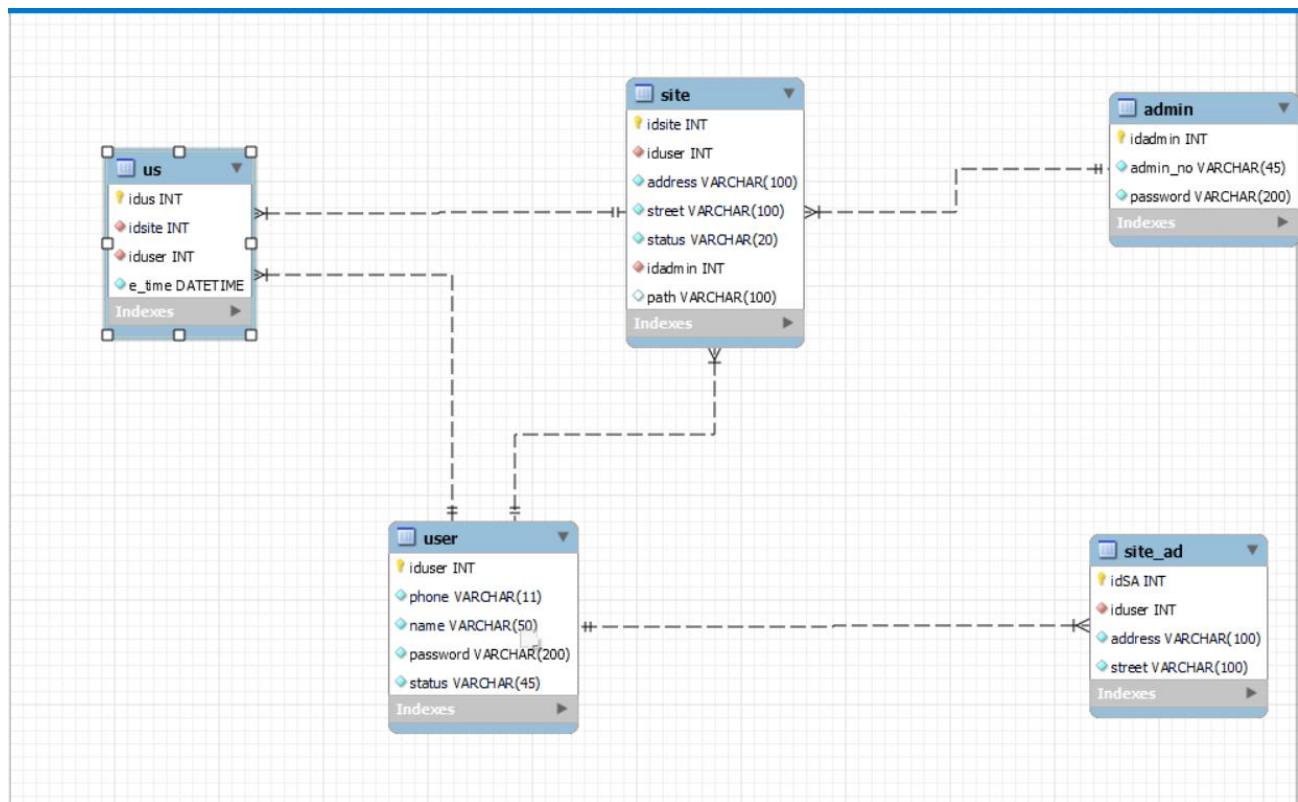
5)Site_ad

When a user applies for a location code, they first enter the data into this table, allowing the system to receive the data for the administrator's review. The data types are int, int, varchar(100) and varchar(100) respectively. This table is used to record user requests, which are stored in this table for review by the administrator and can be deleted once the review is complete.

Result Grid							
	COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	EXTRA	COLUMN_COMMENT	COLUMN_DEFAULT	COLUMN_KEY
▶	idSA	int	NO	auto_increment		NULL	PRI
	iduser	int	NO			NULL	MUL
	address	varchar(100)	NO			NULL	
	street	varchar(100)	NO			NULL	

Foreign Key Name	Referenced Table	Column	Referenced Column
fk_sad_user	'epidemic_site_code_management_system'`.`...	<input type="checkbox"/> idSA <input checked="" type="checkbox"/> iduser <input type="checkbox"/> address <input type="checkbox"/> street	iduser

2.6.3 E-R Model



2.6.4 Functional design

1)Connection

```

Server=localhost:3306;
Username=root;
Password= ;
Db=epidemic_site_code_management_system;
Link=mysqli_connect(Server,Username,Password,Db);
Sql="use epidemic_site_code_management_system"
Function my(sql)
{res=mysqli_query(Link,sql);
If(!res)
    Exit;
Return res;
}
  
```

2)Register function

Include Connection;

```

Name=name;
Phone=phone;
Password=password;
Sql=insert into user('null','Phone',Name,'Password','health');
my(Sql);
3)Login function
Include Connection;
Phone=phone;
Password=password;
If(role==user)
    Sql=select phone, password from user where phone=Phone,password=Password;
Else
    Sql=select admin_no , password from admin where admin_no=Phone,password=Password;
Result=my(Sql);
User=mysqli_fetch_assoc(result);
If(role==user&&User)
    {Header(url=main)
     Print login success
    }
Else If(role==admin&&User)
    {Header(url=management)
     Print login success
    }
Else
    Print login unsuccessful;

```

4)Apply function

```

Include Connection;
Picture=.../.../.../...jpg;
Address=haidianqushuangqinglu
Street=bjhkhtdxwlssl;
sql=insert into site values('null','iduser','address','street','status','idadmin');
Result=my(sql);
If(Result)
Print pass apply;
Else
Print apply unsuccessful;

```

5)Modify function

```

Include Connection;
Sql=update site set status=abnormal where id=1;
Result=my(Sql);
If(Result)
    Print modify successful;

```

```
Else
    Print modify unsuccessful;
```

5)View function

```
Include Connection;
Sql=Select idsite,,iduser,address,street,status from site;
Result=my(Sql);
Site=array();
While(r=mysqli_fetch_assoc(result))
    Site=r;
Include 'management.html';
```

6)Query function

```
Include Connection;
Phone=phone;
Sql=Select * from user where phone=Phone;
Result=my(Sql);
My_info=mysqli_fetch_assoc(Result);
Include post.html
```

7)Site function

```
Include Connection;
Idsite=idsite;
Sql=delete from site where idsite=Idsite;
Result=my(Sql);
If(Result)
    Print Log off successfully;
Else
    Print Log off unsuccessfully;
```

8)QR_Generation function

```
Include Connection;
Idsite=idsite;
Sql=Select address,street from site where idsite=Idsite;
Result=my(Sql);
r=mysqli_fetch_assoc(Result);
Generate_QR_Code(r[address],r[street]);
Return QR_code;
```

9)QR_Scan function

```
Include Connection;
Scan_QR_Code()
{
    Iduser=iduser;
    Sql=Insert into us values('null',Iduser,idsite,time_now());
    Result=my(Sql);
    If(Result)
```

Print Scan code successfully, pay attention to personal safety.
 Else
 Print Code scanning failed. Please take precautions.

3. Program Code

3.1 Overview

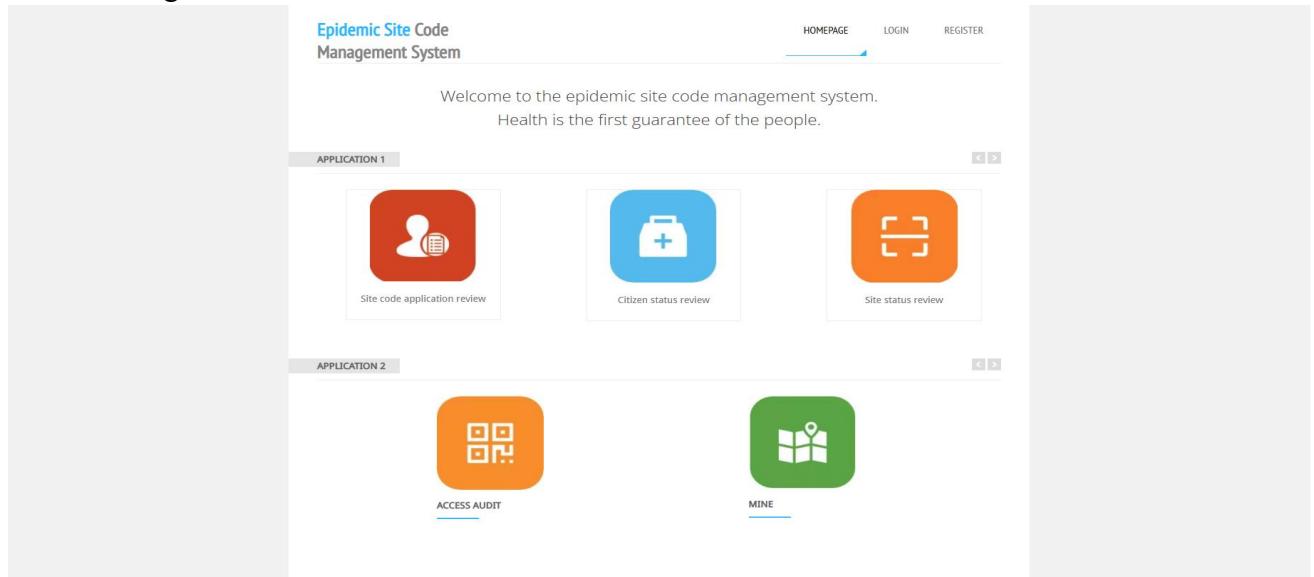
There are four apps in the epidemic prevention and control site code management system, namely index, user, QR code and supervise.

App Name	Comment
index	This app contains the user's main interface for using the system and the administrator's main interface, through which you can access other functions of the entire system.
user	This app is the main part of the user's registration and login function. It can also display personal information, view personal location code and apply.
Qr code	The main function of this app is to store the information of the place and generate the QR code. It can also decode the QR code and transmit the access record and generated information to the database.
supervise	This app is mainly used by administrators. In this module, the admin can view the information of users and places, and also view the user's access records. The function of applying through site code is necessary. This app can also modify the status of users and places.

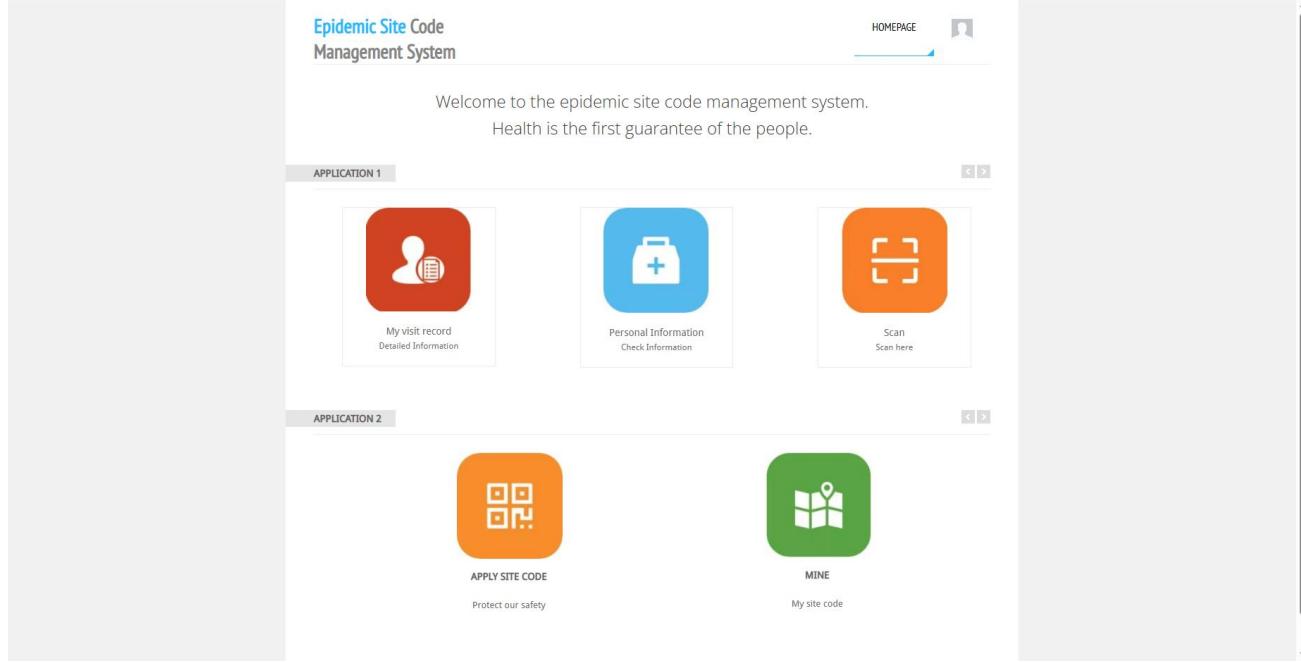
3.2 Index App

3.2.1 The Index Page for The User

If no user logs in, this will be the case.



If a user logs in, the upper right corner will change to the logged-in style.



3.2.1.1 The static page code is as follows:

```
{% load static %}  
<!DOCTYPE HTML>  
<!--[if lt IE 7 ]>  
<html class="ie ie6" lang="en"><![endif]-->  
<!--[if IE 7 ]>  
<html class="ie ie7" lang="en"><![endif]-->  
<!--[if IE 8 ]>  
<html class="ie ie8" lang="en"><![endif]-->  
<!--[if (gte IE 9)|(IE)]><!-->  
<html lang="en">  
<!--<![endif]-->  
  
<head>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
    <link class="TopmenuFont" href="#" rel="stylesheet" type="text/css"/>  
    <link class="TextFont" href="#" rel="stylesheet" type="text/css"/>  
    <link class="TitleFont" href="#" rel="stylesheet" type="text/css"/>  
    <link  
        href='http://fonts.googleapis.com/css?family=Lato:300,400,700,400italic|Droid+Sans|PT+Sans:400,700,400italic,700italic|PT+Sans+Narrow|Open+Sans:400,300'  
            rel='stylesheet' type='text/css'>  
    <script type="text/javascript" src="{% static 'javascript/jquery-latest.js' %}"></script>  
    <script type="text/javascript" src="{% static 'javascript/red-sky-options.js' %}"></script>  
    <script type="text/javascript" src="{% static 'shortcodes/shortcodes.js' %}"></script>  
    <script type="text/javascript" src="{% static 'javascript/jquery.prettyPhoto.js' %}"></script>  
<!-- Mobile Specific Metas -->
```

```

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
<!-- CSS -->
<!---[if lt IE 9]>
<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
<link href="{% static 'css/style.css' %}" rel="stylesheet" type="text/css">
<link href="{% static 'css/responsive.css' %}" rel="stylesheet" type="text/css">
<link href="{% static 'shortcodes/shortcodes.css' %}" rel="stylesheet" type="text/css">
<link href="{% static 'css/prettyPhoto.css' %}" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="../static/bootstrap/css/bootstrap.min.css">
<!-- START SETTINGS BOX -->
<script type="text/javascript" src="{% static 'settingsbox/farbtastic/farbtastic.js' %}"></script>
<script type="text/javascript" src="{% static 'settingsbox/settingsbox.js' %}"></script>
<link href="{% static 'settingsbox/settingsbox.css' %}" rel="stylesheet" type="text/css">
<link href="{% static 'settingsbox/farbtastic/farbtastic.css' %}" rel="stylesheet"
type="text/css">
<!-- END SETTINGS BOX -->

<title>Epidemic Site Code Management System</title>
</head>
<style>
.t {
    float: left;
    display: flex;
    margin-right: 80px !important;
    margin-left: 40px !important;
}

.disappear {
    float: left;
    display: flex;
    margin-left: 120px !important;
    margin-right: 100px !important;
}

#bor {
    margin-left: 5% !important;
}

.chai {
    margin-left: 0 !important;
}

</style>
<body>
<!-- START TOP HIDDEN AREA -->
<div id="box">
    <div class="container">

```

```

<!-- START HEADER -->
<div id="header">
    <div class="one-third columns">
        <div class="logo"><a href="<% url 'index' %>"><span>Epidemic Site
</span>Code Management System</a></div>
    </div>
    <div class="two-thirds_last columns">
        <ul class="topmenu">
            <li class="active"><a class="active" href="">Homepage</a></li>

            {%
                if not request.session.Info %
                    <li><a href="<% url 'login' %>">Login</a></li>
                    <li><a href="<% url 'register' %>">Register</a></li>
                {% else %}
                    <li><a href="#"></a>
                        <ul class="MenuDropdown">
                            <li><a href="<% url 'user_info' %>">Personal</a></li>
                            <li><a href="<% url 'logout' %>">Logout</a></li>
                        </ul>
                    </li>
                {% endif %}
            </ul>
        </div>
        <div class="clear"></div>
    </div>
    <!-- END HEADER -->
    <!-- START QUOTE -->
    <div class="quote one columns"> Welcome to the epidemic site code management
system.<br/>
        Health is the first guarantee of the people.
    </div>
    <!-- END QUOTE -->

    <!-- START WORKS -->
    <div class="works carousel">
        <div class="carouselNext"></div>
        <div class="carouselBack"></div>
        <div class="title">Application 1</div>
        <div class="clear"></div>
        <div class="titleline"></div>
        <div class="carouselView">
            <div class="carouselContainer chai" >
                <div class="columns four t">
                    <div class="work"><a href="/user/usersite" rel="prettyPhoto[a]"></a>
<h3><a href="/user/usersite">My visit record</a></h3>
<h4>Detailed Information</h4>
</div>
</div>
<div class="columns four t">
<div class="work"><a href="/user/info" rel="prettyPhoto[a]"></a>
<h3><a href="/user/info">Personal Information</a></h3>
<h4>Check Information</h4>
</div>
</div>
<div class="columns four t">
<div class="work"><a href="/QR/scan" rel="prettyPhoto[a]"></a>
<h3><a href="/QR/scan">Scan</a></h3>
<h4>Scan here</h4>
</div>
</div>
<div class="clear"></div>
</div>
</div>
</div>
<!-- END WORKS -->
<div class="blogs carousel">
<div class="carouselNext"></div>
<div class="carouselBack"></div>
<div class="title">Application 2</div>
<div class="clear"></div>
<div class="titleline"></div>
<div class="carouselView">
<div class="carouselContainer" id="bor"
style="width: 1954px; margin-left: 0px; transition: margin-left 0.2s linear
0s;">
<div class="columns four disappear">
<div class="newpost">
<div class="viewpost"><a href="/user/site">View</a></div>
<a href="/user/site" class="fade" title="This is the
description"><span></span></a>
<h2><a href="/user/site">Apply site code</a></h2>
<p>Protect our safety</p>

```

```

        </div>
    </div>
    <div class="columns four disappear">
        <div class="newpost">
            <div class="viewpost"><a href="/user/mysite">View</a></div>
            <a href="/user/mysite" class="fade" title="This is the
description"><span></span><img
                alt="cards"></a>
                    <h2><a href="/user/mysite">Mine</a></h2>
                    <p>My site code</p>
                </div>
            </div>
            <div class="clear"></div>
        </div>
    </div>
</div>

<!-- START BOTTOMAREA -->
<div style="display:none">
    <script src='http://v7.cnzz.com/stat.php?id=155540&web_id=155540'
language='JavaScript'
        charset='gb2312'></script>
</div>
<!-- END BOTTOMAREA -->
</div>
</div>
</body>
</html>

```

3.2.1.2 The url path is as follows.

path('index/', index.views.index, name='index'),

3.2.1.3 The view function is as follows

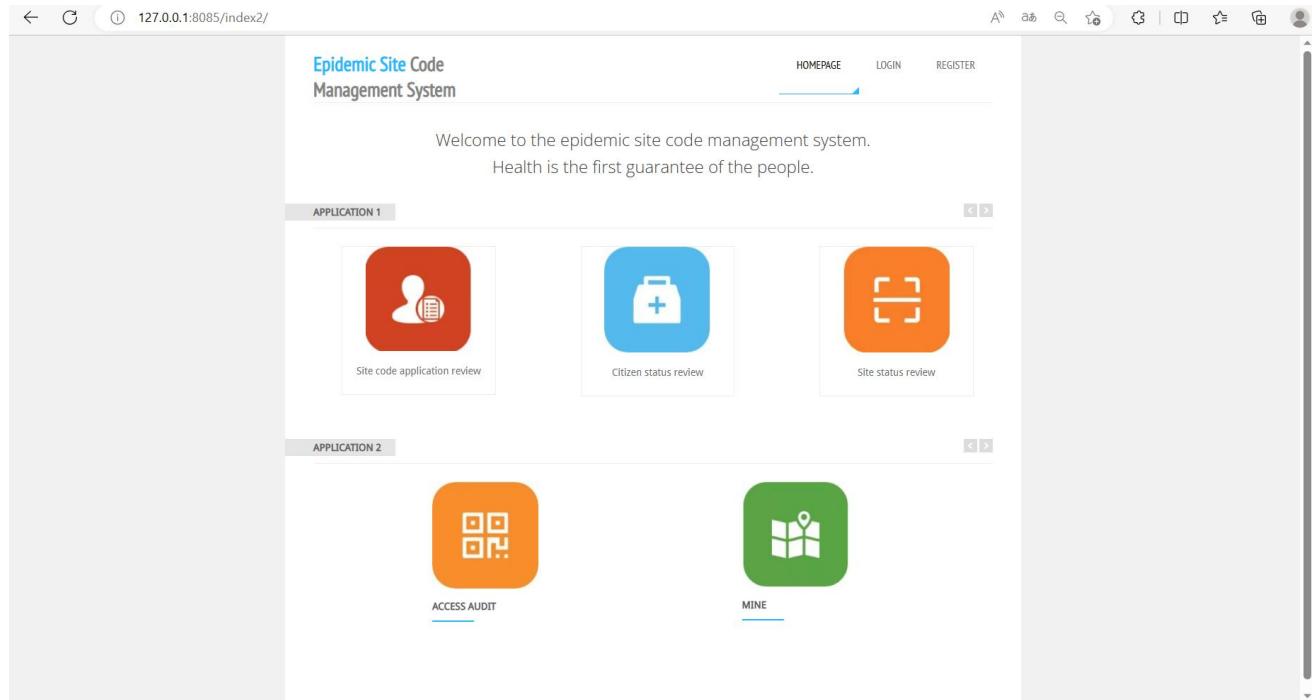
```

from django.shortcuts import render
"""This is the main page view function,
accessed through the URL path,
and then returned to the user index interface."""
def index(req):
    return render(req, "index.html")

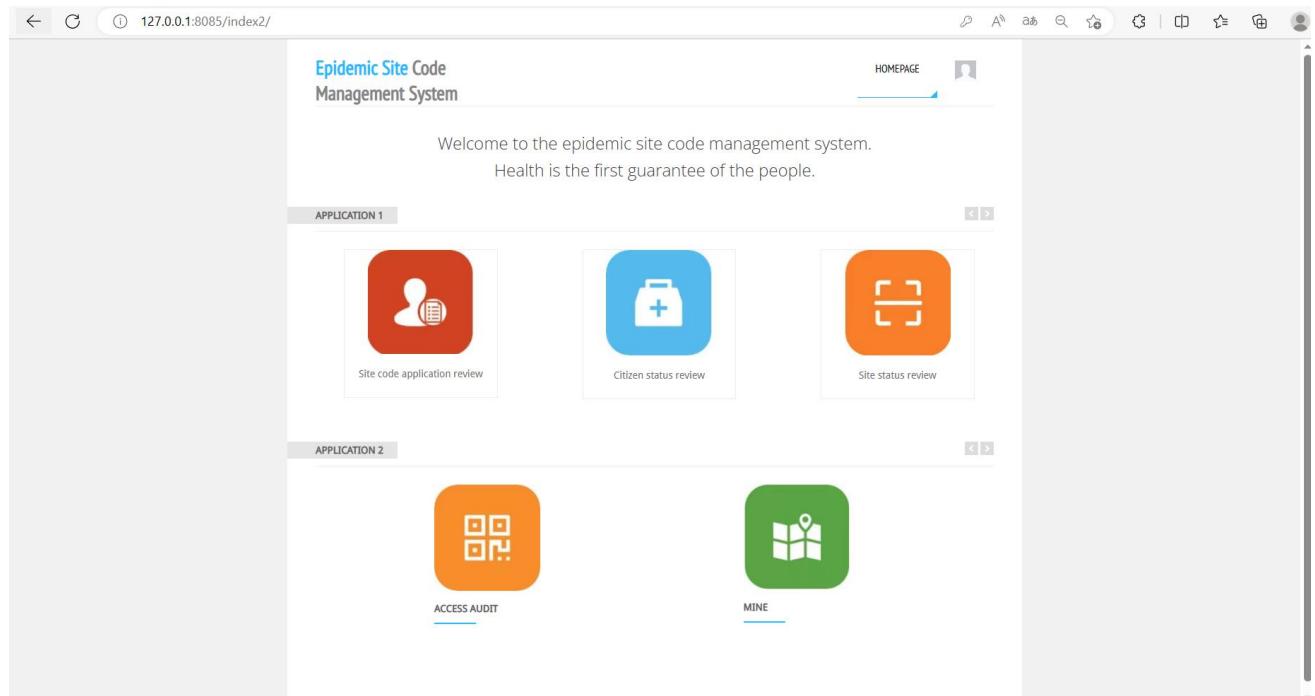
```

3.2.2 The Index Page for The Admin

This is the status that the administrator's function interface is not logged in. It looks the same as the user interface, but its functions are different.



This is the login status of the administrator's function interface.



3.2.2.1 The static page code is as follows:

```
{% load static %}  
<!DOCTYPE HTML>
```

```

<!--[if lt IE 7 ]>
<html class="ie ie6" lang="en"> <![endif]-->
<!--[if IE 7 ]>
<html class="ie ie7" lang="en"> <![endif]-->
<!--[if IE 8 ]>
<html class="ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 9)|(IE)]><!-->
<html lang="en">
<!--<![endif]-->
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <link class="TopmenuFont" href="#" rel="stylesheet" type="text/css"/>
    <link class="TextFont" href="#" rel="stylesheet" type="text/css"/>
    <link class="TitleFont" href="#" rel="stylesheet" type="text/css"/>
    <link
        href='http://fonts.googleapis.com/css?family=Lato:300,400,700,400italic|Droid+Sans|PT+Sans:400,700,400italic,700italic|PT+Sans+Narrow|Open+Sans:400,300'
        rel='stylesheet' type='text/css'>
    <script type="text/javascript" src="<% static 'javascript/jquery-latest.js' %>"></script>
    <script type="text/javascript" src="<% static 'javascript/red-sky-options.js' %>"></script>
    <script type="text/javascript" src="<% static 'shortcodes/shortcodes.js' %>"></script>
    <script type="text/javascript" src="<% static 'javascript/jquery.prettyPhoto.js' %>"></script>
    <!-- Mobile Specific Metas -->
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <!-- CSS -->
    <!--[if lt IE 9]>
        <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
        <link href="<% static 'css/style.css' %>" rel="stylesheet" type="text/css">
        <link href="<% static 'css/responsive.css' %>" rel="stylesheet" type="text/css">
        <link href="<% static 'shortcodes/shortcodes.css' %>" rel="stylesheet" type="text/css">
        <link href="<% static 'css/prettyPhoto.css' %>" rel="stylesheet" type="text/css">
        <link rel="stylesheet" href="../static/bootstrap/css/bootstrap.min.css">
    <!-- START SETTINGS BOX -->
        <script type="text/javascript" src="<% static 'settingsbox/farbtastic/farbtastic.js' %>"></script>
        <script type="text/javascript" src="<% static 'settingsbox/settingsbox.js' %>"></script>
        <link href="<% static 'settingsbox/settingsbox.css' %>" rel="stylesheet" type="text/css">
        <link href="<% static 'settingsbox/farbtastic/farbtastic.css' %>" rel="stylesheet"
type="text/css">
    <!-- END SETTINGS BOX -->

    <title>Epidemic Site Code Management System</title>
</head>
<style>
    .t {
        float: left;
        display: flex;

```

```

margin-right: 80px !important;
margin-left: 40px !important;
}

.disappear {
  float: left;
  display: flex;
  margin-left: 120px !important;
  margin-right: 100px !important;
}

#bor {
  margin-left: 5% !important;
}

.chai{
  margin-left: 0 !important;
}

</style>
<body>

<!-- START TOP HIDDEN AREA -->

<div id="box">
  <div class="container">
    <!-- START HEADER -->
    <div id="header">
      <div class="one-third columns">
        <div class="logo"><a href="#"><% url 'index' %}><span>Epidemic Site
</span>Code Management System</a></div>
      </div>
      <div class="two-thirds_last columns">

        <ul class="topmenu">
          <li class="active"><a class="active" href="#">Homepage</a></li>

          {%
            if not request.session.Info %
          }
          <li><a href="#"><% url 'login' %}>Login</a></li>
          <li><a href="#"><% url 'register' %}>Register</a></li>
        {%
          else %
        }
        <li><a href="#"><% static 'images/av.png' %}>
width="30"></a>
          <ul class="MenuDropdown">
            <li><a href="#"><% url 'user_info' %}>Personal</a></li>

```

```

                <li><a href="#"><% url 'logout' %}>Logout</a></li>
            </ul>
        </li>
    {% endif %}
</ul>
</div>
<div class="clear"></div>
</div>
<!-- END HEADER --&gt;

<!-- START QUOTE --&gt;
&lt;div class="quote one columns"&gt; Welcome to the epidemic site code management
system.&lt;br/&gt;
    Health is the first guarantee of the people.
&lt;/div&gt;
<!-- END QUOTE --&gt;

<!-- START WORKS --&gt;
&lt;div class="works carousel"&gt;
    &lt;div class="carouselNext"&gt;&lt;/div&gt;
    &lt;div class="carouselBack"&gt;&lt;/div&gt;
    &lt;div class="title"&gt;Application 1&lt;/div&gt;
    &lt;div class="clear"&gt;&lt;/div&gt;
    &lt;div class="titleline"&gt;&lt;/div&gt;
    &lt;div class="carouselView"&gt;
        &lt;div class="carouselContainer chai" &gt;
            &lt;div class="columns four t"&gt;
                &lt;div class="work"&gt;&lt;a href="/SP/verify" rel="prettyPhoto[a]"&gt;&lt;img
                    src="#"><% static 'images/images/L_R.jpg' %}>
                    alt="cards"/></a>
                <h3><a href="/SP/verify">Site code application review</a></h3>
                </div>
            </div>
            <div class="columns four t">
                <div class="work"><a href="/SP/user_status"
rel="prettyPhoto[a]"><% static 'images/images/PersonInfo.jpg' %}>
                    alt="cards"/></a>
                <h3><a href="/SP/user_status">Citizen status review</a></h3>
                </div>
            </div>
            <div class="columns four t">
                <div class="work"><a href="/SP/site_status"
rel="prettyPhoto[a]"><% static 'images/images/Scan.jpg' %}>
                    alt="cards"/></a>

```

```

        <h3><a href="/SP/site_status">Site status review</a></h3>
        </div>
    </div>

        <div class="clear"></div>
    </div>
</div>
<!-- END WORKS -->

<div class="blogs carousel">
    <div class="carouselNext"></div>
    <div class="carouselBack"></div>
    <div class="title">Application 2</div>
    <div class="clear"></div>
    <div class="titleline"></div>
    <div class="carouselView">
        <div class="carouselContainer" id="bor"
            style="width: 1954px; margin-left: 0px; transition: margin-left 0.2s linear
0s;">
            <div class="columns four disappear">
                <div class="newpost">
                    <div class="viewpost"><a href="/SP/us_status">View</a></div>
                    <a href="/SP/us_status" class="fade" title="This is the
description"><span></span></a>
                <h2><a href="/SP/us_status">Access audit</a></h2>
                </div>
            </div>
            <div class="columns four disappear">
                <div class="newpost">
                    <div class="viewpost"><a href="/user/info">View</a></div>
                    <a href="/user/info" class="fade" title="This is the
description"><span></span></a>
                <h2><a href="/user/info">Mine</a></h2>
                </div>
            </div>
            <div class="clear"></div>
        </div>
    </div>
</div>
[
```

```

<!-- START BOTTOMAREA -->

<div style="display:none">
    <script src='http://v7.cnzz.com/stat.php?id=155540&web_id=155540'
language='JavaScript'
        charset='gb2312'></script>
</div>
<!-- END BOTTOMAREA -->
</div>
</body>

</html>

```

3.2.2.2 The url path is as follows.

```
path('index2/', index.views.index2, name='index2'),
```

3.2.2.3 The view function is as follows

```

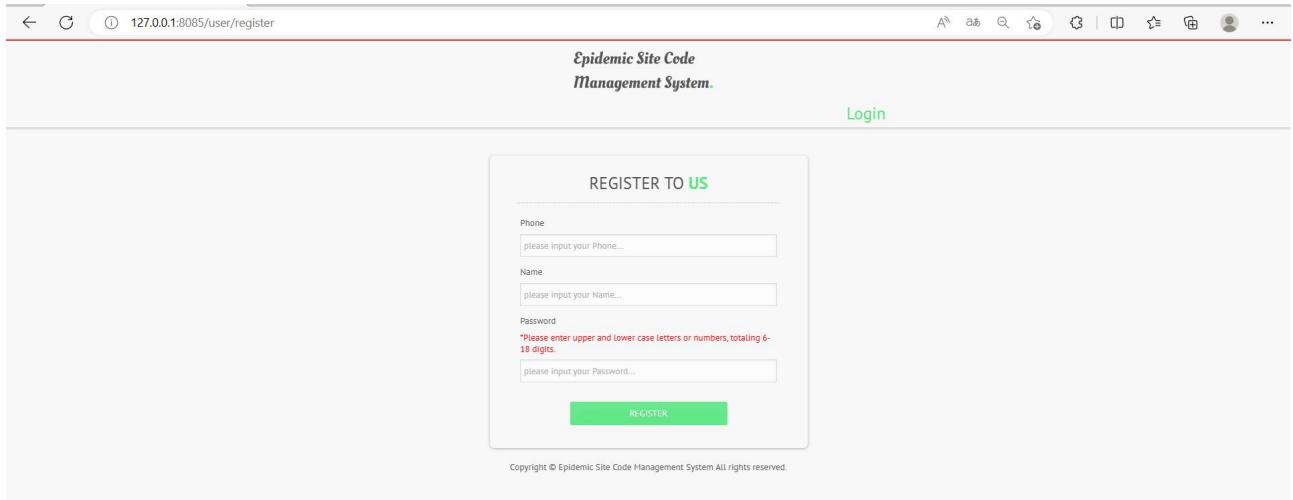
from django.shortcuts import render
"""This is the admin main page view function,
accessed through the URL path,
and then returned to the admin index interface."""
def index2(req):
    return render(req, "index2.html")

```

3.3 User App

3.3.1 The Register Page

This page is the user registration interface. Users can register through this interface.



3.3.1.1 Front End Display Interface Code

```

{%
load static %
}
<!DOCTYPE html>
<html lang="en">

<head>

<meta charset="utf-8">
<title>Register</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">

<!-- CSS -->
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=PT+Sans:400,700'>
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=Oleo+Script:400,700'>
<link rel="stylesheet" href="{% static 'bootstrap/css/bootstrap1.min.css' %}">
<link rel="stylesheet" href="{% static 'css/style1.css' %}">
<script src="{% static 'js/bootstrap.js' %}"></script>
<!-- HTML5 shim, for IE6-8 support of HTML5 elements -->
<!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->

</head>

<style>
.note{
    color: red;
}

.green{
    color: #5de784;
}

```

```

    }
    #bgc_c{
        background-color: #5de784;
    }
</style>
<body>
    <div class="header">
        <div class="container">
            <div class="row">
                <div class="logo span4">
                    <h1><a href="/index">Epidemic Site Code Management System<span
class="green">.</span></a></h1>
                </div>
                <div class="links span8">
                    <a class="home green" href="/user/login" rel="tooltip"
data-placement="bottom" data-original-title="Home">Login</a>
                </div>
            </div>
        </div>
        <div class="register-container container">
            <div class="row">
                <div class="register span6">
                    <form method="post" novalidate>
                        <h2>REGISTER TO <span
class="green"><strong>US</strong></span></h2>
                        {{ csrf_token }}
                        {{ for field in form }}
                        <label>{{ field.label }} </label>
                        {{ if field.label == "Password"}}
                            <label class="note">*Please enter upper and lower case
letters or numbers, totaling 6-18 digits.</label>
                            {{ endif }}
                            {{ field }}
                        <span style="color: red">{{ field.errors.0 }}</span>
                        {{ endfor }}
                        <div>
                            <button type="submit" id="bgc_c">REGISTER</button>
                        </div>
                    </form>
                </div>
            </div>
        </div>
        <div align="center">Copyright &copy; Epidemic Site Code Management System All rights
reserved.<a target="_blank" href="#"></a>
    </div>

```

```

<!-- Javascript -->
<script src="<% static 'js/jquery1-1.8.2.min.js' %>"></script>
<script src="<% static '/bootstrap/js/bootstrap1.min.js' %>"></script>
<script src="<% static 'js/jquery1.backstretch.min.js' %>"></script>
<script src="<% static 'js/scripts1.js' %>"></script>
</body>
</html>

```

3.3.1.2 The url path is as follows.

```
path('user/register', user.views.user_register, name='register'),
```

3.3.1.3 The view function is as follows

"""This is a registration view function that guides the user to the registration interface.

After entering reasonable information, the user can register an account,
and the entered information is entered into the database."""

```

def user_register(req):
    if req.method == "GET":
        form = MyForm()
        return render(req, "user_register.html", {"form": form})
    form = MyForm(data=req.POST)
    if form.is_valid():
        phone = req.POST.get("phone")
        name = req.POST.get("name")
        password = pwd_encrypt(req.POST.get("password"))
        User.objects.create(phone=phone, name=name, password=password, status="health")
        return redirect("/user/login")
    else:
        return render(req, "user_register.html", {"form": form})

```

The MD5 encryption method is also used.

"""This is a password encryption function
that uses MD5 encryption to encrypt the user's password."""

```

def pwd_encrypt(pwd):
    md5 = hashlib.md5()
    md5.update(pwd.encode())
    result = md5.hexdigest()
    return result

```

The model form class is used for tables in the database so that they can be displayed on front-end pages with some selectors.

"""This is a register class that inherits from ModelForm.
It can display information boxes on front-end pages,
as well as functions that validate data input types."""

```

class MyForm(forms.ModelForm):
    password = forms.CharField(label="Password",
                               validators=[RegexValidator(r'^[0-9a-zA-Z]{6,18}$')])
    phone = forms.CharField(label="Phone", validators=[RegexValidator(r'^[0-9]{7,12}$')])

    class Meta:
        model = User
        fields = ["phone", "name", "password"]

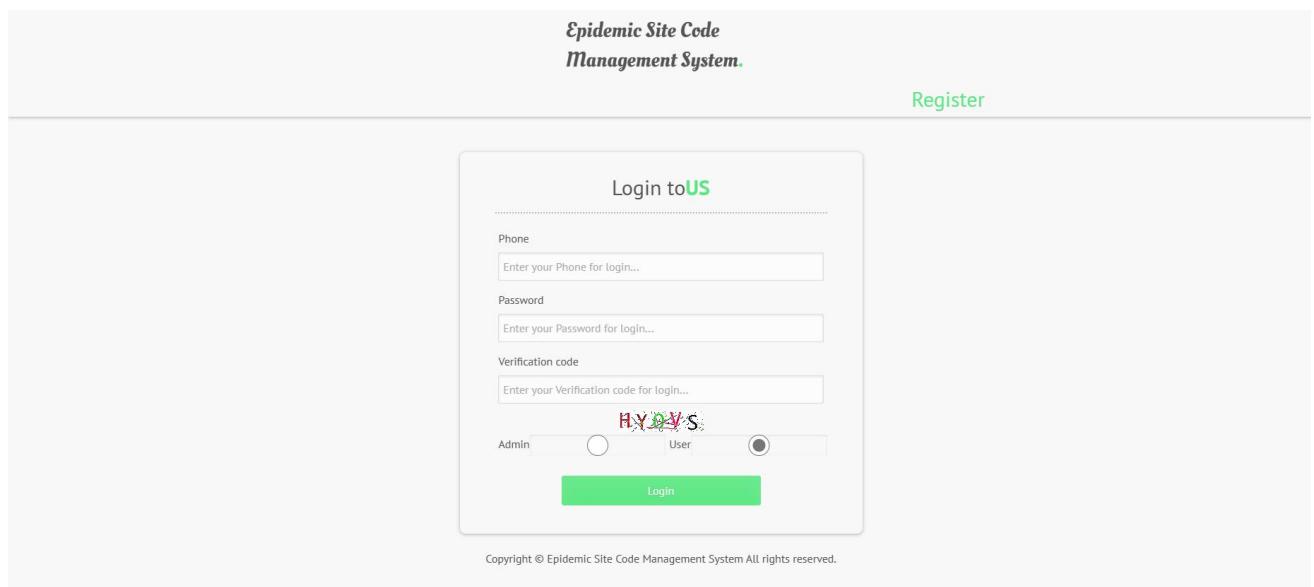
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for name, field in self.fields.items():
            if name == "password":
                field.widget.input_type = 'password'

            field.widget.attrs = {"placeholder": "please input your " + field.label + "..."}

```

3.3.2 The Login Page

This is the login interface where users can log in and use other functions after entering the correct username and password (they cannot use other functions before logging in).



3.3.2.1 Front End Display Interface Code

```

{%
load static %}

<!DOCTYPE html>
<html lang="en">

<head>

<meta charset="utf-8">

```

```

<title>Login</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">
<!-- CSS -->
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=PT+Sans:400,700'>
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=Oleo+Script:400,700'>
<link rel="stylesheet" href="<% static 'bootstrap/css/bootstrap1.min.css' %}">
<link rel="stylesheet" href="<% static 'css/style1.css' %}">
<script src="<% static 'js/bootstrap.js' %}"></script>
<!-- HTML5 shim, for IE6-8 support of HTML5 elements -->
<!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<style>
#Role
{
    display: flex;
}
.green{
    color: #5de784;
}
#bgc_c{
    background-color: #5de784;
}
</style>

<body>

<div class="header">
    <div class="container">
        <div class="row">
            <div class="logo span4">
                <h1><a href="<% url 'index' %}" class="green">Epidemic Site Code
Management System<span class="green">.</span></a></h1>
            </div>
            <div class="links span8">
                <a class="green" href="/user/register" rel="tooltip"
data-placement="bottom" data-original-title="Home">Register</a>
            </div>
        </div>
    </div>
</div>

```

```

<div class="register-container container">
    <div class="row">
        <div class="register span6">
            <form method="post" novalidate>
                {% csrf_token %}
                <h2>Login to<span class="green"><strong>US</strong></span></h2>
                {% for field in frm %}
                    <label>{{ field.label }}</label>
                    {{ field }}
                    {% if field.name == "code" %}
                        
                    {% endif %}
                    <span style="color: red">{{ field.errors.0 }}</span>
                {% endfor %}
                <div id="Role">
                    <label for="Role">Admin</label>
                    <input type="radio" id="admin" name="role" value="admin" >
                    <label for="Role">User</label>
                    <input type="radio" id="user" name="role" value="user" checked="checked">
                </div>

                <div>
                    <button type="submit" id="bgc_c" >Login</button>
                </div>
            </form>
        </div>
        </div>
        </div>
        <div align="center">Copyright ©; Epidemic Site Code Management System All rights reserved.<a target="_blank" href="#"></a>
    </div>
</body>
</html>

```

3.3.2.2 The url path is as follows.

path('user/login', user.views.user_login, name='login'),

3.3.2.3 The view function is as follows

"""This is a login view function that guides the user to the login interface.

After entering the correct user name and password,
users can log in to the home page and use other functions."""

```
def user_login(req):
    if req.method == "GET":
        frm = Login_Form()
        return render(req, "user_login.html", {"frm": frm}, )
    frm = Login_Form(data=req.POST)
    if frm.is_valid():
        username = req.POST.get("phone")
        role = req.POST.get("role")
        code = req.session.get('image_code', "")
        user_code = req.POST.get("code")
        if user_code.upper() != code:
            frm.add_error("code", "Verification code error")
            return render(req, "user_login.html", {"frm": frm})
    if role == "user":
        password = pwd_encrypt(req.POST.get("password"))
        value = models.User.objects.filter(phone=username, password=password)
        if value:
            userInfo = models.User.objects.filter(phone=username).values('iduser', 'name',
            'status')
            req.session["Info"] = {'id': userInfo[0]['iduser'], 'phone': username,
            'name': userInfo[0]['name'], 'status':
            userInfo[0]['status'],
            'role': role}
            return redirect("/index")
        else:
            return render(req, "user_login.html", {"frm": frm})
    else:
        password = req.POST.get("password")
        value = models.Admin.objects.filter(admin_no=username, password=password)
        if value:
            userInfo = models.Admin.objects.filter(admin_no=username).values('idadmin',
            'admin_no')
            req.session["Info"] = {'id': userInfo[0]['idadmin'], 'phone':
            userInfo[0]['admin_no'], 'role': role}
            return redirect("/index2")
        else:
            return render(req, "user_login.html", {"frm": frm})

    else:
        return render(req, "user_login.html", {"frm": frm})
```

The model form class is used for tables in the database so that they can be displayed on front-end pages with some selectors.

"""This is a login class that inherits from ModelForm.

It can display information boxes on front-end pages,
as well as functions that validate data input types."""

```

class Login_Form(forms.ModelForm):
    class Meta:
        model = User
        fields = ["phone", "password"]

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.p = None
        for name, field in self.fields.items():
            if name == "password":
                field.widget.input_type = 'password'
                field.widget.attrs = {"placeholder": "Enter your " + field.label + " for login..."}

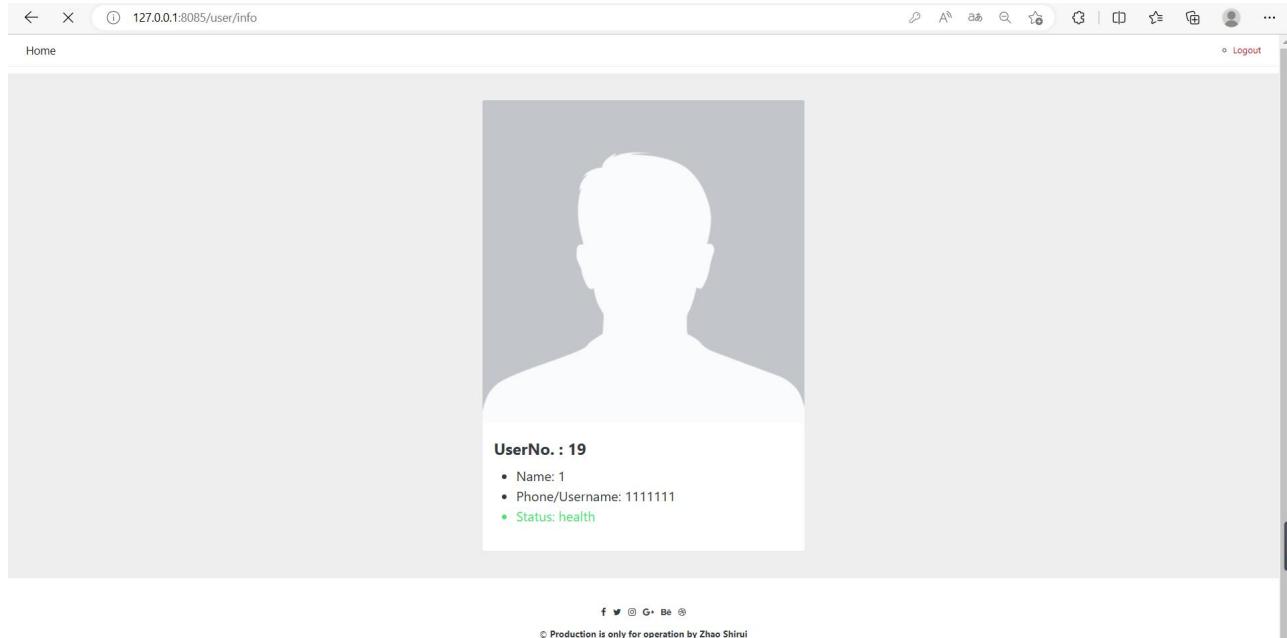
    def clean_phone(self):
        txt_phone = self.cleaned_data["phone"]
        if self.data.get("role") == "user":
            query = User.objects.filter(phone=txt_phone)
            if not bool(query):
                # print(self.cleaned_data)
                raise ValidationError("Wrong phone number!")
            self.p = txt_phone
            return self.p
        else:
            query = Admin.objects.filter(admin_no=txt_phone)
            if not bool(query):
                raise ValidationError("Login without permission!")
            self.p = txt_phone
            return self.p

    def clean_password(self):
        if self.p is not None:
            if self.data.get("role") == "user":
                txt_password = pwd_encrypt(self.cleaned_data["password"])
                query = User.objects.filter(phone=self.p, password=txt_password)
                if not bool(query):
                    raise ValidationError("Please enter the correct password!")
                return txt_password
            txt_password = self.cleaned_data["password"]
            query = Admin.objects.filter(admin_no=self.p, password=txt_password)
            if not bool(query):
                raise ValidationError("Please enter the correct password!")
            return txt_password
        raise ValidationError("Please enter the correct user name first")

```

3.3.3 The Info Page

This interface can display user's personal information, including id, name, etc.



3.3.3.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <title>PersonInfo</title>  
    <script type="text/javascript"> (function() { var css = document.createElement('link'); css.href = 'https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css'; css.rel = 'stylesheet'; css.type = 'text/css'; document.getElementsByTagName('head')[0].appendChild(css); })();  
    </script>  
    <link rel="stylesheet" href="{% static 'css/app.css' %}">  
    <link rel="stylesheet" href="{% static 'css/theme.css' %}">  
</head>  
<style>  
    #submit  
    {  
        cursor: pointer;  
        margin: 5px;  
        padding: 5px;  
        background: #00FFFF;  
        -moz-border-radius: 3px;  
        -webkit-border-radius: 3px;  
    }  
</style>
```

```

border-radius: 3px;
border: 0;
-moz-box-shadow: 0 15px 30px 0 rgba(255,255,255,.1) inset;
-webkit-box-shadow: 0 15px 30px 0 rgba(255,255,255,.1) inset;
box-shadow: 0 15px 30px 0 rgba(255,255,255,.1) inset;
font-family: 'PT Sans', Helvetica, Arial, sans-serif;
font-size: 16px;
font-weight: 400;
color: #fff;
text-shadow: 0 1px 2px rgba(0,0,0,.1);
-o-transition: all .2s;
-moz-transition: all .2s;
-webkit-transition: all .2s;
-ms-transition: all .2s;
}

.cyan
{
    color: #00FFFF;
}
#cyan{
    color: #00FFFF;
}
.lightgreen{
    color: #46e473;
}
.big{
    font-size: 24px;
    font-weight: normal;
    font-style: normal;
}

```

</style>

<body>

```

<nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
    <a class="navbar-brand font-weight-bolder mr-3"
        {%
            if request.session.Info.role == "user" %
        %}
        href="{% url 'index' %}"
        {%
            else %
        %}
        href="{% url 'index2' %}"
        {%
            endif %
        }
        >Home</a>
    <button class="navbar-light navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbarsDefault" aria-controls="navbarsDefault" aria-expanded="false"
        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

```

```

<div class="collapse navbar-collapse" id="navbarsDefault">

    <ul class="navbar-nav ml-auto align-items-center">
        {% if not request.session %}
            <li><a href="{% url 'login' %}">Login</a></li>
            <li><a href="{% url 'register' %}">Register</a></li>
        {% else %}
            <ul class="MenuDropdown">
                <li><a href="{% url 'logout' %}">Logout</a></li>
            </ul>
        {% endif %}
    </ul>
</div>
</nav>
<main role="main">

    <section class="bg-gray200 pt-5 pb-5">
        <div class="container">
            <div class="row justify-content-center">
                <div class="col-md-7">
                    <article class="card">
                        
                        <div class="card-body">
                            <h1 class="card-title display-4">UserNo. : {{ request.session.Info.id }} </h1>
                            <ul>
                                <li class="big">Name: {{ request.session.Info.name }}</li>
                                <li class="big">Phone/Username: {{ request.session.Info.phone }}</li>
                                <li class="lightgreen big">Status: {{ request.session.Info.status }}</li>
                            </ul>
                        </div>
                    </article>
                </div>
            </div>
        </div>
        <div id="comments" class="mt-4">
            <div id="disqus_thread">
            </div>
            <script type="text/javascript">
                var disqus_shortname = 'demoweb';
                var disqus_developer = 0;
                (function() {
                    var dsq = document.createElement('script'); dsq.type =
                    'text/javascript'; dsq.async = true;
                    dsq.src = window.location.protocol + '//' + disqus_shortname
                    + '.disqus.com/embed.js';
            
```

```

        (document.getElementsByTagName('head')[0] ||
document.getElementsByTagName('body')[0]).appendChild(dsq);
    })();
</script>
<noscript>
    Please enable JavaScript to view the <a href="http://disqus.com/?ref_noscript">comments powered by Disqus.</a>
</noscript>
</div>
</div>
</article>
</div>
</div>
</div>
</div>
</div>
</section>
</main>

<script src="assets/js/app.js"></script>
<script src="assets/js/theme.js"></script>

<footer class="footer pt-5 pb-5 text-center">
<div class="container">
    <div class="socials-media">
        <ul class="list-unstyled">
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-facebook"></i></a></li>
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-twitter"></i></a></li>
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-instagram"></i></a></li>
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-google-plus"></i></a></li>
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-behance"></i></a></li>
            <li class="d-inline-block ml-1 mr-1"><a href="#" class="text-dark"><i class="fa fa-dribbble"></i></a></li>
        </ul>
    </div>
</div>

<p>© <span class="credits font-weight-bold">
    <a target="_blank" class="text-dark" href="#">

```

```

    </div>
</footer>
</body>
</html>

```

3.3.3.2 The url path is as follows.

```
path('user/info', user.views.user_info, name='user_info'),
```

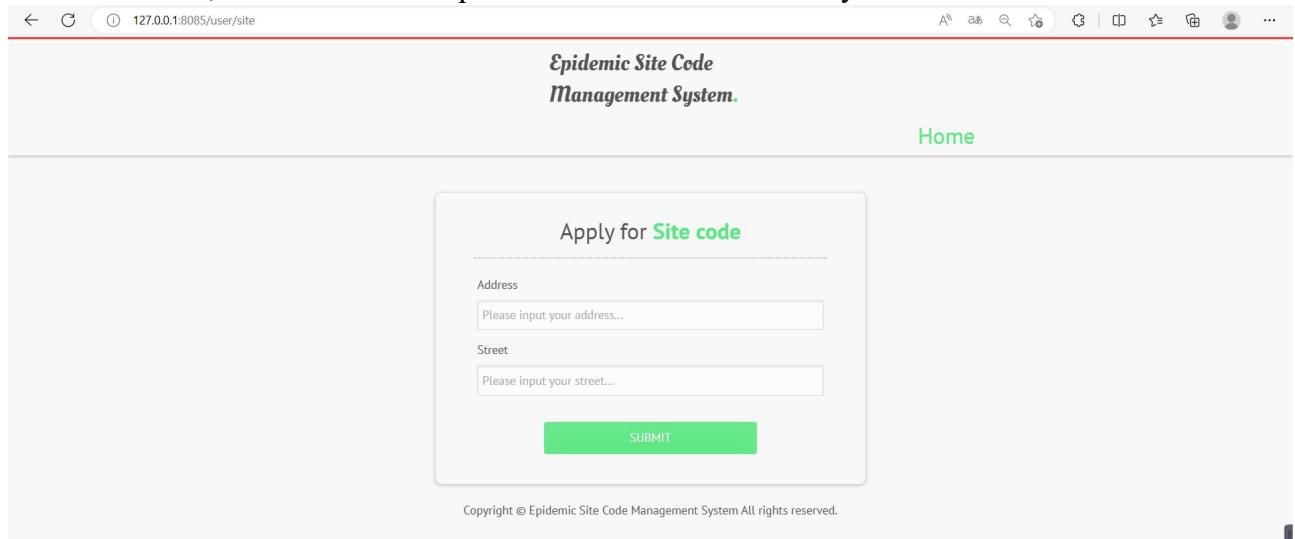
3.3.3.3 The view function is as follows

"""This is a user's personal information view function
that guides the user to the personal information display interface,
where you can view personal information and status."""

```
def user_info(req):
    if req.method == "GET":
        return render(req, "Info.html")
```

3.3.4 The Site Register Page

In this interface, users can submit specific addresses for review by the administrator.



3.3.4.1 Front End Display Interface Code

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

    <head>

        <meta charset="utf-8">
        <title>Site-Register</title>

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">

<!-- CSS -->
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=PT+Sans:400,700'>
<link rel='stylesheet' href='http://fonts.googleapis.com/css?family=Oleo+Script:400,700'>
<link rel="stylesheet" href="<% static 'bootstrap/css/bootstrap1.min.css' %}">
<link rel="stylesheet" href="<% static 'css/style1.css' %}">
<script src="<% static 'js/bootstrap.js' %}"></script>
<!-- HTML5 shim, for IE6-8 support of HTML5 elements -->
<!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>

<style>
.note{
    color: red;
}
.green{
    color: #5de784;
}
#bgc_c{
    background-color: #5de784;
}

</style>
<body>

<div class="header">
    <div class="container">
        <div class="row">
            <div class="logo span4">
                <h1><a href="">Epidemic Site Code Management System<span
class="green">.</span></a></h1>
            </div>
            <div class="links span8">
                <a class="home green" href="/index" rel="tooltip"
data-placement="bottom" data-original-title="Home">Home</a>
            </div>
        </div>
    </div>
</div>

<div class="register-container container">

```

```

<div class="row">
    <div class="register span6">
        <form method="post" novalidate>
            <h2>Apply for <span class="green"><strong>Site
code</strong></span></h2>
            {% csrf_token %}
            {% for field in form %}
                <label>{{ field.label }} </label>
                {{ field }}
                <span style="color: red">{{ field.errors.0 }}</span>
            {% endfor %}
            <div>
                <button type="submit" id="bgc_c">SUBMIT</button>
            </div>
        </form>
    </div>
</div>
<div align="center">Copyright © Epidemic Site Code Management System All rights
reserved.<a target="_blank" href="#"></a>
</div>

<!-- Javascript -->
<script src="{% static 'js/jquery1-1.8.2.min.js' %}"></script>
<script src="{% static '/bootstrap/js/bootstrap1.min.js' %}"></script>
<script src="{% static 'js/jquery1.backstretch.min.js' %}"></script>
<script src="{% static 'js/scripts1.js' %}"></script>

<div style="display:none"><script src='http://v7.cnzz.com/stat.php?id=155540&web_id=155540'
language='JavaScript' charset='gb2312'></script></div>
</body>
</html>

```

3.3.4.2 The url path is as follows.

path('user/site', user.views.user_site, name='site_register'),

3.3.4.3 The view function is as follows

"""This is the application site code view function,
which guides the user to the application site code interface.
After entering reasonable information, users can apply for registration site code."""

```

def user_site(req):
    if req.method == "GET":
        form = Site_A()
        return render(req, "user_site_register.html", {"form": form})

```

```

form = Site_A(data=req.POST)
if form.is_valid():
    address = req.POST.get("address")
    street = req.POST.get("street")
    iduser = User.objects.filter(iduser=req.session.get("Info")["id"]).first()
    Site_Ad.objects.create(iduser=iduser, address=address, street=street)
    return redirect("/index")
else:
    return render(req, "user_site_register.html", {"form": form})

```

The model form class is used for tables in the database so that they can be displayed on front-end pages with some selectors.

"""This is a place code registration class
that inherits from ModelForm and
can display an information box on the front-end page."""

```

class Site_A(forms.ModelForm):
    class Meta:
        model = Site_Ad
        fields = ["address", "street"]

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for name, field in self.fields.items():
            field.widget.attrs = {"placeholder": "Please input your " + name + "..."}

```

3.3.5 The My Site Page

This interface shows the audited site of the user, and shows the information and status of the current site.

Site ID	Address	Street	Status
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health
8	bjshdq	huayuanlijiedao	health
14	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment	health
15	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment2	health
16	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment3	health

3.3.5.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>My Site</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>  
    <script src="{% static 'js/bootstrap.min.js' %}"></script>  
    <style>  
        .navbar{  
            border-radius: 0;  
        }  
        #bgc_c{  
            color: #5de784;  
        }  
        #bgc_r{  
            color: #bd081c;  
        }  
    </style>  
</head>  
<body>  
<nav class="navbar navbar-default">  
    <div class="container-fluid">  
        <!-- Brand and toggle get grouped for better mobile display -->  
        <div class="navbar-header">  
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"  
data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  
                <span class="sr-only">Toggle navigation</span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
            </button>  
            <a class="navbar-brand" id="bgc_c" href="/index">Epidemic Site Code Management  
System</a>  
        </div>  
        <!-- Collect the nav links, forms, and other content for toggling -->  
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  
            <form class="navbar-form navbar-left">  
                <div class="form-group">  
                    <input type="text" class="form-control" placeholder="Search">  
                </div>
```

```

<button type="submit" class="btn btn-default">Submit</button>
</form>
<ul class="nav navbar-nav navbar-right">
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
aria-haspopup="true" aria-expanded="false">
            {{ request.session.Info.name }} <span class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="{% url 'user_info' %}">Personal</a></li>
                <li><a href="{% url 'logout' %}">Logout</a></li>
            </ul>
        </li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div>
    <div class="container">
        <div class="bs-example" data-example-id="panel-without-body-with-table">
            <div class="panel panel-default">
                <!-- Default panel contents -->
                <div style="display: flex" class="panel-heading">My Site
&nbsp;&nbsp;&nbsp;
                <p style="color: #5de784">total count: {{ count }}</p></div>

                <!-- Table -->
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>Site ID</th>
                            <th>Address</th>
                            <th>Street</th>
                            <th>Status</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for obj in search %}
                        <tr>
                            <th scope="row">{{ obj.idsite }}</th>
                            <td>{{ obj.address }}</td>
                            <td>{{ obj.street }}</td>
                            {% if obj.status == "health" %}
                                <td id="bgc_c">{{ obj.status }}</td>
                            {% else %}
                                <td id="bgc_r">{{ obj.status }}</td>
                            {% endif %}
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>
</div>

```

```

        {%- endif %}
    </tr>
    {%- endfor %}
</tbody>
</table>
</div>
</div>
</div>
</div>
</body>
</html>

```

3.3.5.2 The url path is as follows.

```
path('user/mysite', user.views.user_mysite, name='mysite'),
```

3.3.5.3 The view function is as follows

"""This is a view function of the places where users view personal applications.
This function allows users to view approved places."""

```
def user_mysite(req):
    iduser = req.session.get("Info")['id']
    search = Site.objects.filter(iduser=iduser).all()
    count = Site.objects.filter(iduser=iduser).aggregate(Count("iduser"))
    content = {'search': search, "count": count}
    return render(req, 'mysite.html', content)
```

3.3.6 The User-Site Page

In this interface, users can see which places they have visited and can query out some visiting information through fuzzy queries.

User Visit Record total count: 12			
Visit Record	Site ID	User ID	Visit time
1	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
2	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
3	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
4	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
5	Site object (13)	User object (24)	April 2, 2023, 10:33 p.m.
6	Site object (16)	User object (24)	April 2, 2023, 10:33 p.m.
7	Site object (12)	User object (24)	April 2, 2023, 10:33 p.m.
8	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.
9	Site object (13)	User object (24)	April 2, 2023, 10:34 p.m.
10	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.

Prev 1 2 Next page... Skip

3.3.6.1 Front End Display Interface Code

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User-Site</title>
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>
    <script src="{% static 'js/bootstrap.min.js' %}"></script>
    <style>
        .navbar {
            border-radius: 0;
        }
        #bgc_c {
            color: #5de784;
        }
        #bgc_r {
            color: #bd081c;
        }
    </style>
</head>
<body>
<nav class="navbar navbar-default">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
                   data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

```

```

<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" id="bgc_c" href="/index">Epidemic Site Code Management
System</a>
</div>
<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
<form class="navbar-form navbar-left" method="get">
<div class="form-group">
<input type="text" name="search" class="form-control" placeholder="Search">
<input type="text" name="s_time" class="form-control" placeholder="Search time...">
</div>

<button type="submit" class="btn btn-default"><span class="glyphicon glyphicon-search"
aria-hidden="true"></span></button>
</form>
<ul class="nav navbar-nav navbar-right">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
aria-expanded="false">

width="30">{{ request.session.Info.name }} <span
class="caret"></span></a>
<ul class="dropdown-menu">
<li><a href="{% url 'user_info' %}">Personal</a></li>
<li><a href="{% url 'logout' %}">Logout</a></li>
</ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div>
<div class="container">
<div class="bs-example" data-example-id="panel-without-body-with-table">
<div class="panel panel-default">
<!-- Default panel contents -->
<div style="display: flex" class="panel-heading">User Visit Record
&nbsp;&nbsp;&nbsp;
<p style="color: #5de784">total count: {{ count }}</p></div>

```

```

<!-- Table -->
<table class="table table-bordered">
    <thead>
        <tr>
            <th>Visit Record</th>
            <th>Site ID</th>
            <th>User ID</th>
            <th>Visit time</th>
        </tr>
    </thead>
    <tbody>
        {% for obj in queryset %}
            <tr>
                <th scope="row">{{ obj.idus }}</th>
                <th>{{ obj.idsite }}</th>
                <td>{{ obj.iduser }}</td>
                <td>{{ obj.e_time }}</td>
            </tr>
        {% endfor %}
    </tbody>
</table>
</div>
<div class="clearfix">
    <ul class="pagination" style="float: left">
        {{ page_str }}
    </ul>
</div>
</div>
</div></body>
</html>

```

3.3.6.2 The url path is as follows.

path('user/usersite', user.views.usersite),

3.3.6.3 The view function is as follows

"""This is a user view function for viewing personal code scanning records.
This function allows users to view their own code scanning records,
and users can search for corresponding data by searching criteria."""

```

def usersite(req):
    if req.method == "GET":
        userInfo = req.session.get("Info")["id"]
        value = req.GET.get("search")
        search_time = req.GET.get("s_time")

```

```

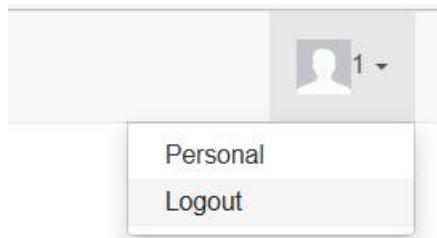
if value:
    queryset = models.Us.objects.filter(Q(idus__contains=value) |
                                         Q(idsite_idsite__contains=value) |
                                         Q(iduser_iduser__contains=value),
                                         iduser_iduser=userInfo
                                         )
elif search_time:
    queryset = models.Us.objects.filter(Q(e_time__year=search_time) |
                                         Q(e_time__day=search_time) |
                                         Q(e_time__month=search_time) |
                                         Q(e_time__hour=search_time) |
                                         Q(e_time__minute=search_time) |
                                         Q(e_time__second=search_time),
                                         iduser_iduser=userInfo
                                         )
else:
    queryset = models.Us.objects.filter(iduser_iduser=userInfo)
count = models.Us.objects.filter(iduser_iduser=userInfo).aggregate(Count('iduser'))
page_object = Pagination(req, queryset)
page_queryset = page_object.page_queryset
page_str = page_object.pages()
content = {"queryset": page_queryset, "page_str": page_str, "count": count,
           "page_start_index": page_start_index}
return render(req, 'usersite.html', content)

```

3.3.7 The Logout Function

The user logoff function, which is accomplished through session technology, is to clear the existing sessions. After clicking, the user can log out of the account directly.

3.3.7.1 Front End Style



3.3.7.2 The url path is as follows.

```
path('logout', user.views.logout, name='logout'),
```

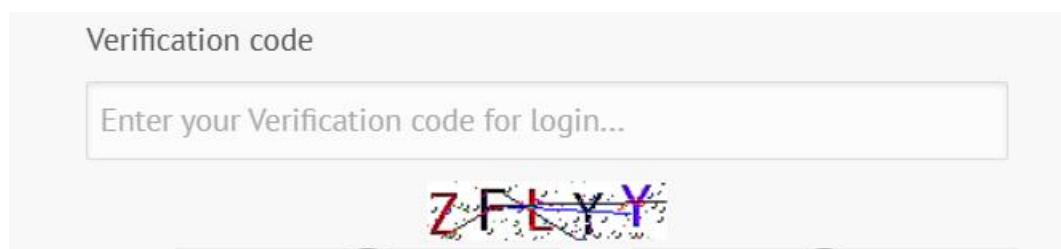
3.3.7.3 The view function is as follows

```
"""This is an account logout view function  
that allows users to log out of their accounts and switch to other accounts."""  
def logout(req):  
    req.session.clear()  
    return redirect('/index')
```

3.3.8 Verification Code Function

The verification code function can prevent some malicious attacks from brute force cracking the user's account and password, playing a defensive role.

3.3.8.1 Front End Style



3.3.8.2 The url path is as follows.

```
path('image/code', user.views.image),
```

3.3.8.3 The view function is as follows

```
"""This is the verification code generation function,  
which obtains a verification code through the verification code tool,  
and then displays the verification code to the front-end interface."""  
def image(req):  
    img, code_str = check_code()  
  
    req.session['image_code'] = code_str  
    stream = BytesIO()  
    img.save(stream, 'png')  
    return HttpResponse(stream.getvalue())
```

The following is the code generated by the verification code function:

```
import random
```

```
from PIL import Image, ImageFont, ImageDraw, ImageFilter
```

"""This is the login verification code tool.

When logging in, users need to enter a verification code

within one minute to log in,

otherwise they cannot log in even if the user has logged in with the correct account and password."""

```
def check_code(width=120, height=30, char_length=5, font_file='Monaco.ttf', font_size=28):
```

```
    code = []
```

```
    img = Image.new(mode='RGB', size=(width, height), color=(255, 255, 255))
```

```
    draw = ImageDraw.Draw(img, mode='RGB')
```

```
    def rndChar():
```

```
        """
```

Generate random letters

```
        """
```

```
        return chr(random.randint(65, 90))
```

```
    def rndColor():
```

```
        """
```

```
    return (random.randint(0, 255), random.randint(10, 255), random.randint(64, 255))
```

Write verification code content

```
    font = ImageFont.truetype(font_file, font_size)
```

```
    for i in range(char_length):
```

```
        char = rndChar()
```

```
        code.append(char)
```

```
        h = random.randint(0, 4)
```

```
        draw.text([i * width / char_length, h], char, font=font, fill=rndColor())
```

Fill interference points

```
    for i in range(40):
```

```
        draw.point([random.randint(0, width), random.randint(0, height)], fill=rndColor())
```

```

for i in range(40):

    draw.point([random.randint(0, width), random.randint(0, height)], fill=rndColor())

    x = random.randint(0, width)

    y = random.randint(0, height)

    draw.arc((x, y, x + 4, y + 4), 0, 90, fill=rndColor())

for i in range(5):

    x1 = random.randint(0, width)

    y1 = random.randint(0, height)

    x2 = random.randint(0, width)

    y2 = random.randint(0, height)

    draw.line((x1, y1, x2, y2), fill=rndColor())

img = img.filter(ImageFilter.EDGE_ENHANCE_MORE)

return img, ''.join(code)

```

3.4 QR code App

3.4.1 The Scan Page

This page pulls out all approved and scanned sites in the database and displays them on the front-end page.

Scan Site [total count: 16]					
Site ID	Address	Street	Status	Operation	
1	sdszcs	qqjd	health	Scan	
2	bjshdq	hyld	health	Scan	
3	wadadaw	dwadawd	health	Scan	
4	fniquqf	1243u890	health	Scan	
5	18ceng	123	health	Scan	
6	18ceng1	1234	health	Scan	
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	Scan	
8	bjshdq	huayuanlujiedao	health	Scan	
9	18ceng23	12345	health	Scan	
10	11111111111111	11111111111111	health	Scan	

Prev 1 2 Next page... Skip

```
1 •  SELECT * FROM epidemic_site_code_management_system.site;
```

idsite	iduser	address	street	status	idadmin	path
1	19	sdszcs	qqjd	health	1	static/qrcode/1.png
2	19	bjshdq	hyljd	health	1	static/qrcode/2.png
3	19	wadadaw	dwadawd	health	1	static/qrcode/3.png
4	19	fniuqf	1243u890	health	1	static/qrcode/4.png
5	19	18ceng	123	health	1	static/qrcode/5.png
6	19	18ceng1	1234	health	1	static/qrcode/6.png
7	24	Yongjia Post, Beijing University of Aeronautics a...	No. 37 Garden Road Street, Haidian District, Bei...	health	1	static/qrcode/7.png
8	24	bjshdq	huayuanlujiedao	health	1	static/qrcode/8.png
9	19	18ceng23	12345	health	1	static/qrcode/9.png
10	19	1111111111111111	1111111111111111	health	1	static/qrcode/10.png
11	43	Beijing.Haidian.Huayuan Road.No.37	BUAA New Building Room D535	health	1	static/qrcode/11.png
12	43	Beijing.Haidian.Huayuan Road.No.37	BUAA New Building Room D536	health	1	static/qrcode/12.png
13	43	Beijing.Haidian.Huayuan Road.No.37	BUAA New Building Room D533	health	1	static/qrcode/13.png
14	24	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment	health	1	static/qrcode/14.png
15	24	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment2	health	1	static/qrcode/15.png
16	24	Beijing.Haidian.Huayuan Road.No.37	New North Gate No15. Apartment3	health	1	static/qrcode/16.png

3.4.1.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Scan</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>  
    <script src="{% static 'js/bootstrap.min.js' %}"></script>  
    <style>  
        .navbar{  
            border-radius: 0;  
        }  
        #bgc_c{  
            color: #5de784;  
        }  
    </style>  
</head>  
<body>  
<nav class="navbar navbar-default">  
    <div class="container-fluid">  
        <!-- Brand and toggle get grouped for better mobile display -->  
        <div class="navbar-header">  
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"  
data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
```

```

Toggle navigation




```

</button>

Epidemic Site Code Management System

</div>

<!-- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<form class="navbar-form navbar-left" method="get">

<div class="form-group">

<input type="text" name="search" class="form-control" placeholder="Search">

</div>

<button type="submit" class="btn btn-default"></button>

</form>

<ul class="nav navbar-nav navbar-right">

<li class="dropdown">

{{ request.session.Info.name }}

<ul class="dropdown-menu">

Personal

Logout

</div><!-- /.navbar-collapse -->

</div><!-- /.container-fluid -->

</nav>

<div>

<div class="container">

<div class="bs-example" data-example-id="panel-without-body-with-table">

<div class="panel panel-default">

<!-- Default panel contents -->

<div style="display: flex" class="panel-heading">Scan Site <p style="color: #5de784">total count: {{ count }}</p></div>

<!-- Table -->

<table class="table table-bordered">

<thead>

```

<tr>
    <th>Site ID</th>
    <th>Address</th>
    <th>Street</th>
    <th>Operation</th>
</tr>
</thead>
<tbody>
    {% for obj in queryset %}
<tr>
    <th scope="row">{{ obj.idsite }}</th>
    <td>{{ obj.address }}</td>
    <td>{{ obj.street }}</td>
    <td>
        <a class="btn btn-primary btn-xs"
        href="{{ obj.idsite }}/scansingle">Scan</a>
    </td>
</tr>
    {% endfor %}
</tbody>
</table>
</div>
<div class="clearfix">
    <ul class="pagination" style="float: left">
        {{ page_str }}
    </ul>
</div>
</div>
</div>
</body>
</html>

```

3.4.1.2 The url path is as follows.

path('QR/scan', QRcode.views.qrscan),

3.4.1.3 The view function is as follows

"""This is a function for displaying all QR codes in the database.
This function allows the user to display all healthy QR codes in the database,
and then scan a single QR code."""

```

def qrscan(req):
    if req.method == "GET":
        value_dict = {"status": "health"}
        value = req.GET.get("search")

```

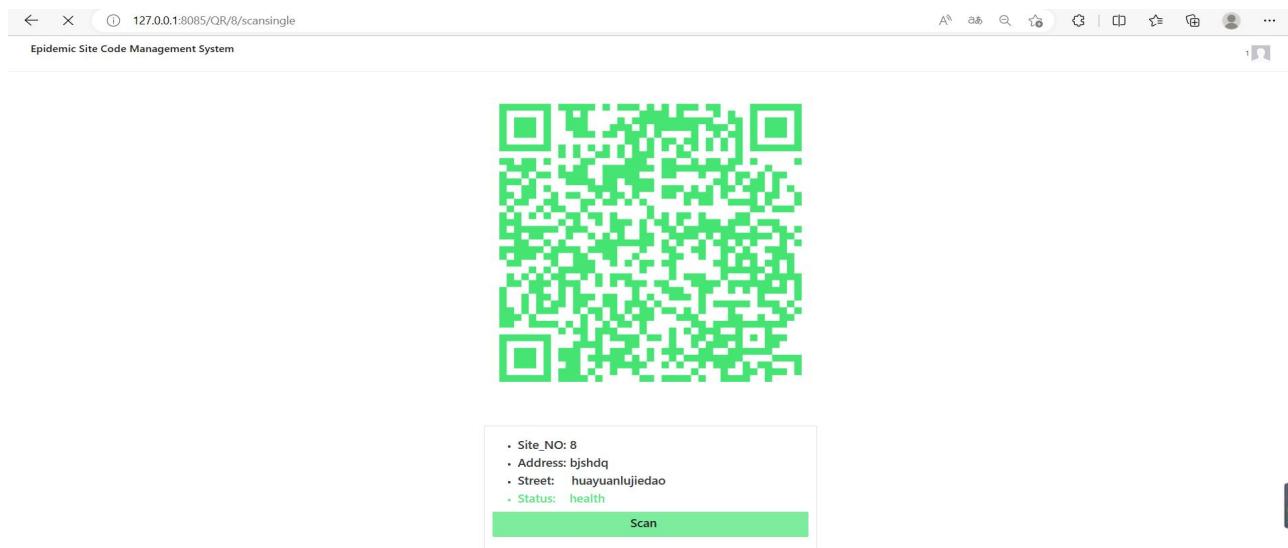
```

if value:
    value_dict["idsite"] = value
    queryset = Site.objects.filter(**value_dict)
    count = Site.objects.all().aggregate(Count("idsite"))
    page_object = Pagination(req, queryset)
    page_queryset = page_object.page_queryset
    page_str = page_object.pages()
    content = {"queryset": page_queryset, "page_str": page_str, "count": count}
    return render(req, "scan.html", content)

```

3.4.2 The Scan Function

This interface shows the specific information of the QR code to be scanned, after clicking on the scan, a user visit information will be entered into the database.



After the scan is completed, a successful interface will be displayed and returned to the index page in three seconds.



User Visit Record			
Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
2	Site object (4)	User object (19)	March 12, 2023, 12:04 p.m.
3	Site object (1)	User object (19)	March 12, 2023, 8:10 p.m.
8	Site object (8)	User object (19)	March 14, 2023, 10:27 p.m.

3.4.2.1 Front End Display Interface Code

A) This is the QR code display interface.

```
{% load static %}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Scan Single</title>
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>
    <script src="{% static 'js/bootstrap.min.js' %}"></script>
    <link rel="stylesheet" href="{% static 'css/app.css' %}">
    <link rel="stylesheet" href="{% static 'css/theme.css' %}">
    <style>
        .navbar{
            border-radius: 0;
        }
        .position{
            display: flex;
            text-align: center;
            vertical-align: middle;
        }
        .border{
            margin: 4%;
        }
        .bgc_c{
            color: #5de784;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
        <a class="navbar-brand font-weight-bolder mr-3" href="/index"><img src=""><h3>Epidemic
        Site Code Management System</h3></a>
        <button class="navbar-light navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbarsDefault" aria-controls="navbarsDefault" aria-expanded="false"
        aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarsDefault">
            <ul class="navbar-nav ml-auto align-items-center">
                <li>{{ request.session.Info.name }}<a href="{% url 'user_info' %}">
```

```

        </a>
    </li>
</ul>
</div>
</nav>

<div>
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-7">
            <article class="card ">
                
                <div class="card-body border">
                    <h1 class="card-title display-4"></h1>

                    <ul>
                        <li><h2>Site_NO: {{ site.idsite }} </h2></li>
                        <li><h2>Address: {{ site.address }}</h2></li>
                        <li><h2>Street:   &nbsp;&nbsp;&nbsp;&nbsp;{{ site.street }}</h2> </li>
                        <li class="bgc_c"><h2>Status: &nbsp;&nbsp;&nbsp;{{ site.status }} </h2>
                    </ul>
                </div>
            </article>
        </div>
        <a class="btn btn-lightgreen btn-xs position" href="decode">
            <h2>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Scan</h2></a>
        </div>
        <div id="comments" class="mt-4">
            <div id="disqus_thread">
            </div>
            <script type="text/javascript">
                var disqus_shortname = 'demoweb';
                var disqus_developer = 0;
                (function() {
                    var dsq = document.createElement('script'); dsq.type =
                    'text/javascript'; dsq.async = true;
                    dsq.src = window.location.protocol + '//' + disqus_shortname
                    + '.disqus.com/embed.js';
                    (document.getElementsByTagName('head')[0] ||
                    document.getElementsByTagName('body')[0]).appendChild(dsq);
                })();
            </script>
        </div>
    </div>
</div>

```

```

<noscript>
    Please enable JavaScript to view the <a href="http://disqus.com/?ref_noscript">comments powered by Disqus.</a>
</noscript>
</div>
</div>
</article>
</div>
</div>
</div>
</div>
</body>
</html>

```

B) This is the QR code scan completion interface.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Decode</title>
</head>
<body>
<p> Complete scan! <span id="jump">3</span>seconds left to jump to the home page</p>
<script>
    onload=function () {
        setInterval(go,1000)
    };
    var x=2;
    function go() {

        if (x>=0){
            document.getElementById("jump").innerText=x;
        }else {
            location.href="/index";
        }
        x--;
    }
</script>
</body>
</html>

```

3.4.2.2 The url path is as follows.

```

path('QR/<int:sid>/scansingle', QRcode.views.scansingle),
path('QR/<int:sid>/decode', QRcode.views.decodeqr),

```

3.4.2.3 The view function is as follows

"""This is a single QR code display view function
that allows you to display all information
about a specific location, and then scan it."""

```
def scansingle(req, sid):  
    if req.method == "GET":  
        site = Site.objects.filter(idsite=sid).first()  
        return render(req, "scan_single.html", {"site": site})
```

"""This is a 2D code decoding view function that allows
the user to simulate the scanning of 2D codes in actual scenes,
decode the information needed, and then enter the scanning information into the database."""

```
def decodeqr(req, sid):  
    if req.method == "GET":  
        site = Site.objects.filter(idsite=sid).first()  
        Info = str(decode(Image.open(site.path))[0][0])  
        Info = ast.literal_eval(Info.split("\n")[1])  
        now_time = datetime.datetime.now()  
        idsite = Site.objects.filter(idsite=Info["idsite"]).first()  
        iduser = User.objects.filter(iduser=req.session.get("Info")["id"]).first()  
        Us.objects.create(idsite=idsite, iduser=iduser, e_time=now_time)  
        return render(req, "decode.html")
```

3.4.3 QR Code Generation Function

For the user's application request, the administrator can generate a QR code after verifying it.

Site verify				
Site ID	User ID	Address	Street	Operation
1	User object (19)	sdszcs	fsnl10086	<button>verify</button>
5	User object (19)	1111111111111111	1111111111111111	<button>verify</button>

After passing the application, there will be a prompt of success and return to the index page in three seconds.

Complete generate!3seconds left to jump to the home page

3.4.3.1 Front End Display Interface Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Generate QRcode</title>
</head>
<body>
<p>Complete generate!<span id="jump">3</span>seconds left to jump to the home page</p>
<script>
    onload=function () {
        setInterval(go,1000)
    };
    var x=2;
    function go() {
        if (x>=0){
            document.getElementById("jump").innerText=x;
        }else {
            location.href="/index";
        }
        x--;
    }
</script>
</body>
</html>
```

3.4.3.2 The url path is as follows.

```
path('QR/generate', QRcode.views.generate),
```

3.4.3.3 The view function is as follows

```
"""This is a two-dimensional code generation view function  
that allows you to generate a two-dimensional code for  
the address of the site that has been applied for and store it in the system."""
```

```
def generate(req):
    if req.method == "GET":
        sql = Site.objects.filter(path=None).first()

    if sql is not None:
        sql = model_to_dict(sql)
        qr = qr_generate(sql)
```

```

img = qr.make_image(fill_color=(70, 228, 115), back_color="white")
img.save(path + str(sql['idsite']) + ".png")
Site.objects.filter(idsite=str(sql['idsite'])).update(path=path + str(sql['idsite']) + ".png")
return render(req, "gqr.html")

else:
    return HttpResponse("The instance was not found")
return render(req, "gqr.html")

```

3.5 Supervise App

3.5.1 The User-Status Page

This interface shows the user's basic information and status. Click Modify to enter the user status modification interface.

User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<button>modify</button>
2	1111111111	zsr	abnormal	<button>modify</button>
3	2222222222	123	health	<button>modify</button>
14	11111	name		<button>modify</button>
16	321321	123		<button>modify</button>
17	0	1	health	<button>modify</button>
18	3213213	123	health	<button>modify</button>
19	1111111	1	health	<button>modify</button>
20	8008208820	An Moli	health	<button>modify</button>
22	1234567890	z1	health	<button>modify</button>

Prev 1 2 3 Next page... Skip

3.5.1.1 Front End Display Interface Code

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User Status</title>
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>
    <script src="{% static 'js/bootstrap.min.js' %}"></script>
    <style>
        .navbar{

```

```

        border-radius: 0;
    }
    #bgc_c{
        color: #5de784;
    }
    #bgc_r{
        color: #bd081c;
    }
    #bgc_y{
        color: #e0b849;
    }

```

</style>

</head>

<body>

<nav class="navbar navbar-default">

<div class="container-fluid">

<!-- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

Toggle navigation

</button>

Epidemic Site Code Management System

</div>

<!-- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<form class="navbar-form navbar-left" method="get">

<div class="form-group">

<input type="text" name="search" class="form-control" placeholder="Search">

</div>

<button type="submit" class="btn btn-default"></button>

</form>

<ul class="nav navbar-nav navbar-right">

<li class="dropdown">

{{ request.session.Info.name }}

<ul class="dropdown-menu">

```

<li><a href="{% url 'user_info' %}">Personal</a></li>
<li><a href="{% url 'logout' %}">Logout</a></li>
</ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div>
<div class="container">
<div class="bs-example" data-example-id="panel-without-body-with-table">
<div class="panel panel-default">
<!-- Default panel contents -->
<div style="display: flex" class="panel-heading">User <p style="color: #5de784">total count: {{ count }}</p></div>

<!-- Table -->
<table class="table table-bordered">
<thead>
<tr>
<th>User ID</th>
<th>Phone</th>
<th>Name</th>
<th>Status</th>
<th>Modify</th>
</tr>
</thead>
<tbody>
{%
  for obj in user
%}
<tr>
<th scope="row">{{ obj.iduser }}</th>
<td>{{ obj.phone }}</td>
<td>{{ obj.name }}</td>
{%
  if obj.status == "health"
%}
<td id="bgc_c">{{ obj.status }}</td>
{%
  elif obj.status == "unknown"
%}
<td id="bgc_y">{{ obj.status }}</td>
{%
  else
%}
<td id="bgc_r">{{ obj.status }}</td>
{%
  endif
%}
<td>
<a class="btn btn-primary btn-xs"
href="{{ obj.iduser }}/user_modify">modify</a>
</td>
</tr>

```

```

        {%- endfor %}
        </tbody>
    </table>
</div>
<div class="clearfix">
    <ul class="pagination" style="float: left">
        {{ page_str }}
    </ul>
</div>
</div>
</div>
</body>
</html>

```

3.5.1.2 The url path is as follows.

path('SP/user_status', supervise.views.user_status),

3.5.1.3 The view function is as follows

```

"""This is a function for administrators to view user information.
This function allows you to list the information of all users and search
for users based on specific criteria."""
def user_status(req):
    if req.method == "GET":
        value = req.GET.get("search")
        if value:
            """Q(iduser=value) | Q(phone=value) | Q(name=value) | Q(status=value)"""
            queryset = models.User.objects.filter(Q(iduser__contains=value) |
                                                    Q(phone__contains=value) |
                                                    Q(name__contains=value) |
                                                    Q(status__contains=value))
        else:
            queryset = models.User.objects.filter()
    page_object = Pagination(req, queryset)
    page_queryset = page_object.page_queryset
    count = models.User.objects.all().aggregate(Count('iduser'))
    page_str = page_object.pages()
    content = {"user": page_queryset, "page_str": page_str, "count": count}
    return render(req, "user_status.html", content)

```

3.5.2 The User-Status Modify Page

This page displays a user's information and status separately. The administrator can modify the user's status according to the user's trip.

User Information

User ID
1

Phone
1111

Name
12

Status
unknown

health

abnormal

unknown

Modify

3.5.2.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>User Modify</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>  
    <script src="{% static 'js/bootstrap.min.js' %}"></script>  
    <style>  
        .navbar {  
            border-radius: 0;  
        }  
  
        #bgc_c {  
            color: #5de784;  
        }  
  
        #bgc_r {  
            color: #bd081c;  
        }  
  
        .bor {  
            margin: 10%;  
        }  
    </style>
```

```

        }
.but{
    margin-top: 5%;
    display: flex;
    width: 100%;
}
.modi{
    width: 100%;
    height: 5%;
}

```

</style>

</head>

<body>

<nav class="navbar navbar-default">

<div class="container-fluid">

<!-- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

Toggle navigation

</button>

Epidemic Site Code Management System

</div>

<!-- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-right">

<li class="dropdown">

{<% request.session.Info.name %}

<ul class="dropdown-menu">

<a href="<% url 'user_info' %>">Personal

<a href="<% url 'logout' %>">Logout

</div><!-- /.navbar-collapse -->

```

</div><!-- /.container-fluid -->
</nav>

<div class="bor table-bordered">
    <h2>User Information</h2>

    <form method="post">
        {% csrf_token %}
        <label>User ID</label>
        <div class="input-group">
            <span class="input-group-addon" id="basic-addon1"></span>
            <input type="text" class="form-control" aria-describedby="basic-addon1"
value="{{ userInfo.iduser }}"
                readonly>
            <span class="input-group-addon"></span>
        </div>

        <label>Phone</label>
        <div class="input-group">
            <span class="input-group-addon"></span>
            <input type="text" class="form-control" aria-describedby="basic-addon2"
value="{{ userInfo.phone }}"
                readonly>
            <span class="input-group-addon" id="basic-addon2"></span>
        </div>

        <label>Name</label>
        <div class="input-group">
            <span class="input-group-addon"></span>
            <input type="text" class="form-control" aria-label="Amount (to the nearest dollar)"
value="{{ userInfo.name }}" readonly>
            <span class="input-group-addon"></span>
        </div>

        <label>Status</label>
        <div class="input-group">
            <span class="input-group-addon"></span>
            <select class="form-control" name="status">
                <option value="{{ userInfo.status }}" selected
hidden>{{ userInfo.status }}</option>
                <option id="bgc_c" value="health">health</option>
                <option id="bgc_r" value="abnormal">abnormal</option>
                <option value="unknown" style="color: #e0b849">unknown</option>
            </select>
            <span class="input-group-addon"></span>
        </div>

```

```

<div class="btn-group but" role="group">
    <button type="submit" class="btn btn-default modi"><h2>Modify</h2></button>
</div>
</form>
</div>
</body>

```

3.5.2.2 The url path is as follows.

path('SP/<int:iduser>/user_modify', supervise.views.user_modify),

3.5.2.3 The view function is as follows

"""This is a function for the administrator to modify the user information view.
This function allows the user to receive modifications to
the user status information on the front-end page,
and save the modified data to the database."""

```

def user_modify(req, iduser):
    userInfo = models.User.objects.filter(iduser=iduser).first()
    if req.method == "GET":
        return render(req, "user_modify.html", {"userInfo": userInfo})
    status = req.POST.get("status")
    user = models.User.objects.filter(iduser=iduser).update(status=status)
    return redirect("/SP/user_status")

```

3.5.3 The Site-Status Page

This interface shows the site's basic information and status. Click Modify to enter the site status modification interface.

Site [total count: 16]						
Site ID	User ID	Admin ID	Address	Street	Status	Modify
1	User object (19)	Admin object (1)	sdszcs	qqjd	health	<button>modify</button>
2	User object (19)	Admin object (1)	bjshdq	hyjd	health	<button>modify</button>
3	User object (19)	Admin object (1)	wadadaw	dwadawd	health	<button>modify</button>
4	User object (19)	Admin object (1)	fniquqf	1243u890	health	<button>modify</button>
5	User object (19)	Admin object (1)	18ceng	123	health	<button>modify</button>
6	User object (19)	Admin object (1)	18ceng1	1234	health	<button>modify</button>
7	User object (24)	Admin object (1)	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	<button>modify</button>
8	User object (24)	Admin object (1)	bjshdq	huayuanlijedao	health	<button>modify</button>
9	User object (19)	Admin object (1)	18ceng23	12345	health	<button>modify</button>
10	User object (19)	Admin object (1)	1111111111111111	1111111111111111	health	<button>modify</button>

3.5.3.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Site Status</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>  
    <script src="{% static 'js/bootstrap.min.js' %}"></script>  
    <style>  
        .navbar{  
            border-radius: 0;  
        }  
        #bgc_c{  
            color: #5de784;  
        }  
        #bgc_r{  
            color: #bd081c;  
        }  
        #bgc_y{  
            color: #e0b849;  
        }  
    </style>  
</head>  
<body>  
<nav class="navbar navbar-default">  
    <div class="container-fluid">  
        <!-- Brand and toggle get grouped for better mobile display -->  
        <div class="navbar-header">  
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"  
data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  
                <span class="sr-only">Toggle navigation</span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
            </button>  
            <a class="navbar-brand" id="bgc_c" href="/index2">Epidemic Site Code Management  
System</a>  
        </div>  
        <!-- Collect the nav links, forms, and other content for toggling -->  
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  
            <form class="navbar-form navbar-left" method="get">
```

```

<div class="form-group">
    <input type="text" name="search" class="form-control" placeholder="Search">
</div>
<button type="submit" class="btn btn-default"><span class="glyphicon glyphicon-search" aria-hidden="true"></span></button>
</form>
<ul class="nav navbar-nav navbar-right">
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
            {{ request.session.Info.name }} <span class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="{% url 'user_info' %}">Personal</a></li>
                <li><a href="{% url 'logout' %}">Logout</a></li>
            </ul>
        </li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div>
    <div class="container">
        <div class="bs-example" data-example-id="panel-without-body-with-table">
            <div class="panel panel-default">
                <!-- Default panel contents -->
                <div style="display: flex" class="panel-heading">Site <p style="color: #5de784">total count: {{ count }}</p></div>

                <!-- Table -->
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>Site ID</th>
                            <th>User ID</th>
                            <th>Admin ID</th>
                            <th>Address</th>
                            <th>Street</th>
                            <th>Status</th>
                            <th>Modify</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for obj in site %}
                        <tr>

```

```

<th scope="row">{{ obj.idsite }}</th>
<td>{{ obj.iduser }}</td>
<th>{{ obj.idadmin }}</th>
<td>{{ obj.address }}</td>
<td>{{ obj.street }}</td>
{%
    if obj.status == "health" %
        <td id="bgc_c">{{ obj.status }}</td>
    {% elif obj.status == "unknown" %}
        <td id="bgc_y">{{ obj.status }}</td>
    {% else %}
        <td id="bgc_r">{{ obj.status }}</td>
    {% endif %}
<td>
    <a class="btn btn-primary btn-xs"
    href="{{ obj.idsite }}/site_modify">modify</a>
</td>
</tr>
{%
    endfor %
}
</tbody>
</table>
</div>
<div class="clearfix">
    <ul class="pagination" style="float: left">
        {{ page_str }}
    </ul>
</div>
</div>
</div>
</body>
</html>

```

3.5.3.2 The url path is as follows.

path('SP/site_status', supervise.views.site_status),

3.5.3.3 The view function is as follows

"""This is a function for administrators to view user's sites information.
This function allows you to list the information of all sites and search
for sites based on specific criteria."""

```

def site_status(req):
    if req.method == "GET":
        value = req.GET.get("search")
        if value:
            queryset = models.Site.objects.filter(Q(iduser__iduser__contains=value) |

```

```

        Q(idsite__contains=value) |
        Q(idadmin__idadmin__contains=value)

    | 
        Q(address__contains=value) |
        Q(street__contains=value) |
        Q(status__contains=value))

else:
    queryset = models.Site.objects.filter()
count = models.Site.objects.all().aggregate(Count('idsite'))
page_object = Pagination(req, queryset)
page_queryset = page_object.page_queryset
page_str = page_object.pages()
content = {"site": page_queryset, "page_str": page_str, "count": count}
return render(req, "site_status.html", content)

```

3.5.4 The Site-Status Modify Page

This page displays a site's information and status separately. The administrator can modify the site's status according to the user's trip.

10	User object (19)	Admin object (1)	11111111111111	11111111111111	abnormal	modify
----	------------------	------------------	----------------	----------------	----------	------------------------

3.5.4.1 Front End Display Interface Code

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Site Modify</title>

```

```

<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
<script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<style>
    .navbar {
        border-radius: 0;
    }

    #bgc_c {
        color: #5de784;
    }

    #bgc_r {
        color: #bd081c;
    }

    .bor {
        margin: 10%;
    }

    .but{
        margin-top: 5%;
        display: flex;
        width: 100%;
    }

    .modi{
        width: 100%;
        height: 5%;
    }
</style>
</head>
<body>
<nav class="navbar navbar-default">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
                   data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" id="bgc_c" href="/index">Epidemic Site Code Management
System</a>
        </div>
    </div>

```

```

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

    <ul class="nav navbar-nav navbar-right">
        <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
aria-expanded="false">
                {{ request.session.Info.name }} <span
class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="<% url 'user_info' %>">Personal</a></li>
                <li><a href="<% url 'logout' %>">Logout</a></li>
            </ul>
        </li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div class="bor table-bordered">
    <h2>Site Information</h2>

    <form method="post">
        {% csrf_token %}
        <label>Site ID</label>
        <div class="input-group">
            <span class="input-group-addon" id="basic-addon1"></span>
            <input type="text" class="form-control" aria-describedby="basic-addon1"
value="{{ siteInfo.idsite }}"
                readonly>
            <span class="input-group-addon"></span>
        </div>

        <label>User ID</label>
        <div class="input-group">
            <span class="input-group-addon"></span>
            <input type="text" class="form-control" aria-describedby="basic-addon2"
value="{{ siteInfo.iduser }}"
                readonly>
            <span class="input-group-addon" id="basic-addon2"></span>
        </div>

        <label>Address</label>
        <div class="input-group">

```

```

<span class="input-group-addon"></span>
<input type="text" class="form-control" aria-label="Amount (to the nearest dollar)"
       value="{{ siteInfo.address }}" readonly>
<span class="input-group-addon"></span>
</div>

<label>Street</label>
<div class="input-group">
    <span class="input-group-addon"></span>
    <input type="text" class="form-control" aria-label="Amount (to the nearest dollar)"
           value="{{ siteInfo.street }}" readonly>
    <span class="input-group-addon"></span>
</div>

<label>Admin ID</label>
<div class="input-group">
    <span class="input-group-addon"></span>
    <input type="text" class="form-control" aria-label="Amount (to the nearest dollar)"
           value="{{ siteInfo.idadmin }}" readonly>
    <span class="input-group-addon"></span>
</div>

<label>Status</label>
<div class="input-group">
    <span class="input-group-addon"></span>
    <select class="form-control" name="status">
        <option value="{{ siteInfo.status }}" selected
hidden>{{ siteInfo.status }}</option>
        <option id="bgc_c" value="health">health</option>
        <option id="bgc_r" value="abnormal">abnormal</option>
        <option value="unknown" style="color: #e0b849">unknown</option>
    </select>
    <span class="input-group-addon"></span>
</div>

<div class="btn-group but" role="group">
    <button type="submit" class="btn btn-default modi"><h2>Modify</h2></button>
</div>
</form>

</div>
</body>

```

3.5.4.2 The url path is as follows.

```
path('SP/<int:idsite>/site_modify', supervise.views.site_modify),
```

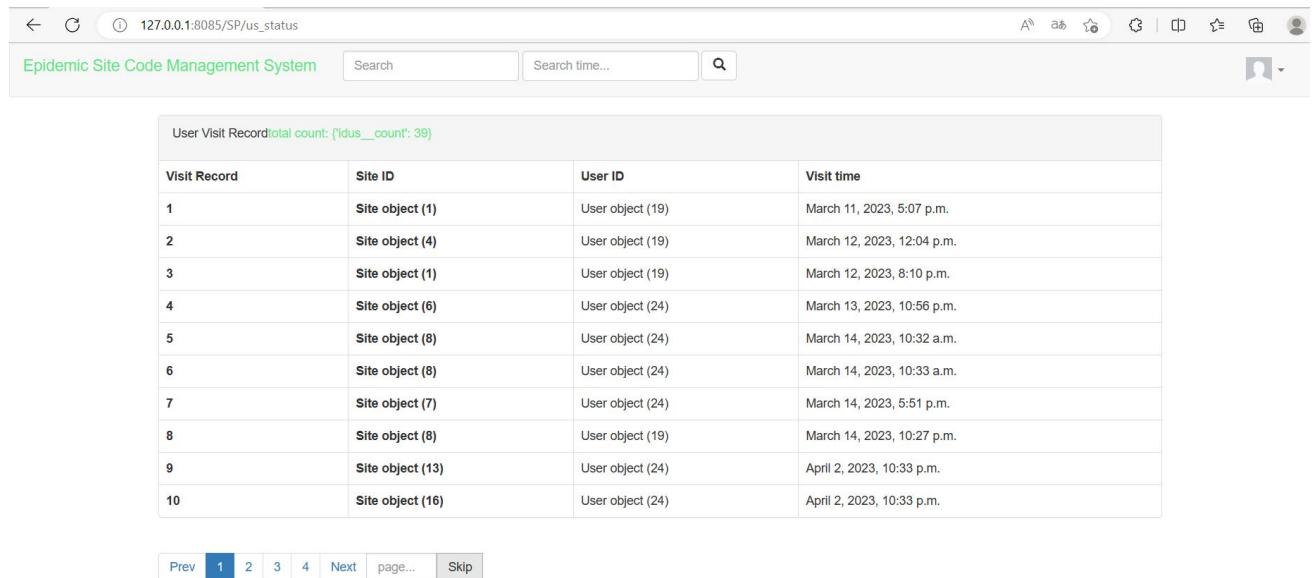
3.5.4.3 The view function is as follows

"""This is a function for the administrator to modify the site information view.
This function allows the user to receive modifications to
the site status information on the front-end page,
and save the modified data to the database."""

```
def site_modify(req, idsite):  
    siteInfo = models.Site.objects.filter(idsite=idsite).first()  
    if req.method == "GET":  
        return render(req, "site_modify.html", {"siteInfo": siteInfo})  
    status = req.POST.get("status")  
    site = models.Site.objects.filter(idsite=idsite).update(status=status)  
    return redirect("/SP/site_status")
```

5.5 The User Visit Page

In this interface, admin can see which places they have visited and can query out some visiting information through fuzzy queries.



The screenshot shows a web browser window with the URL 127.0.0.1:8085/SP/us_status. The page title is 'Epidemic Site Code Management System'. There are two search input fields: 'Search' and 'Search time...', each with a magnifying glass icon. Below the search fields is a table titled 'User Visit Record [total count: 39]'. The table has four columns: 'Visit Record', 'Site ID', 'User ID', and 'Visit time'. The data in the table is as follows:

Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
2	Site object (4)	User object (19)	March 12, 2023, 12:04 p.m.
3	Site object (1)	User object (19)	March 12, 2023, 8:10 p.m.
4	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
5	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
6	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
8	Site object (8)	User object (19)	March 14, 2023, 10:27 p.m.
9	Site object (13)	User object (24)	April 2, 2023, 10:33 p.m.
10	Site object (16)	User object (24)	April 2, 2023, 10:33 p.m.

Below the table are navigation buttons: 'Prev', '1' (highlighted), '2', '3', '4', 'Next', 'page...', and 'Skip'.

3.5.5.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>User-Site</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
```

```

<script src="<% static 'js/jquery-3.6.4.min.js' %>"></script>
<script src="<% static 'js/bootstrap.min.js' %>"></script>
<style>
    .navbar {
        border-radius: 0;
    }

    #bgc_c {
        color: #5de784;
    }

    #bgc_r {
        color: #bd081c;
    }
</style>
</head>
<body>
<nav class="navbar navbar-default">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
                   data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" id="bgc_c" href="/index2">Epidemic Site Code
Management System</a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <form class="navbar-form navbar-left" method="get">
                <div class="form-group">
                    <input type="text" name="search" class="form-control"
placeholder="Search">
                </div>
                <button type="submit" class="btn btn-default"><span class="glyphicon glyphicon-search"
aria-hidden="true"></span></button>
            </form>
            <ul class="nav navbar-nav navbar-right">

```

```

<li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
        role="button" aria-haspopup="true"
        aria-expanded="false">
        {{ request.session.Info.name }} <span
            class="caret"></span></a>
    <ul class="dropdown-menu">
        <li><a href="{% url 'user_info' %}">Personal</a></li>
        <li><a href="{% url 'logout' %}">Logout</a></li>
    </ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div>
    <div class="container">
        <div class="bs-example" data-example-id="panel-without-body-with-table">
            <div class="panel panel-default">
                <!-- Default panel contents -->
                <div style="display: flex" class="panel-heading">User Visit Record
                <p style="color: #5de784">total count: {{ count }}</p></div>

                <!-- Table -->
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>Visit Record</th>
                            <th>Site ID</th>
                            <th>User ID</th>
                            <th>Visit time</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for obj in queryset %}
                        <tr>
                            <th scope="row">{{ obj.idus }}</th>
                            <th>{{ obj.idsite }}</th>
                            <td>{{ obj.iduser }}</td>
                            <td>{{ obj.e_time }}</td>
                        </tr>
                        {% endfor %}
                    </tbody>
                </table>

```

```

</div>
<div class="clearfix">
    <ul class="pagination" style="float: left">
        {{ page_str }}
    </ul>
</div>
</div>
</div>
</body>
</html>

```

3.5.5.2 The url path is as follows.

```
path('SP/us_status', supervise.views.us_status),
```

3.5.5.3 The view function is as follows

```

"""This is a view function for administrators to view all user scanning records.
Administrators can also search based on specific criteria,
and the searched content will be returned to this page."""
def us_status(req):
    if req.method == "GET":
        value = req.GET.get("search")
        search_time = req.GET.get("s_time")
        if value:
            queryset = models.Us.objects.filter(Q(idus__contains=value) |
                                                 Q(idsite_idsite__contains=value) |
                                                 Q(iduser_iduser__contains=value) |
                                                 Q(e_time__contains=value)
                                                 )
        elif search_time:
            queryset = models.Us.objects.filter(Q(e_time__year=search_time) |
                                                Q(e_time__day=search_time) |
                                                Q(e_time__month=search_time) |
                                                Q(e_time__hour=search_time) |
                                                Q(e_time__minute=search_time) |
                                                Q(e_time__second=search_time)
                                                )
        else:
            queryset = models.Us.objects.filter()
    count = models.Us.objects.all().aggregate(Count('idus'))
    page_object = Pagination(req, queryset)
    page_queryset = page_object.page_queryset
    page_str = page_object.pages()
    content = {"queryset": page_queryset, "page_str": page_str, "count": count}

```

```
return render(req, "us_status.html", content)
```

3.5.6 The Application Verification Page

The administrator can view the user's site code application and generate a QR code after review.



Site verify				
Site ID	User ID	Address	Street	Operation
1	User object (19)	sdszcs	fsnl10086	<button>verify</button>
5	User object (19)	11111111111111	11111111111111	<button>verify</button>

3.5.6.1 Front End Display Interface Code

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Site Verify</title>  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
    <script src="{% static 'js/jquery-3.6.4.min.js' %}"></script>  
    <script src="{% static 'js/bootstrap.min.js' %}"></script>  
    <style>  
        .navbar{  
            border-radius: 0;  
        }  
        #bgc_c{  
            color: #5de784;  
        }  
    </style>  
</head>  
<body>  
<nav class="navbar navbar-default">  
    <div class="container-fluid">  
        <!-- Brand and toggle get grouped for better mobile display -->  
        <div class="navbar-header">  
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"  
            data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  
                <span class="sr-only">Toggle navigation</span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
                <span class="icon-bar"></span>  
            </button>  
        </div>
```

```

<a class="navbar-brand" id="bgc_c" href="/index">Epidemic Site Code Management
System</a>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<form class="navbar-form navbar-left" method="get">
  <div class="form-group">
    <input type="text" name="search" class="form-control" placeholder="Search">
  </div>
  <button type="submit" class="btn btn-default"><span class="glyphicon glyphicon-search" aria-hidden="true"></span></button>
</form>
<ul class="nav navbar-nav navbar-right">
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
      {{ request.session.Info.name }} <span class="caret"></span></a>
      <ul class="dropdown-menu">
        <li><a href="{% url 'user_info' %}">Personal</a></li>
        <li><a href="{% url 'logout' %}">Logout</a></li>
      </ul>
    </li>
  </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div>
  <div class="container">
    <div class="bs-example" data-example-id="panel-without-body-with-table">
      <div class="panel panel-default">
        <!-- Default panel contents -->
        <div class="panel-heading">Site verify</div>

        <!-- Table -->
        <table class="table table-bordered">
          <thead>
            <tr>
              <th>Site ID</th>
              <th>User ID</th>
              <th>Address</th>
              <th>Street</th>
              <th>Operation</th>
            </tr>
          </thead>
        </table>
    </div>
  </div>
</div>

```

```

</tr>
</thead>
<tbody>
{%
  for obj in query
%}
<tr>
  <th scope="row">{{ obj.idsa }}</th>
  <td>{{ obj.iduser }}</td>
  <td>{{ obj.address }}</td>
  <td>{{ obj.street }}</td>
  <td>
    <a class="btn btn-primary btn-xs"
      href="{{ obj.idsa }}/vfcomplete">verify</a>
  </td>
</tr>
{%
  endfor
%}
</tbody>
</table>
</div>
</div>
</div>
</div>
</body>
</html>

```

3.5.6.2 The url path is as follows.

path('SP/verify', supervise.views.verify),

3.5.6.3 The view function is as follows

```

""""
This is the administrator's review site code view function.
This function allows you to display the site code information submitted by users.
After approval, you can create a QR code."""

def verify(req):
    query = models.SiteAd.objects.all()
    return render(req, 'siteverify.html', {"query": query})

```

3.6. Plug-in Components

3.6.1 Pagination

This plug-in will be used in some pages that display data. This control sets every ten pieces of data as one page, and can jump to the searched page.

3.6.1.1 Pagination code

```
from django.utils.safestring import mark_safe
from django.http.request import QueryDict
import copy

class Pagination(object):

    def __init__(self, request, queryset, page_size=10, page_param="page", plus=5):

        query_dict = copy.deepcopy(request.GET)
        query_dict.mutable = True
        self.query_dict = query_dict
        self.page_param = page_param
        page = request.GET.get(page_param, "1")
        if page.isdecimal():
            page = int(page)
        else:
            page = 1
        self.page = page
        self.page_size = page_size
        self.start = (page - 1) * 10
        self.end = page * 10

        self.page_queryset = queryset[self.start:self.end]

        #total count
        total = queryset.count()

        page_count, div = divmod(total, page_size)
        plus = 5
        if div:
            page_count += 1
        self.page_count = page_count
        self.plus = plus

    def pages(self):
        #Plus with total pages less than twice the number of pages
        if self.page_count <= 2 * self.plus + 1:
            start_page = 1
            end_page = self.page_count
        else:
            #The current number of pages is less than 5, and the start page is set to 1
            if self.page <= self.plus:
                start_page = 1
```

```

        end_page = 2 * self.plus
    else:
        #The number of pages is greater than the total number of data
        if (self.page + self.plus) > self.page_count:
            start_page = self.page - 5
            end_page = self.page_count
            start_page = self.page - self.plus
            end_page = self.page + self.plus

    page_list = []

    if self.page > 1:
        self.query_dict.setlist(self.page_param, [self.page - 1])
        prev = '<li class=""><a href=?{}>Prev</a></li>'.format(self.query_dict.urlencode())
    else:
        self.query_dict.setlist(self.page_param, [1])
        prev = '<li class=""><a href=?{}>Prev</a></li>'.format(self.query_dict.urlencode())
    page_list.append(prev)

    for i in range(start_page, end_page + 1):
        self.query_dict.setlist(self.page_param, [i])
        if i is self.page:
            ele = '<li class="active"><a href=?{}>{}</a></li>'.format(self.query_dict.urlencode(), i)
        else:
            ele = '<li ><a href=?{}>{}</a></li>'.format(self.query_dict.urlencode(), i)
        page_list.append(ele)

    if self.page < self.page_count:
        self.query_dict.setlist(self.page_param, [self.page + 1])
        Next = '<li class=""><a href=?{}>Next</a></li>'.format(self.query_dict.urlencode())
    else:
        self.query_dict.setlist(self.page_param, [self.page_count])
        Next = '<li class=""><a href=?{}>Next</a></li>'.format(self.query_dict.urlencode())
    page_list.append(Next)

    search_string = """
        <li>
            <form method="get" style="float: left; margin-left: -1px">
                <div class="input-group" style="width: 200px"></div>
                <input type="text" name="q" value="{}">
            </form>
        </li>
    """

```

```

style="position:relative;float:left;display:
inline-block;width: 80px;border-radius: 0"
name="page" class="form-control"
placeholder="page...">
<button class="btn btn-default" style="border-radius: 0"
type="submit">Skip</button>
</form>
</li>
      """
page_list.append(search_string)
page_str = mark_safe("".join(page_list))
return page_str

```

3.6.2 Middleware

Middleware settings. Its function is that when users access the system, they cannot access other functions in the system without logging in first.

3.6.2.1 Middleware code

```

from django.utils.deprecation import MiddlewareMixin
from django.shortcuts import HttpResponseRedirect, render, redirect

class Author(MiddlewareMixin):
    def process_request(self, request):
        if request.path_info == "/index/":
            return
        if request.path_info == "/user/register":
            return
        if request.path_info == "/user/login":
            return
        if request.path_info == "/index2/":
            return
        info_dict = request.session.get("Info")
        if info_dict:
            return

```

```

        return redirect("/user/login")

class Role(MiddlewareMixin):

    def process_request(self, request):
        Info = request.session.get("Info")
        if Info is not None:
            role = Info["role"]
            if (request.path.startswith('/SP/') or request.path.startswith(
                '/index2')) \
                and not role == "admin":
                print(role)
                return redirect('/index')
            response = self.get_response(request)
            return response
        else:
            return

```

3.6.3 Verification Code Function

Verification code generation function, where the system randomly generates verification codes. Users must enter the correct verification code to log in. This can prevent some violent cracking and malicious access.

3.6.3.1 Verification Code

```

import random

from PIL import Image, ImageFont, ImageDraw, ImageFilter

```

"""This is the login verification code tool.

When logging in, users need to enter a verification code

within one minute to log in,

otherwise they cannot log in even if the user has logged in with the correct account and password."""

```

def check_code(width=120, height=30, char_length=5, font_file='Monaco.ttf', font_size=28):
    code = []

```

```


```

img = Image.new(mode='RGB', size=(width, height), color=(255, 255, 255))

draw = ImageDraw.Draw(img, mode='RGB')

def rndChar():
 """
 Generate random letters
 """
 return chr(random.randint(65, 90))

def rndColor():
 """
 Generate random colors
 :return:
 """
 return (random.randint(0, 255), random.randint(10, 255), random.randint(64, 255))

Write verification code content

font = ImageFont.truetype(font_file, font_size)

for i in range(char_length):
 char = rndChar()
 code.append(char)

 h = random.randint(0, 4)
 draw.text([i * width / char_length, h], char, font=font, fill=rndColor())

Fill interference points

for i in range(40):
 draw.point([random.randint(0, width), random.randint(0, height)], fill=rndColor())

Fill interference circle

```


```

```

for i in range(40):
    draw.point([random.randint(0, width), random.randint(0, height)], fill=rndColor())
    x = random.randint(0, width)
    y = random.randint(0, height)
    draw.arc((x, y, x + 4, y + 4), 0, 90, fill=rndColor())

# Fill interference line
for i in range(5):
    x1 = random.randint(0, width)
    y1 = random.randint(0, height)
    x2 = random.randint(0, width)
    y2 = random.randint(0, height)

    draw.line((x1, y1, x2, y2), fill=rndColor())

img = img.filter(ImageFilter.EDGE_ENHANCE_MORE)
return img, ''.join(code)

```

4. Program Test

4.1 App Test

Tester: Zhao Shirui

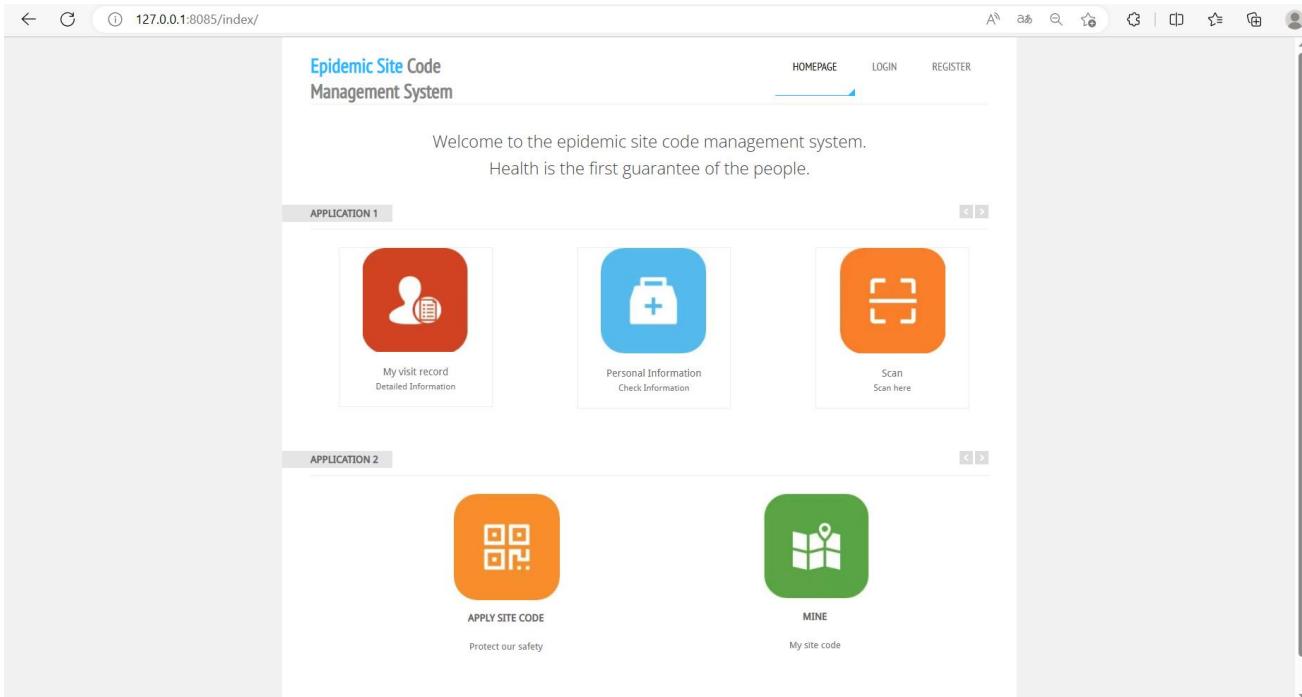
4.1.1 Index App

4.1.1.1 User Main Interface

Action: Enter the website for users of the epidemic site code management system.

Expected: Display users interface.

Actual: The main interface used by users can be displayed.



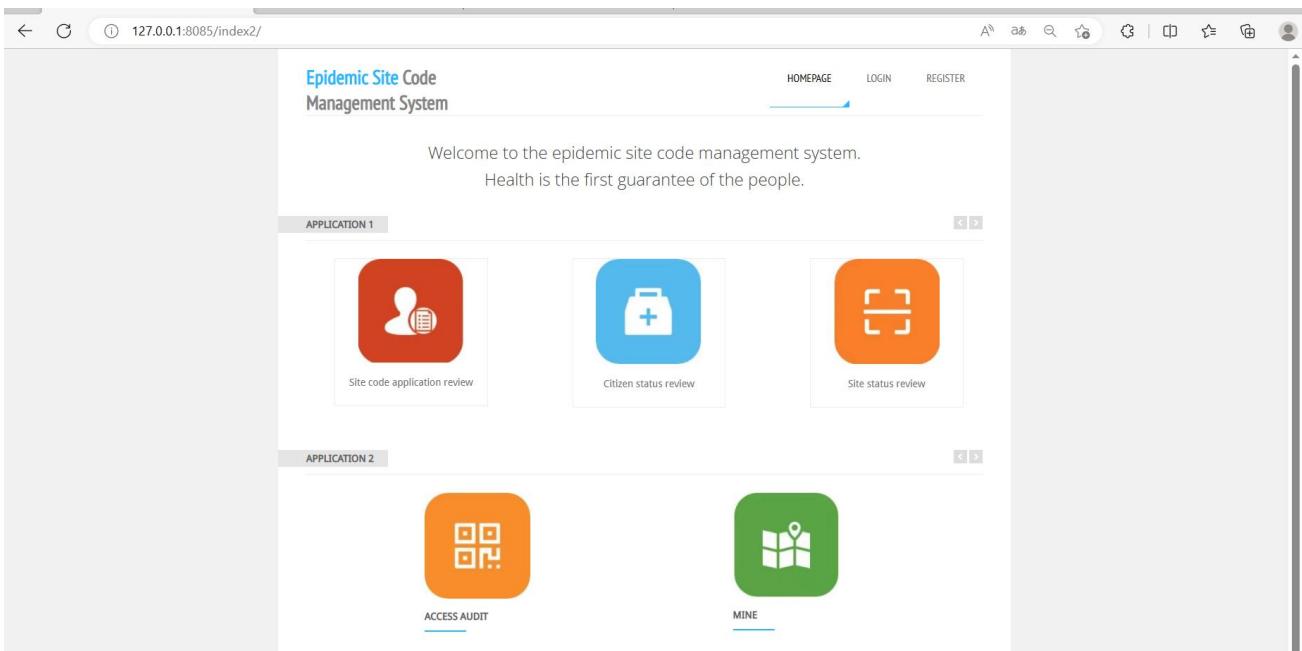
Result: Pass

4.1.1.2 Admin Main Interface

Action: Enter the website for admins of the epidemic site code management system.

Expected: Display admins interface.

Actual: The main interface used by admins can be displayed.



Result: Pass

4.1.2 User App

4.1.2.1 Register Interface

Action: Click Register

Expected: Enter the register interface.

Actual: The register interface is displayed. If there is no account, the user can register first.

The screenshot shows a web browser window with the URL 127.0.0.1:8085/user/register. The page title is 'Epidemic Site Code Management System.' A 'Login' link is visible in the top right corner. The main content is a registration form titled 'REGISTER TO US'. It contains three input fields: 'Phone' (placeholder: 'please input your Phone...'), 'Name' (placeholder: 'please input your Name...'), and 'Password' (placeholder: 'please input your Password...'). Below the password field is a red error message: '*Please enter upper and lower case letters or numbers, totaling 6-18 digits.' A green 'REGISTER' button is at the bottom of the form. The browser's address bar and various icons are visible at the top.

Result: Pass

4.1.2.2 Register

Action: Enter the appropriate account password.

Expected: The account uses the user's mobile phone number and the password is for user's own use. If the user enter a password that does not meet the requirements, the user will be asked to re-enter it.

Actual: If the user enter the correct information, the registration will be successful, while if the user enter the wrong information, the user will be prompted to re-enter the required information.

Enter incorrect phone number and password.

Management System.

Login

REGISTER TO US

Phone

Ensure this value has at most 11 characters (it has 12).

Name

Password

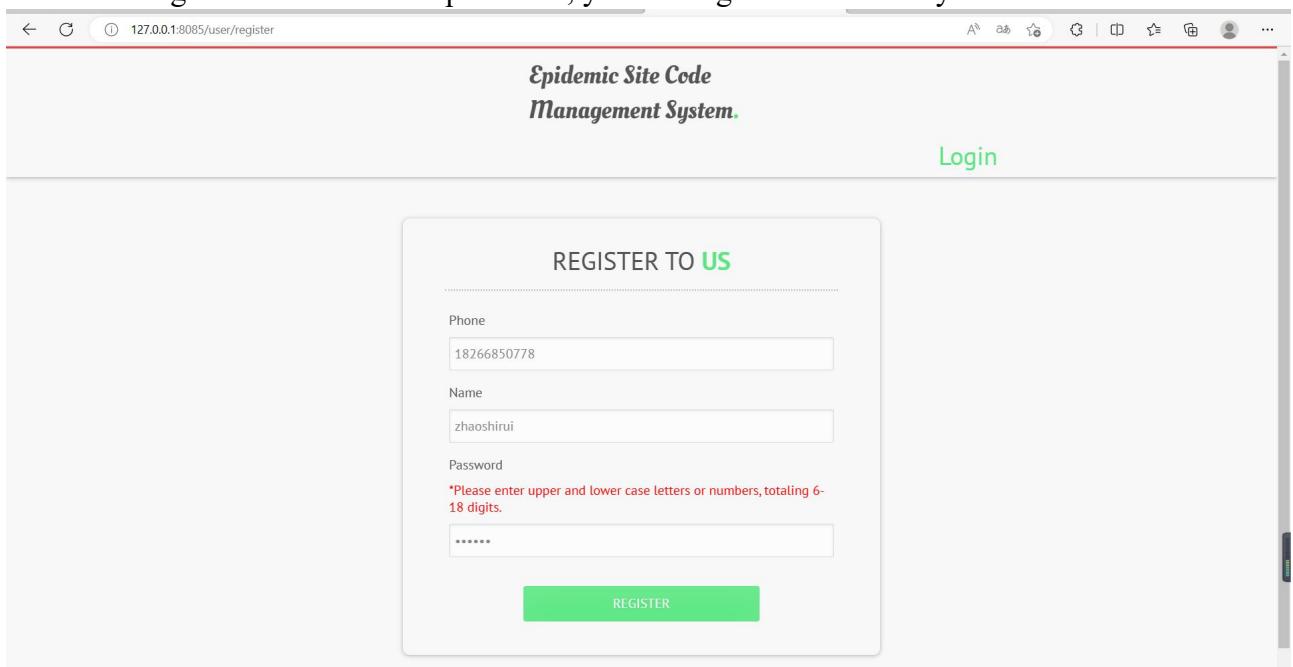
*Please enter upper and lower case letters or numbers, totaling 6-18 digits.

Enter a valid value.

REGISTER

Copyright © Epidemic Site Code Management System All rights reserved.

After entering the correct account password, you can register successfully.



The screenshot shows a web browser window with the URL `127.0.0.1:8085/user/register` in the address bar. The page title is *Epidemic Site Code Management System.* On the right side of the header, there is a green "Login" button. The main content is a registration form titled "REGISTER TO US". It includes fields for "Phone" (value: 18266850778), "Name" (value: zhaoshirui), and "Password" (value: *****). There are validation messages: "Ensure this value has at most 11 characters (it has 12)" for the phone number and "Enter a valid value." for the password field. A green "REGISTER" button is at the bottom of the form.

```
1 •  SELECT * FROM epidemic_site_code_management_system.user;
```

The screenshot shows a MySQL Workbench result grid with the following data:

	iduser	phone	name	password	status
28	18266850777	amll		96e79218965eb72c92a549dd5a330112	health
29	18266850779	amll		96e79218965eb72c92a549dd5a330112	health
30	18266850770	amlll		96e79218965eb72c92a549dd5a330112	health
31	15965101095	zsr		96e79218965eb72c92a549dd5a330112	health
32	15965101096	zsrr		96e79218965eb72c92a549dd5a330112	health
33	15965101098	zsrrr		96e79218965eb72c92a549dd5a330112	health
34	15965101099	Dawn		96e79218965eb72c92a549dd5a330112	health
35	15965101085	Dawn1		96e79218965eb72c92a549dd5a330112	health
36	15965101084	Dawn11		96e79218965eb72c92a549dd5a330112	health
37	15965101083	Dawn111		96e79218965eb72c92a549dd5a330112	health
38	17853799875			96e79218965eb72c92a549dd5a330112	health
39	17853799874			96e79218965eb72c92a549dd5a330112	health
40	17853799873	dz		96e79218965eb72c92a549dd5a330112	health
41	17853799872	litangdingzhen		96e79218965eb72c92a549dd5a330112	health
42	17853799870	litangwang		96e79218965eb72c92a549dd5a330112	health
43	17853799832	dsm		96e79218965eb72c92a549dd5a330112	health
*	NULL	NULL	NULL	NULL	NULL

If the user registers using the same phone number.

The screenshot shows a 'REGISTER TO US' form with the following fields:

- Phone: 18266850777
- Name: zss
- Password: (redacted)

A red error message at the bottom left of the form states: "Please enter upper and lower case letters or numbers, totaling 6-18 digits."

REGISTER

IntegrityError at /user/register

(1062, "Duplicate entry '18266850777' for key 'user.phone_UNIQUE'")

Request Method: POST

Request URL: http://127.0.0.1:8085/user/register

Django Version: 4.1.5

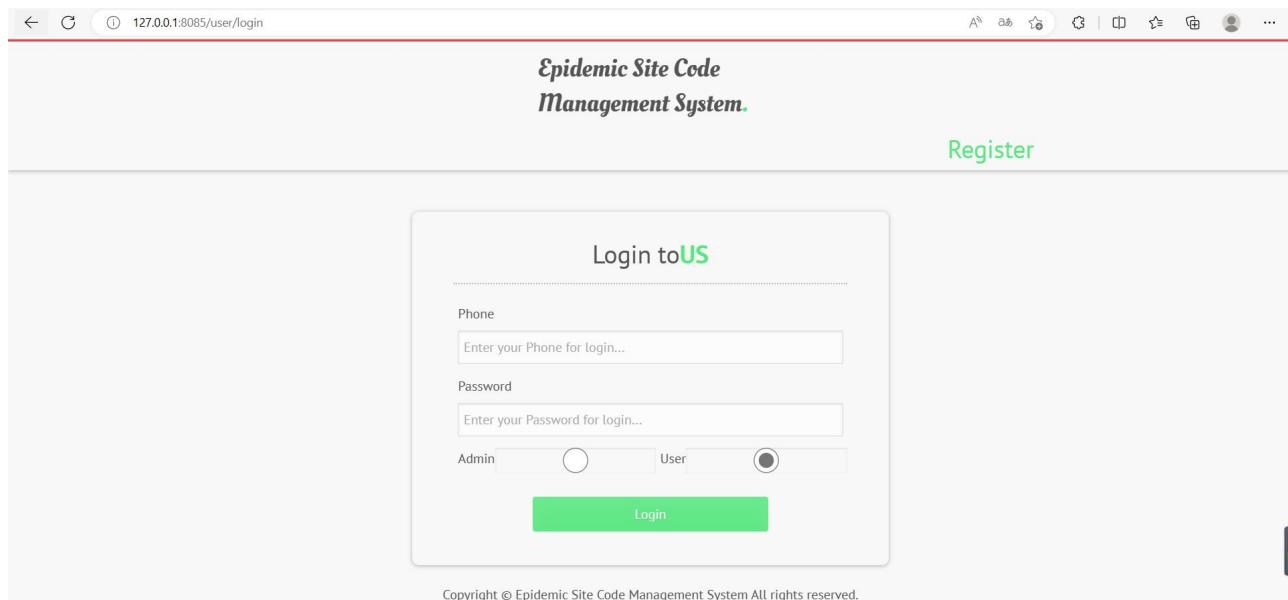
Exception Type: IntegrityError

The system displays an error indicating that this violates the unique key.

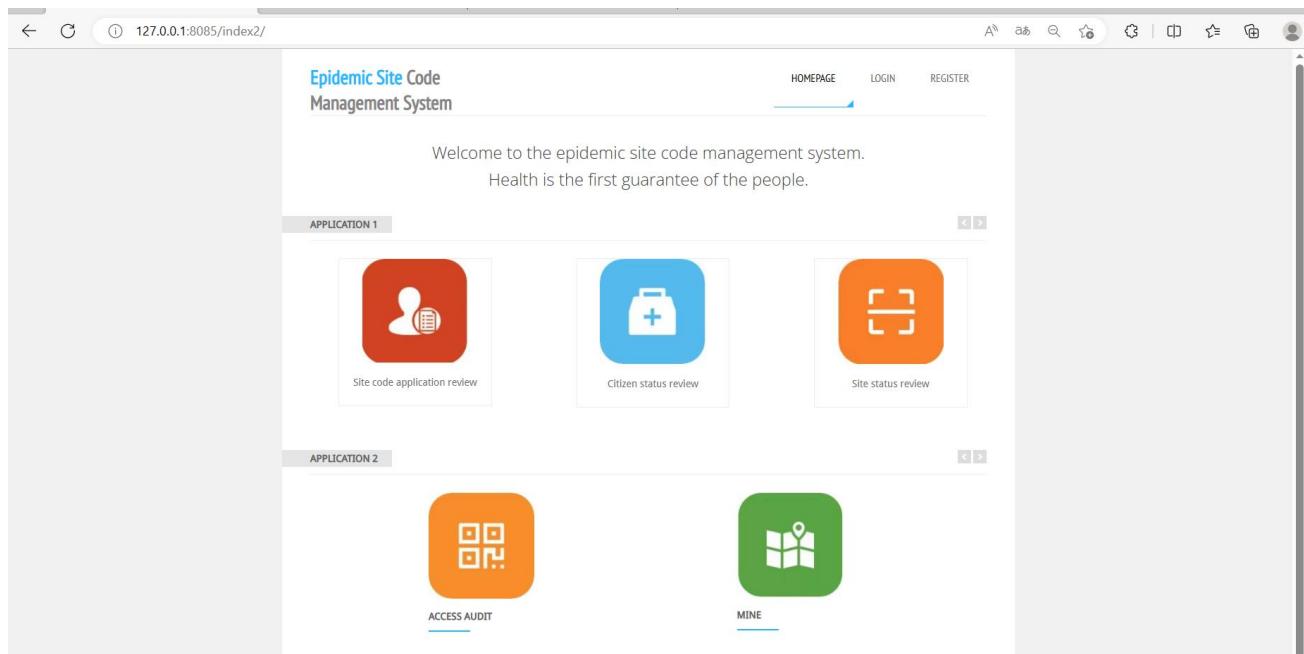
Result: Fail

4.1.2.3 Login Interface

- Action: Click Login
Expected: Enter the login interface.
Actual: The login interface is displayed. Users can log in according to existing accounts.



Because middleware is set, users cannot access other functions without logging in, except registration and login. Even if the user click other functions, the user will first jump to the login interface.



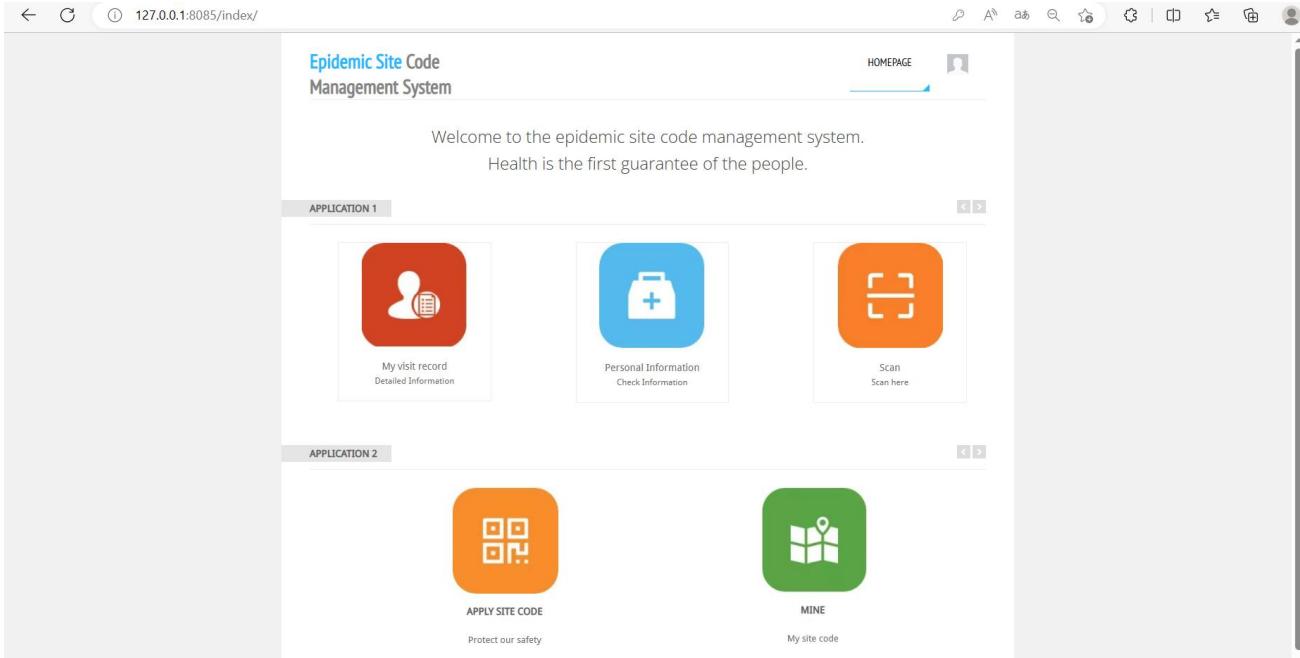
Result: Pass

4.1.2.4 Login

- Action: Enter the appropriate account and password.
- Expected: After entering the correct account and password, the user can use other features of the website. If the user make a mistake, the user will not be able to log in normally.
- Actual: If the user enter the correct information, the login will be successful, while if the user enter the wrong information, the user will be prompted to re-enter the required information.

This is an incorrect input.

When the user enter the correct information, the user can jump back to the home page and be prompted to log in at the top right corner.



Result: Pass

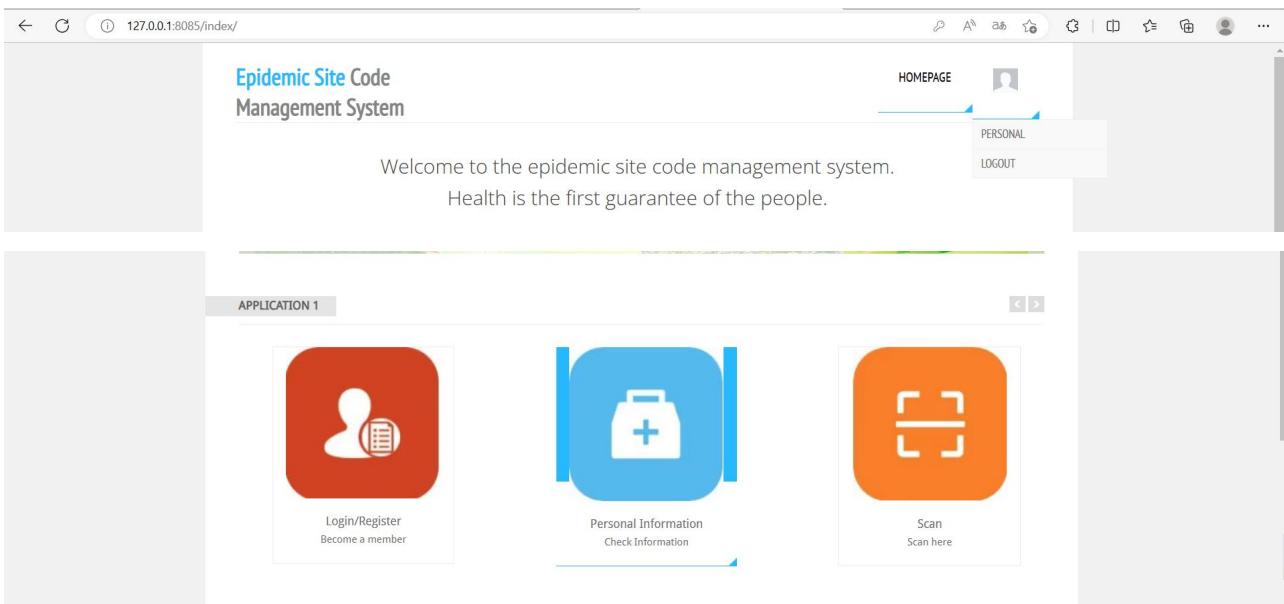
4.1.2.5 View User Information

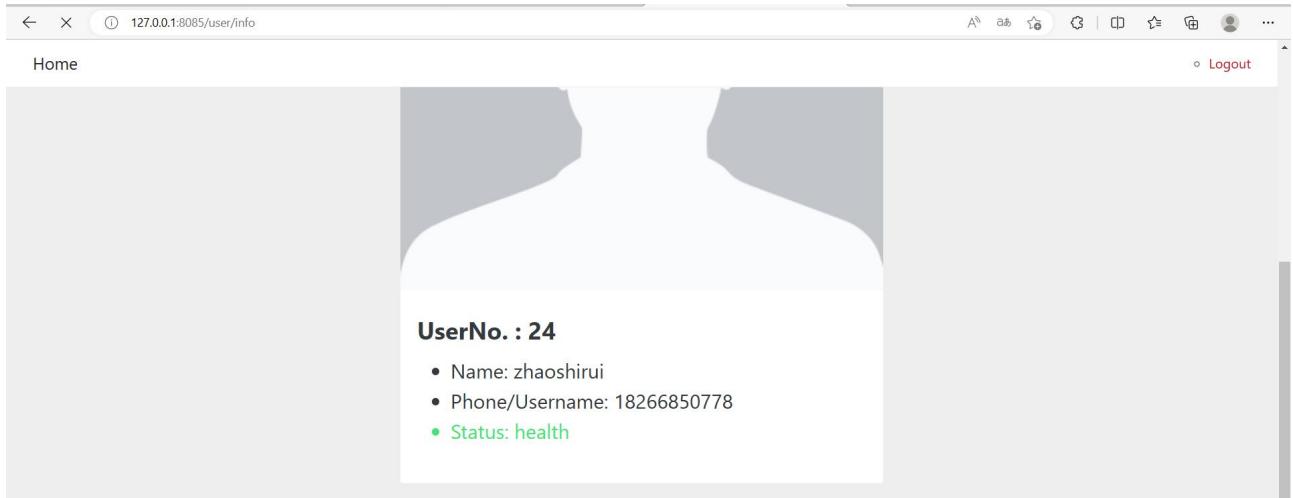
Action: Click on Personal Information Display.

Expected: After clicking this function, the main page will jump to the display interface of personal information, which contains the user name, user number, name and the health status of the user.

Actual: The main page can successfully jump to the personal information display interface.

Click the drop-down box in the upper right corner and the user information display module below to view the user's personal information.





Result: Pass

4.1.2.6 Site code application

- Action: Click to enter the site code application interface.
- Expected: After entering the correct site address and street, the information will be submitted to the Administrator. Once the Administrator has verified the location of the application, the application can be passed, and the site code will be generated.
- Actual: The user can enter the site code application interface and submit the correct information to the background database.

A screenshot of a web browser showing the "Epidemic Site Code Management System". The title is centered above a form titled "Apply for Site code". The form has two input fields: "Address" containing "Yongjia Post, Beijing University of Aeronautics and Astronautics." and "Street" containing "No. 37 Garden Road Street, Haidian District, Beijing". Below the form is a green "SUBMIT" button. At the bottom of the page, there is a copyright notice: "Copyright © Epidemic Site Code Management System All rights reserved.".

```
1 •  SELECT * FROM epidemic_site_code_management_system.site_ad;
```

	idSA	iduser	address	street
▶	1	19	sdszcs	fsnl10086
2	19	18ceng		123
*	3	24	Yongjia Post, Beijing University of Aeronautics a...	No. 37 Garden Road Street, Haidian District, Bei...
		HULL	HULL	HULL

Result: Pass

4.1.2.7 My Site Interface

Action: Click to enter MySite interface.

Expected: When a user enters my site code interface, he or she can see the site code that the user has been approved for.

Actual: Since the site code that this user had applied for has not been passed, the site code that the user has applied for cannot be seen at this time.

My Site			
Site ID	Address	Street	Status
1	sdszcs	qqjd	abnormal
2	bjshdq	hyjd	health
3	wadadaw	dwdadawd	health
4	fnlquqf	1243u890	health
5	18ceng	123	health
6	18ceng	123	health

The tester logs into another account for the test to see the sites that have been applied for. The administrator changed one place to an exception.

My Site			
Site ID	Address	Street	Status
1	sdszcs	qqjd	abnormal
2	bjshdq	hyjd	health
3	wadadaw	dwdadawd	health
4	fnlquqf	1243u890	health
5	18ceng	123	health
6	18ceng	123	health

Result: Pass

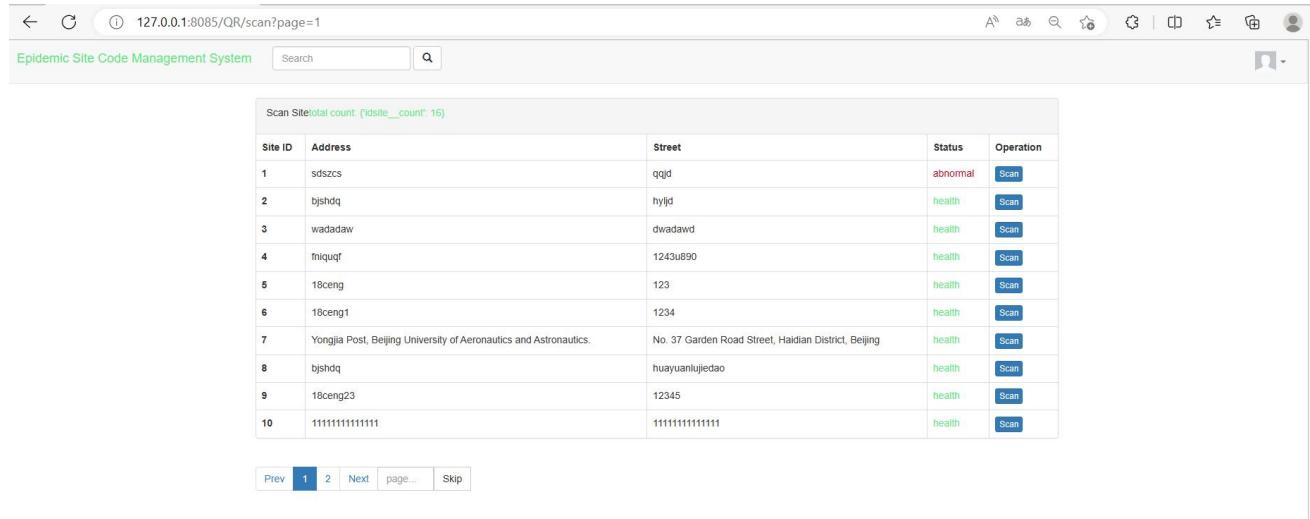
4.1.3 QR Code App

4.1.3.1 Scan Interface

Action: Click to enter Scan interface.

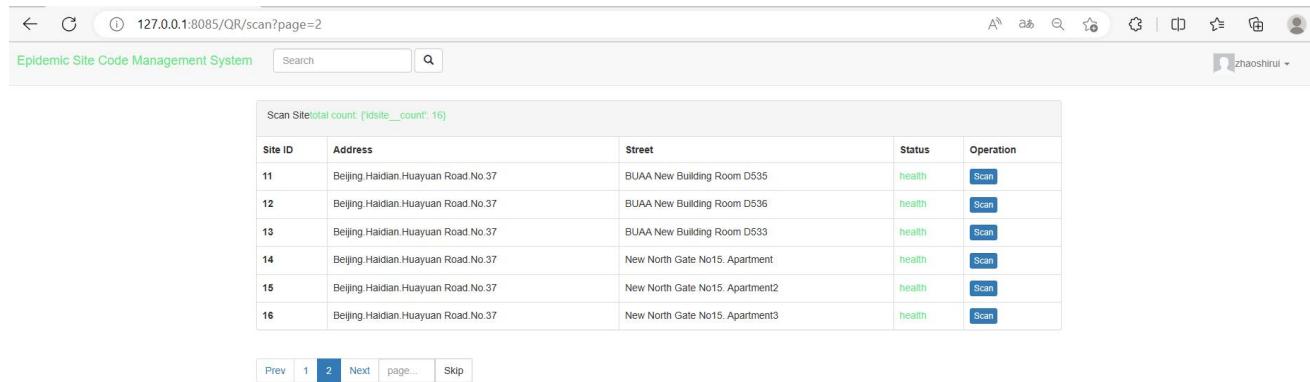
Expected: When a user enters scan interface, users can see the site that has been verified and scan the site code.

Actual: Users can see existing sites, search for them, and paginate redundant site information.



Scan Site (total count: 16)				
Site ID	Address	Street	Status	Operation
1	sdszcs	qqjd	abnormal	Scan
2	bjshdq	hyld	health	Scan
3	wadadaw	dwdadawd	health	Scan
4	fniquqf	1243u890	health	Scan
5	18ceng	123	health	Scan
6	18ceng1	1234	health	Scan
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	Scan
8	bjshdq	huayanlujiedao	health	Scan
9	18ceng23	12345	health	Scan
10	1111111111111111	1111111111111111	health	Scan

Prev 1 2 Next page... Skip



Scan Site (total count: 16)				
Site ID	Address	Street	Status	Operation
11	Beijing Haidian.Huayuan Road.No.37	BUAA New Building Room D535	health	Scan
12	Beijing Haidian.Huayuan Road.No.37	BUAA New Building Room D536	health	Scan
13	Beijing Haidian.Huayuan Road.No.37	BUAA New Building Room D533	health	Scan
14	Beijing Haidian.Huayuan Road.No.37	New North Gate No15. Apartment	health	Scan
15	Beijing Haidian.Huayuan Road.No.37	New North Gate No15. Apartment2	health	Scan
16	Beijing Haidian.Huayuan Road.No.37	New North Gate No15. Apartment3	health	Scan

Prev 1 2 Next page... Skip

Result: Failed

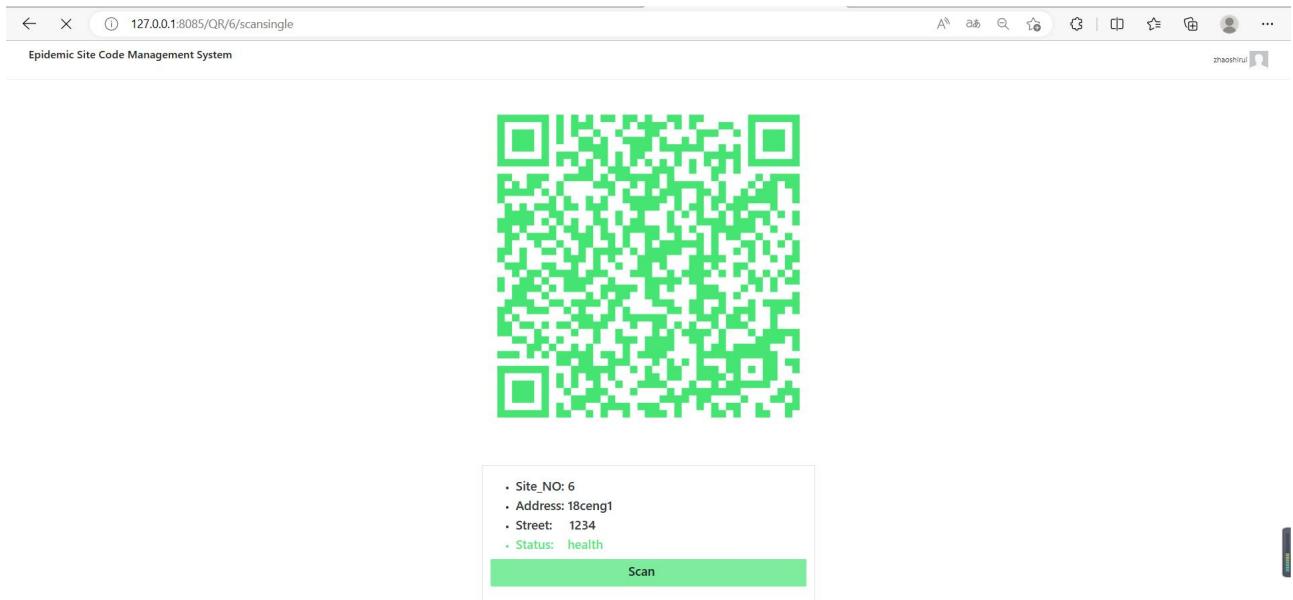
Comment: Users cannot enter places that have been adjusted to be abnormal by the administrator.

4.1.3.2 Scan QR code Interface

Action: Click Scan Function.

Expected: Users can see the detailed scanner QR code function, clicking on the scan will prompt the scan to complete.

Actual: The QR code scanning detailed interface shows a site code, as well as the site ID, site address and site status. When the scan is complete, a user's data entering the site enters the database.



The scan will be notified when clicked.



There will be a new piece of data in the database.

```
1 •  SELECT * FROM epidemic_site_code_management_system.us;  
2  
3  
4  
5
```

Result Grid			
	idsite	iduser	e_time
24	11	43	2023-04-02 22:37:49
25	16	43	2023-04-02 22:38:05
26	8	43	2023-04-02 22:38:22
27	5	43	2023-04-02 22:38:37
28	15	43	2023-04-02 22:38:49
29	14	43	2023-04-02 22:39:00
30	8	43	2023-04-02 22:39:14
31	4	43	2023-04-02 22:39:23
32	9	43	2023-04-02 22:39:35
33	8	40	2023-04-02 22:40:53
34	4	40	2023-04-02 22:41:02
35	12	40	2023-04-02 22:41:11
36	4	40	2023-04-02 22:41:21
37	15	40	2023-04-02 22:41:30
38	4	40	2023-04-02 22:41:37
39	6	24	2023-04-03 10:33:49
	HULL	HULL	HULL

Result: Pass

4.1.3.3 Generate QR code Interface

Action: Administrator Click Verify

Expected: Once the administrator has verified the user's site application, the site code can be generated.

Actual: After the administrator's review is complete, a new site code is generated and entered into the database.

At this point the admin can see the application submitted by the previous user, Zhao Shirui, for passage here.

Site ID	User ID	Address	Street	Operation
1	User object (19)	sdszcs	fsnl10086	<button>verify</button>
2	User object (19)	18ceng	123	<button>verify</button>
3	User object (24)	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	<button>verify</button>

Successful auditing is prompted and a QR code is generated.

complete generate

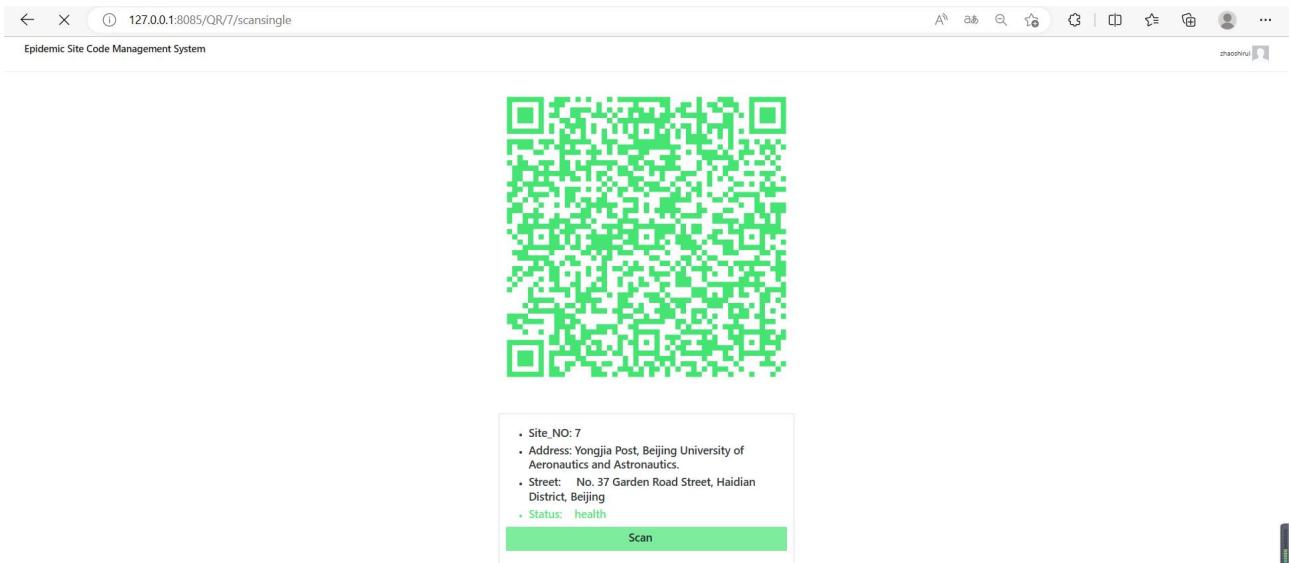
The data has been successfully generated and stored in the database.

```
1 •   SELECT * FROM epidemic_site_code_management_system.site;
```

idsite	iduser	address	street	status	idadmin	path
1	19	sdszcs	qqjd	abnormal	1	static/qrcode/1.png
2	19	bjshdq	hyjd	health	1	static/qrcode/2.png
3	19	wadadaw	dwadawd	health	1	static/qrcode/3.png
4	19	fniuqf	1243u890	health	1	static/qrcode/4.png
5	19	18ceng	123	health	1	static/qrcode/5.png
6	19	18ceng1	1234	health	1	static/qrcode/6.png
7	24	Yongjia Post, Beijing University of Aeronautics a...	No. 37 Garden Road Street, Haidian District, Bei...	health	1	static/qrcode/7.png
*	HULL	HULL	HULL	HULL	HULL	HULL

The site code for the application can now be viewed.

Site ID	Address	Street	Status
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health



Result: Pass

4.1.4 Supervise App

4.1.4.1 Login Admin

Action: Click Administrator Login

Expected: The admin cannot enter the Administrator Interface unless the admin select the Administrator role and enter the correct Administrator information to log in.

Actual: After entering the correct information, the admin can log in to the administrator interface.

Administrator information in the database.

```
1 •  SELECT * FROM epidemic_site_code_management_system.admin;
```

Result Grid			Filter Rows:	Edit:	Export/Import:
idadmin	admin_no	password			
1	admin	123			

Illegal information entered

**Epidemic Site Code
Management System.**

Register

Login toUS

Phone

Login without permission!

Password

Please enter the correct user name first

Admin User

Login

Copyright © Epidemic Site Code Management System All rights reserved.

← ⌂ ⓘ 127.0.0.1:8085/user/login

**Epidemic Site Code
Management System.**

Register

Login toUS

Phone

Password

Please enter the correct password!

Admin User

Login

Copyright © Epidemic Site Code Management System All rights reserved.

Even if the admin enter the correct information, the admin cannot log in with the wrong role.

← ⌂ ⓘ 127.0.0.1:8085/user/login

**Epidemic Site Code
Management System.**

Register

Login toUS

Phone

Password

Please enter the correct password!

Admin User

Login

Copyright © Epidemic Site Code Management System All rights reserved.

Result: Pass

4.1.4.2 Verify Site

Action: Click verify to pass the application for site code

Expected: After the administrator clicks "Pass", the data will be entered into the database.

Actual: As Expected.

Site verify					
Site ID	User ID	Address	Street		Operation
1	User object (19)	sdszcs	fsnl10086		<button>verify</button>
2	User object (19)	18ceng	123		<button>verify</button>
3	User object (24)	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing		<button>verify</button>

← ⏪ 127.0.0.1:8085/QR/generate

complete generate

idsite	iduser	address	street	status	idadmin	path
1	19	sdszcs	qqjd	abnormal	1	static/qrcode/1.png
2	19	bjsjhdq	hyljd	health	1	static/qrcode/2.png
3	19	wadadaw	dwdadawd	health	1	static/qrcode/3.png
4	19	fniuqf	1243u890	health	1	static/qrcode/4.png
5	19	18ceng	123	health	1	static/qrcode/5.png
6	19	18ceng1	1234	health	1	static/qrcode/6.png
7	24	Yongjia Post, Beijing University of Aeronautics a...	No. 37 Garden Road Street, Haidian District, Bei...	health	1	static/qrcode/7.png
*	HULL	HULL	HULL	HULL	HULL	HULL

Result: Pass

4.1.4.3 User Status modify function

Action: Click to modify user status.

Expected: Click Modify to view the user's registration information and modify the status.

Actual: The Admin can modify three states, namely health, abnormal and unknown.

← ⌂ ⓘ 127.0.0.1:8085/SP/1/user_modify

Epidemic Site Code Management System

User Information

User ID	1
Phone	1111
Name	12
Status	<input type="button" value="abnormal"/> <input type="button" value="health"/> <input type="button" value="abnormal"/> <input type="button" value="unknown"/>
Modify	

The unknown status should be displayed in yellow instead of red.

← ⌂ ⓘ 127.0.0.1:8085/SP/user_status

Epidemic Site Code Management System

User total count: {#user__count: 30}

User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<input type="button" value="modify"/>
2	1111111111	zsr	abnormal	<input type="button" value="modify"/>
3	2222222222	123	health	<input type="button" value="modify"/>
14	11111	name		<input type="button" value="modify"/>
16	321321	123		<input type="button" value="modify"/>
17	0	1	health	<input type="button" value="modify"/>
18	3213213	123	health	<input type="button" value="modify"/>
19	1111111	1	health	<input type="button" value="modify"/>
20	8008208820	An Moli	health	<input type="button" value="modify"/>
22	1234567890	z1	health	<input type="button" value="modify"/>

Prev Next page... Skip

Result: Failed

Comment: The unknown status should be displayed in yellow instead of red.

4.1.4.4 User Status function

Action: Click to enter the user status modification interface.

Expected: The administrator can view the user information in this interface. Each page can view ten user information. The admin can turn pages or enter the user number to view the specified user information.

Actual: After entering the specified page number, the search content of the user number is invalid.

APPLICATION 1

Site code application review

Citizen status review

Scan

127.0.0.1:8085/SP/user_status

Epidemic Site Code Management System

Search

User total count: ('iduser__count': 30)				
User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<button>modify</button>
2	1111111111	zsr	abnormal	<button>modify</button>
3	2222222222	123	health	<button>modify</button>
14	11111	name		<button>modify</button>
16	321321	123		<button>modify</button>
17	0	1	health	<button>modify</button>
18	3213213	123	health	<button>modify</button>
19	111111	1	health	<button>modify</button>
20	8008208820	An Moli	health	<button>modify</button>
22	1234567890	z1	health	<button>modify</button>

Prev **1** 2 3 Next page... Skip

127.0.0.1:8085/SP/user_status?page=2

Epidemic Site Code Management System

Search

User total count: ('iduser__count': 30)

User ID	Phone	Name	Status	Modify
23	1234567	z2	health	<button>modify</button>
24	18266850778	zhaoshirui	health	<button>modify</button>
25	18510507255	aml	health	<button>modify</button>
26	18510507256	aml1	health	<button>modify</button>
28	18266850777	amll	health	<button>modify</button>
29	18266850779	amill	health	<button>modify</button>
30	18266850770	amilll	health	<button>modify</button>
31	15965101095	zsr	health	<button>modify</button>
32	15965101096	zsrr	health	<button>modify</button>
33	15965101098	zsrrr	health	<button>modify</button>

Prev **1** 2 3 Next page... Skip

127.0.0.1:8085/SP/user_status?page=3

Epidemic Site Code Management System Search

User total count: ("iduser__count": 30)

User ID	Phone	Name	Status	Modify
34	15965101099	Dawn	health	<input type="button" value="modify"/>
35	15965101085	Dawn1	health	<input type="button" value="modify"/>
36	15965101084	Dawn11	health	<input type="button" value="modify"/>
37	15965101083	Dawn111	health	<input type="button" value="modify"/>
38	17853799875	lll	health	<input type="button" value="modify"/>
39	17853799874	llll	health	<input type="button" value="modify"/>
40	17853799873	dz	health	<input type="button" value="modify"/>
41	17853799872	litangdingzhen	health	<input type="button" value="modify"/>
42	17853799870	litangwang	health	<input type="button" value="modify"/>
43	17853799832	dsm	health	<input type="button" value="modify"/>

Prev 1 2 3 Next page... Skip

After entering the specified page number, the search content of the user number is invalid.

Epidemic Site Code Management System Search

User

User ID	Phone	Name	Status	Modify
23	1234567	z2	health	<input type="button" value="modify"/>
24	18266850778	zhaoshirui	health	<input type="button" value="modify"/>

Prev 1 2 Next page... Skip

Result: Failed

Comment: The paging function needs to be optimized.

4.1.4.5 Site Status function

Action: Click to enter the site status modification interface.

Expected: The administrator can view the site information in this interface. Each page can view ten site information. The admin can turn pages or enter the user number to view the specified user information.

Actual: Administrators can view all site information.

127.0.0.1:8085/SP/site_status?page=2

Epidemic Site Code Management System Search

Site total count: ("idsite__count": 16)

Site ID	User ID	Admin ID	Address	Street	Status	Modify
11	User object (43)	Admin object (1)	Beijing Haidian Huayuan Road No.37	BUAA New Building Room D535	health	<input type="button" value="modify"/>
12	User object (43)	Admin object (1)	Beijing Haidian Huayuan Road No.37	BUAA New Building Room D536	health	<input type="button" value="modify"/>
13	User object (43)	Admin object (1)	Beijing Haidian Huayuan Road No.37	BUAA New Building Room D533	health	<input type="button" value="modify"/>
14	User object (24)	Admin object (1)	Beijing Haidian Huayuan Road No.37	New North Gate No.15. Apartment	health	<input type="button" value="modify"/>
15	User object (24)	Admin object (1)	Beijing Haidian Huayuan Road No.37	New North Gate No.15. Apartment2	health	<input type="button" value="modify"/>
16	User object (24)	Admin object (1)	Beijing Haidian Huayuan Road No.37	New North Gate No.15. Apartment3	health	<input type="button" value="modify"/>

Prev 1 2 Next page... Skip

127.0.0.1:8085/SP/site_status?page=1

Epidemic Site Code Management System

Search

Site total count: {idsite__count': 16}						
Site ID	User ID	Admin ID	Address	Street	Status	Modify
1	User object (19)	Admin object (1)	sdszcs	qqjd	abnormal	modify
2	User object (19)	Admin object (1)	bjshdq	hylijd	health	modify
3	User object (19)	Admin object (1)	wadadaw	dwadawd	health	modify
4	User object (19)	Admin object (1)	fniquqf	1243u890	health	modify
5	User object (19)	Admin object (1)	18ceng	123	health	modify
6	User object (19)	Admin object (1)	18ceng1	1234	health	modify
7	User object (24)	Admin object (1)	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	modify
8	User object (24)	Admin object (1)	bjshdq	huayuanlujiedao	health	modify
9	User object (19)	Admin object (1)	18ceng23	12345	health	modify
10	User object (19)	Admin object (1)	11111111111111	11111111111111	health	modify

Prev [1](#) [2](#) Next page... Skip

Result: Pass

4.1.4.6 Site Status modify function

Action: Click to modify site status.

Expected: Click Modify to view the site's registration information and modify the status.

Actual: The Admin can modify three status, namely health, abnormal and unknown.

127.0.0.1:8085/SP/1/site_modify

Site Information

Site ID	1
User ID	User object (19)
Address	sdszcs
Street	qqjd
Admin ID	Admin object (1)
Status	health

[Modify](#)

Other information cannot be changed, only three statuses of the site can be selected.

User						
Site ID	User ID	Admin ID	Address	Street	Status	Modify
1	User object (19)	Admin object (1)	sdszcs	qqjd	health	<button>modify</button>

The health status of the 1 site has been updated. And the data in the database has changed.

```
1 •  SELECT * FROM epidemic_site_code_management_system.site;
```

idsite	iduser	address	street	status	idadmin	path
1	19	sdszcs	qqjd	health	1	static/qrcode/1.png

Result: Pass

4.1.4.7 User-Site Record Function

Action: Click to view the information of the user entering the site

Expected: The records of all users entering the site will be displayed, and the user who entered the site and when can be queried according to the site number.

Actual: The administrator can view the records of all users entering the site, and modify the status of users and sites based on this.

User Visit Record total count: ('idus__count': 39)			
Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
2	Site object (4)	User object (19)	March 12, 2023, 12:04 p.m.
3	Site object (1)	User object (19)	March 12, 2023, 8:10 p.m.
4	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
5	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
6	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
8	Site object (8)	User object (19)	March 14, 2023, 10:27 p.m.
9	Site object (13)	User object (24)	April 2, 2023, 10:33 p.m.
10	Site object (16)	User object (24)	April 2, 2023, 10:33 p.m.

Prev [1](#) [2](#) [3](#) [4](#) Next page... Skip

127.0.0.1:8085/SP/us_status?page=2

Epidemic Site Code Management System

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
11	Site object (12)	User object (24)	April 2, 2023, 10:33 p.m.
12	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.
13	Site object (13)	User object (24)	April 2, 2023, 10:34 p.m.
14	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.
15	Site object (13)	User object (24)	April 2, 2023, 10:35 p.m.
16	Site object (11)	User object (43)	April 2, 2023, 10:35 p.m.
17	Site object (16)	User object (43)	April 2, 2023, 10:36 p.m.
18	Site object (16)	User object (43)	April 2, 2023, 10:36 p.m.
19	Site object (13)	User object (43)	April 2, 2023, 10:36 p.m.
20	Site object (12)	User object (43)	April 2, 2023, 10:36 p.m.

Prev 1 2 3 4 Next page... Skip

127.0.0.1:8085/SP/us_status?page=3

Epidemic Site Code Management System

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
21	Site object (7)	User object (43)	April 2, 2023, 10:37 p.m.
22	Site object (6)	User object (43)	April 2, 2023, 10:37 p.m.
23	Site object (12)	User object (43)	April 2, 2023, 10:37 p.m.
24	Site object (11)	User object (43)	April 2, 2023, 10:37 p.m.
25	Site object (16)	User object (43)	April 2, 2023, 10:38 p.m.
26	Site object (8)	User object (43)	April 2, 2023, 10:38 p.m.
27	Site object (5)	User object (43)	April 2, 2023, 10:38 p.m.
28	Site object (15)	User object (43)	April 2, 2023, 10:38 p.m.
29	Site object (14)	User object (43)	April 2, 2023, 10:39 p.m.
30	Site object (8)	User object (43)	April 2, 2023, 10:39 p.m.

Prev 1 2 3 4 Next page... Skip

127.0.0.1:8085/SP/us_status?page=4

Epidemic Site Code Management System

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
31	Site object (4)	User object (43)	April 2, 2023, 10:39 p.m.
32	Site object (9)	User object (43)	April 2, 2023, 10:39 p.m.
33	Site object (8)	User object (40)	April 2, 2023, 10:40 p.m.
34	Site object (4)	User object (40)	April 2, 2023, 10:41 p.m.
35	Site object (12)	User object (40)	April 2, 2023, 10:41 p.m.
36	Site object (4)	User object (40)	April 2, 2023, 10:41 p.m.
37	Site object (15)	User object (40)	April 2, 2023, 10:41 p.m.
38	Site object (4)	User object (40)	April 2, 2023, 10:41 p.m.
39	Site object (6)	User object (24)	April 3, 2023, 10:33 a.m.

Prev 1 2 3 4 Next page... Skip

Result: Pass

4.2 Additional Tests

- Action: In the search box, search based on any information (except time).
- Expected: According to the search criteria, a fuzzy search is performed to search for the corresponding data.
- Actual: Find the desired data.

Find the visit record numbered 23.

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
23	Site object (12)	User object (43)	April 2, 2023, 10:37 p.m.

Prev 1 Next page... Skip

Find the scan record for user 24.

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
4	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
5	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
6	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
9	Site object (13)	User object (24)	April 2, 2023, 10:33 p.m.
10	Site object (16)	User object (24)	April 2, 2023, 10:33 p.m.
11	Site object (12)	User object (24)	April 2, 2023, 10:33 p.m.
12	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.
13	Site object (13)	User object (24)	April 2, 2023, 10:34 p.m.
14	Site object (11)	User object (24)	April 2, 2023, 10:34 p.m.

Prev 1 2 Next page... Skip

In the search bar on the right, you can search based on time. The time can be refined to minutes for fuzzy search. The results include year, month, day, hour, minute, and so on.

User Visit Record total count: ('idus__count': 39)

Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
2	Site object (4)	User object (19)	March 12, 2023, 12:04 p.m.
3	Site object (1)	User object (19)	March 12, 2023, 8:10 p.m.
4	Site object (6)	User object (24)	March 13, 2023, 10:56 p.m.
5	Site object (8)	User object (24)	March 14, 2023, 10:32 a.m.
6	Site object (8)	User object (24)	March 14, 2023, 10:33 a.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.
8	Site object (8)	User object (19)	March 14, 2023, 10:27 p.m.
39	Site object (6)	User object (24)	April 3, 2023, 10:33 a.m.

Prev 1 Next page... Skip

The screenshot shows a web browser window with the URL `127.0.0.1:8085/SP/us_status?search=&s_time=17`. The title bar says "Epidemic Site Code Management System". The main content area displays a table titled "User Visit Record" with a total count of 39 records. The table has four columns: "Visit Record", "Site ID", "User ID", and "Visit time". Two rows are visible: one for "Site object (1)" with User ID 19 and Visit time March 11, 2023, 5:07 p.m.; and another for "Site object (7)" with User ID 24 and Visit time March 14, 2023, 5:51 p.m. Below the table are navigation buttons: "Prev", "1", "Next", "page...", and "Skip".

User Visit Record [total count: 'idus__count': 39]			
Visit Record	Site ID	User ID	Visit time
1	Site object (1)	User object (19)	March 11, 2023, 5:07 p.m.
7	Site object (7)	User object (24)	March 14, 2023, 5:51 p.m.

Prev **1** Next page... Skip

All interfaces that provide search can be searched in this way.

Result: Pass

4.3. Debug

4.3.1 Scan Interface

```
def qrscan(req):
    if req.method == "GET":
        value_dict = {"status": "health"}
        value = req.GET.get("search")
        if value:
            value_dict["idsite"] = value
        queryset = Site.objects.filter(**value_dict)
        page_object = Pagination(req, queryset)
        page_queryset = page_object.page_queryset
        page_str = page_object.pages()
        content = {'queryset': page_queryset, "page_str": page_str}
        return render(req, "scan.html", content)
```

Cause error: The search condition 'status' is not taken into account when searching with model.

Error code: queryset = Site.objects.filter(**value_dict)

How to Fix: Add search criteria in the model.

Fix Result:

Scan Site site count: {idsite__count: 16}				
Site ID	Address	Street	Status	Operation
1	sdszcs	qqjd	abnormal	Scan
2	bjsdq	hyld	health	Scan
3	wadadaw	dwdadawd	health	Scan
4	fniquqf	1243u890	health	Scan
5	18ceng	123	health	Scan
6	18ceng1	1234	health	Scan
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	Scan
8	bjsdq	huayuanlijiedao	health	Scan
9	18ceng23	12345	health	Scan
10	1111111111111111	1111111111111111	health	Scan

The abnormal site will not appear on the scan site.

Scan Site [total count: 16]					
Site ID	Address	Street	Status	Operation	
2	bjsdq	hyldj	health	<button>Scan</button>	
3	wadadaw	dwdadawd	health	<button>Scan</button>	
4	fniuqf	1243u890	health	<button>Scan</button>	
5	18ceng	123	health	<button>Scan</button>	
6	18ceng1	1234	health	<button>Scan</button>	
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	<button>Scan</button>	
8	bjsdq	huayuanlujedao	health	<button>Scan</button>	
9	18ceng23	12345	health	<button>Scan</button>	
10	11111111111111	11111111111111	health	<button>Scan</button>	
11	Beijing Haidian Huayuan Road No.37	BUAA New Building Room D535	health	<button>Scan</button>	

Prev 1 2 Next page... Skip

Result: Pass

4.3.2 User Status function

```
def user_status(req):
    if req.method == "GET":
        value_dict = {}
        value = req.GET.get("search")
        if value:
            value_dict["iduser"] = value
        queryset = models.User.objects.filter(**value_dict)
        page_object = Pagination(req, queryset)
        page_queryset = page_object.page_queryset

        page_str = page_object.pages()
        content = {"user": page_queryset, "page_str": page_str}
    return render(req, "user_status.html", content)
```

Cause error: Neither in the view function nor in the plug-in, the problem of url parameter coverage has been mentioned.

Error code: `user_status(req):`

How to Fix: Splice the content covered by the url.

Fix Result:

In the plug-in, assign values to the parameters passed by the url and keep them during page switching.

```

class Pagination(object):

    def __init__(self, request, queryset, page_size=10, page_param="page", plus=5):

        query_dict = copy.deepcopy(request.GET)
        query_dict.mutable = True
        self.query_dict = query_dict
        self.page_param = page_param
        .....
        if self.page > 1:
            self.query_dict.setlist(self.page_param, [self.page - 1])
            prev = '<li class=""><a href="?{}">Prev</a></li>'.format(self.query_dict.urlencode())
        else:

```

The content in the original search box is successfully retained and the search is completed.

User				
User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<button>modify</button>

Prev **1** Next page... Skip

Result: Pass

4.3.3 User Status Modify function

```

    {% if obj.status == "health" %}
        <td id="bgc_c">{{ obj.status }}</td>
    {% else %}
        <td id="bgc_r">{{ obj.status }}</td>
    {% endif %}

```

Cause error: The logic is not flexible. It only considers the health status and non-health status, without thinking about other states.

Error code:

```

    {% if obj.status == "health" %}
        <td id="bgc_c">{{ obj.status }}</td>
    {% else %}
        <td id="bgc_r">{{ obj.status }}</td>
    {% endif %}

```

How to Fix: Add logic judgment and unknown state judgment to it.

Fix Result:

```

#bgc_c{
    color: #5de784;
}
#bgc_r{
    color: #bd081c;
}
#bgc_y{
    color: #e0b849;
}

```

Add a color selector to enable the system to select colors.

```

<% if obj.status == "health" %>
    <td id="bgc_c">{{ obj.status }}</td>
<% elif obj.status == "unknown" %>
    <td id="bgc_y">{{ obj.status }}</td>
<% else %>
    <td id="bgc_r">{{ obj.status }}</td>
<% endif %>

```

The unknown status is displayed.

User				
User ID	Phone	Name	Status	Modify
1	1111	12	unknown	<button>modify</button>
2	1111111111	zsr	abnormal	<button>modify</button>
3	2222222222	123	health	<button>modify</button>

Result: Pass

4.3.4 User Register function

```

class MyForm(forms.ModelForm):
    password = forms.CharField(label="Password",
    validators=[RegexValidator(r'^[0-9a-zA-Z]{6,18}$')])
    phone = forms.CharField(label="Phone", validators=[RegexValidator(r'^[0-9]{7,12}$')])

    class Meta:
        model = User
        fields = ["phone", "name", "password"]

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for name, field in self.fields.items():
            if name == "password":
                field.widget.input_type = 'password'

            field.widget.attrs = {"placeholder": "please input your " + field.label + "..."}

```

Cause error: The submitted data is not compared with the information in the database. If the comparison is not performed, it will be judged that the unique key does not match.

How to fix: Add a judgment function for the submitted data. When it is verified that there is already relevant data in the database, the front-end page will be returned and an error will be prompted.

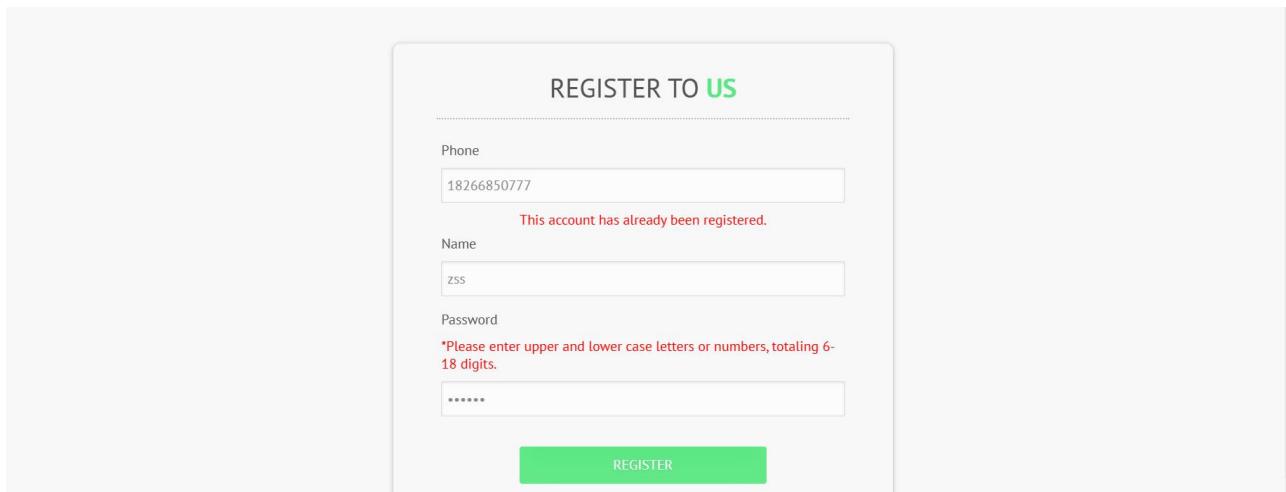
Fix Result:

```
class MyForm(forms.ModelForm):
    password = forms.CharField(label="Password",
                               validators=[RegexValidator(r'^[0-9a-zA-Z]{6,18}$')])
    phone = forms.CharField(label="Phone", validators=[RegexValidator(r'^[0-9]{7,12}$')])

    class Meta:
        model = User
        fields = ["phone", "name", "password"]

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for name, field in self.fields.items():
            if name == "password":
                field.widget.input_type = 'password'

        field.widget.attrs = {"placeholder": "please input your " + field.label + "..."}  
  
    def clean_phone(self):
        txt_phone = self.cleaned_data["phone"]
        query = User.objects.filter(phone=txt_phone)
        if bool(query):
            raise ValidationError("This account has already been registered.")
        return txt_phone
```

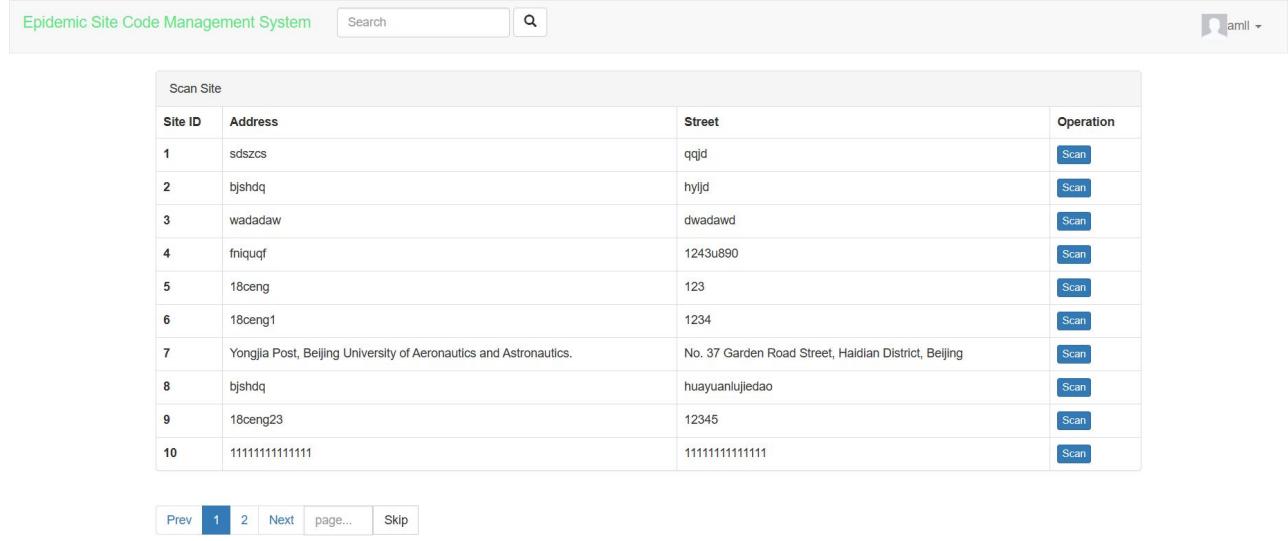


The screenshot shows a registration form titled 'REGISTER TO US'. It has three fields: 'Phone' (containing '18266850777'), 'Name' (containing 'ZSS'), and 'Password' (containing '*****'). Below the 'Phone' field, a red error message says 'This account has already been registered.' Below the 'Password' field, another red message says '*Please enter upper and lower case letters or numbers, totaling 6-18 digits.' At the bottom is a green 'REGISTER' button.

Result: Pass

4.3.5 Optimize the scanning interface

In the scanning interface, the user needs to click in a single QR code interface to view the status of the location, and cannot directly display location information.



The screenshot shows a web application titled "Epidemic Site Code Management System". At the top, there is a search bar with a placeholder "Search" and a magnifying glass icon. To the right of the search bar is a user profile icon with the name "Jiamli". Below the header is a table titled "Scan Site". The table has columns for "Site ID", "Address", "Street", and "Operation". Each row contains a "Scan" button in the "Operation" column. The data in the table is as follows:

Site ID	Address	Street	Operation
1	sdszcs	qqjd	Scan
2	bjshdq	hyld	Scan
3	wadadaw	dwadawd	Scan
4	fniquqf	1243u890	Scan
5	18ceng	123	Scan
6	18ceng1	1234	Scan
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	Scan
8	bjshdq	huayuanlujiedao	Scan
9	18ceng23	12345	Scan
10	1111111111111111	1111111111111111	Scan

At the bottom of the table, there are navigation links: "Prev", "1", "2", "Next", "page...", and "Skip".

Cause error: In the scan interface, a table was not added to display the status of each location, making it easy for users to wonder if the location is a safe location.

How to Fix: In the scan interface, add a column of tables to display the status of each location.

Fix code:

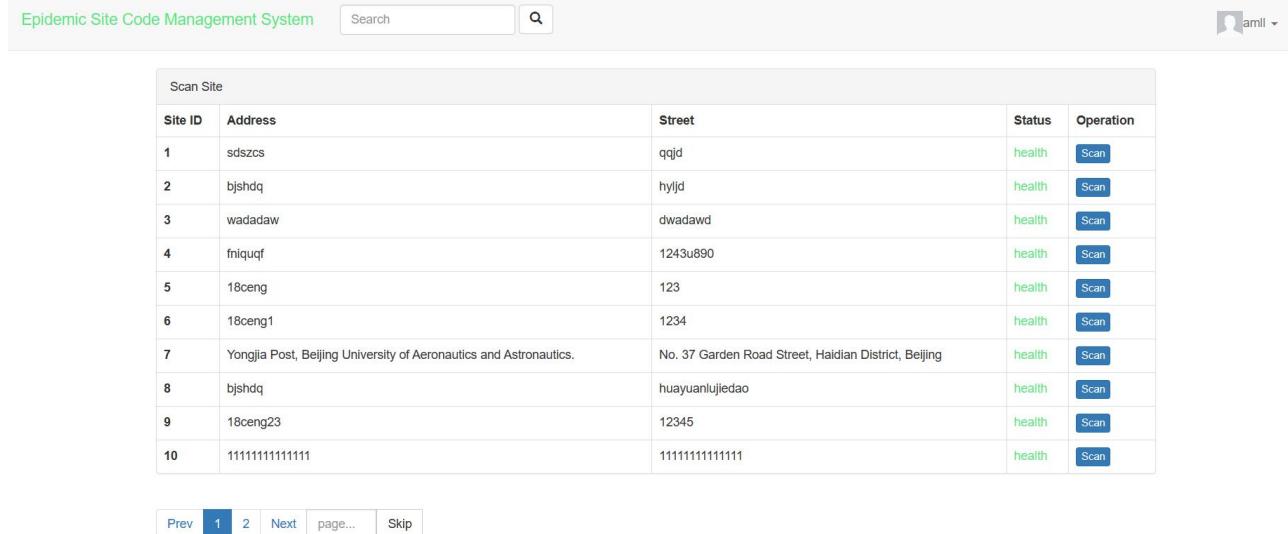
```
<table class="table table-bordered">
    <thead>
        <tr>
            <th>Site ID</th>
            <th>Address</th>
            <th>Street</th>
            <th>Status</th>
            <th>Operation</th>
        </tr>
    </thead>
    <tbody>
        {% for obj in queryset %}
        <tr>
            <th scope="row">{{ obj.idsite }}</th>
            <td>{{ obj.address }}</td>
            <td>{{ obj.street }}</td>
            {% if obj.status == "health" %}
                <td id="bgc_c">{{ obj.status }}</td>
            {% elif obj.status == "unknown" %}
                <td id="bgc_y">{{ obj.status }}</td>
            {% else %}
                <td id="bgc_r">{{ obj.status }}</td>
            {% endif %}
            <td>
```

```

<a class="btn btn-primary btn-xs" href="#"><{obj.idsite}/scansingle">Scan</a>
</td>
</tr>
{% endfor %}
</tbody>
</table>

```

Fix Results:



The screenshot shows a web application interface titled "Epidemic Site Code Management System". At the top, there is a search bar with a magnifying glass icon and a user profile icon labeled "amli". Below the header is a table titled "Scan Site". The table has columns: Site ID, Address, Street, Status, and Operation. There are 10 rows of data, each with a "Scan" button in the Operation column. The first row's Address is "sdszcs" and Street is "qqjd", both in green. The second row's Address is "bjshdq" and Street is "hyljd", both in green. The third row's Address is "wadadaw" and Street is "dwadawd", both in green. The fourth row's Address is "fniquqf" and Street is "1243u890", both in green. The fifth row's Address is "18ceng" and Street is "123", both in green. The sixth row's Address is "18ceng1" and Street is "1234", both in green. The seventh row's Address is "Yongjia Post, Beijing University of Aeronautics and Astronautics." and Street is "No. 37 Garden Road Street, Haidian District, Beijing", both in green. The eighth row's Address is "bjshdq" and Street is "huayuanlujiedao", both in green. The ninth row's Address is "18ceng23" and Street is "12345", both in green. The tenth row's Address is "1111111111111111" and Street is "1111111111111111", both in green. Below the table is a navigation bar with buttons for "Prev", "1", "2", "Next", "page...", and "Skip".

Scan Site				
Site ID	Address	Street	Status	Operation
1	sdszcs	qqjd	health	<button>Scan</button>
2	bjshdq	hyljd	health	<button>Scan</button>
3	wadadaw	dwadawd	health	<button>Scan</button>
4	fniquqf	1243u890	health	<button>Scan</button>
5	18ceng	123	health	<button>Scan</button>
6	18ceng1	1234	health	<button>Scan</button>
7	Yongjia Post, Beijing University of Aeronautics and Astronautics.	No. 37 Garden Road Street, Haidian District, Beijing	health	<button>Scan</button>
8	bjshdq	huayuanlujiedao	health	<button>Scan</button>
9	18ceng23	12345	health	<button>Scan</button>
10	1111111111111111	1111111111111111	health	<button>Scan</button>

Result: Pass.

4.4 Conclusion

4.4.1 Purpose of testing:

This document has been tested for these purposes in order to verify that the outbreak site code management system works properly and to check the system for defects and to assess the performance of the system.

4.4.2 Test scope:

This test covers all modules, which includes index, user, QRcode and supervise. The test covers the registration and login of users, the scanning of codes by users and viewing of individual codes, and user requests for site codes. It will also include changes to user and site status by the administrator, review of requests and viewing of scan logs. In addition, the test includes how to perform search queries, paging and more.

4.4.3 Testing methods:

This test uses both black box and white box testing. Black box testing is carried out by the accompanying personnel in various aspects. In contrast, white-box testing is carried out by the developers to further optimise and correct the code in response to the results of the black-box testing.

4.4.4 Test results:

In this test, there were a total of five fixes or optimizations needed, namely scan interface, user status function, user status modify function, user register function and Optimize the The errors were that the colours were not displayed properly, that the data was not determined to exist, that the contents of the search box could not be saved, and that the status of the place was not displayed. After fixing these errors, these functions worked fine when tested again.

4.4.5 Test conclusion:

The program has been tested as well as amended to meet the intended purpose and has largely met the requirements. This means that the test content is completed and passes the test.

5. Evaluating Stage

5.1 Assignment Outline And Degree of Development

The system is aimed at ordinary users and administrators.

5.1.1 Ordinary users

Ordinary users can apply for and scan the place code based on actual conditions, and can also view their own scanning records, personal information, and so on.

5.1.2 Administrator

The administrator can review the applied site code, and if the application is approved, the actual QR code can be generated, and the current site and user's health status can also be modified.

5.1.3 The Degree of Project Completion

Completed functions:

1. Completed the functions that ordinary users can register, log in, and log out.
2. Completed the functions such as applying for and viewing place codes for ordinary users.
3. Completed the function that ordinary users can scan codes and view personal information.

4. Completed the function that ordinary users can view personal scanning records.
5. Completed the function that administrators can review and view the site code.
6. Completed the function that administrators can modify user and site status.
7. The administrator can view the scanning records of all users.

5.2 Project Advantages and Disadvantages Assessment

5.2.1 Project Advantages

- 1) Easy to operate, with clear guidance and direction.
- 2) The UI is concise and can directly display the functions of each part.
- 3) Code scanning is simple and requires only a click, without the need for traditional complex processes.
- 4) The users can directly check when and where they have been. And this system provides a query system that can query corresponding search information based on search criteria.
- 5) The system provides a paging system that can display data in categories without blindly tiling the page.
- 6) The system provides a protection mechanism. When the system detects that a user has not logged in, it will first let the user log in before continuing to use other functions.
- 7) The program comes with annotations for corresponding module explanations, making it easy for management personnel to understand and maintain the code.
- 8) This program is easy to expand and provides some template tools. When the program needs to extend other functions, it can also be easily extended. This means that the coupling of the program is very low.
- 9) The system is reliable, so it can run without errors or crashes.

5.2.2 Project Disadvantages

- 1) Without contacting the mobile control, it cannot be used for code scanning and parsing like traditional code scanning tools.
- 2) The use of mobile terminals is still relatively complex, requiring functional expansion and page zooming.
- 3) For some other functions, such as the destruction of user accounts, the cancellation of site codes, and the clearing of data, there is no implementation.
- 4) For system protection, such as preventing duplicate information access, database protection, and some basic website attack protection are not implemented.
- 5) The system has not been able to make an intelligent review judgment on the site code application submitted by the user, and it must be manually reviewed to pass.

5.3 Future development

5.3.1 Solutions

- 1) Through online learning, learn how to build a WeChat or an Alipay applet based on this system, which can then use mobile terminal operations, allowing for more reasonable code scanning using mobile phone controls.
- 2) Through learning, the website can be built into a responsive structure, allowing users to access the system through applications such as mobile browsers and use it smoothly, even without using functions such as WeChat applets.
- 3) In the next update to the system, it can add functions such as destroying user accounts, deleting site codes that users have applied for, and clearing database data.

5.3.2 Reasons

- 1) In China, more and more people use WeChat. Popularizing the system to WeChat allows users to further use it. WeChat, Alipay and other small programs can help developers better connect to various controls on mobile phones.
- 2) The development of mobile terminals is necessary. Nowadays, people will always carry mobile phones, not computers, when traveling. If the system can only be accessed from the PC terminal, it will cause great inconvenience to users.
- 3) Functions such as destroying user accounts, destroying site codes, and releasing database data are basic functions like system login and registration, and are necessary for a complete system.

5.4 Modifications

5.4.1 Modification content

- 1) When I first designed this system, I hoped that it could perform a series of functions such as code scanning on the mobile terminal, just like the common health treasure. It is very complicated to operate the controls without the help of small programs such as Alipay and WeChat. Of course, this system will be transformed into an ordinary click for analog code scanning, and then the database will be entered.
- 2) This system uses python and is developed based on Django. However, when implementing the system, it was found that python does not have a hibernation function like php and can implement the automatic jump function after hibernation. Therefore, such a function must use front-end technology to achieve automatic jump.

5.4.2 Impact of Modification

- 1) When using click scan code instead of using phone controls to connect, the development difficulty is greatly reduced, and users do not need to open other software when using it. The system solves all problems in one stop.

- 2) Although there are no functions or methods inside Django that can be called to automatically jump after hibernation, it can be implemented in the front end through javascript to automatically jump after a few seconds. Compared to PHP, it is indeed a bit troublesome, but it can help me learn some new knowledge.

5.5 Identification of Any Knowledge And Skills

5.5.1 Personal gains

- 1) During this development process, I strengthened my learning of Python and learned about the use of several libraries in Python and the use of Django. Through Django, I learned a lot about the mature functions in the framework, such as the use of middleware, model form, and model, which have reduced the development difficulty. In addition, individuals can also write their own functional modules in Django based on actual situations, such as paging functions, verification code functions, and so on.
- 2) Having a rough understanding of javascript, it can be used in conjunction with the front-end to achieve functions that cannot be achieved using Django alone.

5.5.2 Improvement project

1) Project Start Phase:

At the beginning of the project, based on having learned courses such as dynamic website development and database design, I was asked to develop an epidemic site code management system. Based on the courses I have learned, I first thought of development languages such as PHP and Java. With the recommendation of my teacher, I chose to use Python to develop based on the Django framework or the Flask framework. After having a basic understanding of the difficulty and scope of application between the two, I chose to use Django for basic development.

2) Project analysis and design stage:

Through courses such as research skills, I have learned that the most suitable way for me to collect user data is through questionnaires. The data collected through the distribution of questionnaires has enabled me to understand the most basic needs of users. Based on these requirements analysis, I designed the initial database, and through reverse engineering, I obtained the E-R model of the database. Combining my knowledge of the object-oriented design course, I used UML to create use case diagrams, class diagrams, and sequence diagrams.

After learning from the actual health treasure, I designed a basic UI interface to display the system's functions and facilitate user interaction. After that, I browsed some front-end page designs online, so I made a front-end style myself according to the pattern.

3) Project development stage:

During the project development stage, I browsed various Django tutorials online, as well as how to design 2D codes, decode, encrypt, and other functions using Python. Through the Django tutorial, I designed my own epidemic prevention and control site code system, and made a test report based on the development code. In addition, I also designed some middleware, my own plug-ins, and

data protection measures through the tutorial, such as controlling the input of user data formats.

6. Documentation

6.1 Normal users

- (1) According to the prompt, the user can register and log in to the system as required. The account number must not exceed 11 valid digits and the password is a string of 6-18 characters that can be alphabetic or numeric.
- 2) The user can check the place code according to the place code number and then scan the place code.
- (3) Users can view their own scanning records.
- 4) Users can apply for the place code themselves and can view the place codes that have been approved.

6.2 Administrator

- 1) According to the prompt message, the administrator can select the admin role to log in.
- 2) The administrator can view specific information about the premises and users, as well as the ability to modify their status.
- 3) The administrator can view the sweep records of all users and can search based on certain criteria.
- 4) The administrator can review the requested venue codes and approve or reject them.

7. References

1. Bootstrap, V (2020) *Bootstrap3 [button-groups]*. Bootstrap 3.0 css. Available from: <https://v3.bootcss.com/components/#btn-groups> (Accessed: 15 February 2023).
2. Katie, S (2021) *Identifying QR codes and barcodes using Python*. 28 June. Available at: https://www.bilibili.com/video/BV12h4118748/?spm_id_from=333.788.top_right_bar_window_custom_collection.content.click&vd_source=6c2ebf299c625b54341a1fb2804efe30 (Accessed: 17 February 2023).
3. Wei, Y (2020) *Generate a QR code using Python*. 20 July. Available at: https://www.bilibili.com/video/BV1bT4y1E7Ab/?spm_id_from=333.788.top_right_bar_window_custom_collection.content.click&vd_source=6c2ebf299c625b54341a1fb2804efe30 (Accessed: 17 February 2023).
4. Wu, P (2020) *Latest Python web development bucket (Django+front-end+database)*. 5 April. Available at: https://www.bilibili.com/video/BV1rT4y1v7uQ/?spm_id_from=333.1007.top_right_bar_window_default_collection.content.click&vd_source=6c2ebf299c625b54341a1fb2804efe30 (Accessed: 16 February 2023).

8. History Version

Filename	File Version	Date	Type	Change
GU2Proposal.doc	Version 1	2022/12/19	doc	Opening Report on the Epidemic Site Code Management System Project.
GU2-Analysis& design-phase.docx	Version 1	2023/1/28	docx	Basic project analysis and design was carried out for the project itself.
GU2-Program code&test reports.doc	Version 2	2023/3/15	doc	The basic program functions and the test results are presented in a report.
GU2-Program code&test reports.doc	Version2	2023/4/3	doc	Some interfaces have been modified and some data added. And further tests have been carried out on the modified pages.
GU2-Analysis& design-phase.docx	Version 2	2023/4/8	docx	For the database design section of the project an update was made to add a table (sitead).
GU2-evaluateing -report.docx	Version 1	2023/4/8	doc	We conducted a review and review of the project, focusing on its advantages and benefits.
GU2-evaluateing -report.docx	Version 2	2023/4/10	doc	Add more advantages and personal feelings to the project.

9. Appendix

Questionnaire on requirements of epidemic air defense site code management system

I am Zhao Shirui from Beijing University of Aeronautics and Astronautics software development major. The purpose of this questionnaire is to investigate the user needs of the epidemic site code management system. This questionnaire will take 3-5 minutes. The result of this survey is only for academic purpose and it is not for the commercial purpose.

1. What is your identity? [Single choice questions] *

- Customer ([Go to question 2](#))
- Business ([Go to question 7](#))

2. What is your age? [Single choice questions] *

- Under 18
- 19-29
- 30-39
- 40-49
- Over 50

[Depends on the first option of the first question.](#)

3. Where can the system be used? [Single choice questions] *

- Nationwide
- Whole province
- Citywide

[Depends on the first option of the first question.](#)

4. Can a user apply for site qualification for family members? [Single choice questions] *

- Yes
- No

[Depends on the first option of the first question.](#)

5. Do you think it is necessary to judge the status of the site through the system? [Single choice questions] *

Yes

No

Depends on the first option of the first question.

6. What do you think this system can help customers understand? [Multiple choice questions] *

Know whether the customer is in a healthy state.

Know whether the place the customer are going to is safe.

Find out whether to travel with abnormal people synchronously.

Know whether the customer have received vaccination

Depends on the first option of the first question.

7. What kind of enterprise do you operate? [Single choice questions] *

Catering

Stores and supermarkets

Medical

Scenic spot

Depends on the second option of the first question.

8. Whether online help can be provided on the interface to solve the difficulties encountered by users during use? [Single choice questions] *

Yes

No

Depends on the second option of the first question.

9. Do you think the site operator has the right to notify and warn customers? [Single choice questions] *

Yes

No

Depends on the second option of the first question.

10. What conditions do you think are needed to apply for the site code? [Multiple choice questions]

*

Manager's personal information

Place address

Business license

Depends on the second option of the first question.

11. If this system is put into use, what functions do you think the site code needs? [Multiple choice questions] *

Quick application

Some basic information of users can be displayed after scanning

Convenient logout

Depends on the second option of the first question.

12. What do you think this system can help the site understand? [Multiple choice questions] *

It can help site managers understand whether each customer is safe.

It can help the site manager to remind the visiting customers

It can help the site management personnel to warn the abnormal customers

Depends on the second option of the first question.

13. Do you think this system needs other functions? Please list. [Fill in the blanks]

Depends on the first and second options of the first question.