

Macau University of Science and Technology
CS103
Software Engineering Practice D1

Library Seats Selection System based on Wechat Mini Program Report

Group Hello world

Report Status: <input type="checkbox"/> Draft <input checked="" type="checkbox"/> Published <input type="checkbox"/> Under Revision	Report ID:	Hello World_Project_Library Seats Selection	
	Version No:	3 rd Version	
	Author:	Wang Haocheng	18098537-i011-0112
		Wu Hanyu	1809853D-i011-0202
	Date:	25 th , Dec, 2021	



History of Versions

Version/Status	Author	Participant	Date Period	Comment
1	All Group	All	10.1-11.10	Only interface shown
2	All Group	All	11.10-11.22	Front-end Coding
3	All Group	All	11.22-12.15	Database Refinement

Contents

1 Requirement Document	4
1.1 Problem	4
1.2 Background Information	4
1.2.1 Domain Analysis document	4
1.2.2 Similar software	5
1.3 Background	7
1.4 Requirements acquisition	8
1.4.1 How to obtain requirements	8
1.4.2 Questionnaires	8
1.4.3 Virtual user idea	10
1.5 Requirement display	10
2 Design Document	11
2.1 Purpose	11
2.2 Modelling for student	11
2.2.1 Expand	11
2.3 Modelling for manager	12
2.3.1 Expand	13
2.4 State diagram	14
2.4.1 E-R diagram	14
2.5 Sequence diagram	17
2.5.1 User login	17
2.5.2 Seat select	18
2.5.3 Manage data	19
3 Development Document	20
3.1 Analysis of the characteristics of project development	20
3.2 Team member roles	21
3.3 Development Process	21
3.3.1 Epic story	21
3.3.2 User Story	22
3.3.3 Product Backlog	22

3.3.4	Stage 1 interface	23
3.3.5	Stage 2 Check the available seats	25
3.3.6	Stage 3 Registration and login	27
3.3.7	Stage 4 Seat reservation	33
3.3.8	Stage 5 Blacklist management	37
3.3.9	Stage 6 Management of seat	41
4	Testing Document	45
4.1	Black box Testing	45
4.1.1	Introduction of Blackbox	45
4.1.2	Boundary value analysis	46
4.1.3	Equivalence partition method	47
4.1.4	Cause and Effect Diagram	48
4.1.5	Decision Table	50
4.2	Function Tests Cases	51
4.3	White box Testing	54
4.3.1	Code	54
4.3.2	Control flow Chart	55
4.3.3	White Box Test Cases	55
4.4	Automatic Test	59
4.4.1	Jmeter Overview	59
4.4.2	Facing to Test	60
4.4.3	Test Environment	60
4.4.4	Testing Process	62
4.4.5	Test Design	64
4.4.6	Result	66
5	Acceptance Test	70
5.1	Key Program Simulation Steps	70
5.2	Database Interface	84
6	Management and Maintenance Document	86
6.1	Maintenance methods	86
6.2	Process of Maintenance	87
6.3	Timeline of Project	88

1 Requirement Document

1.1 Problem

Many students in Macau university of science and technology are worried about seats in library. Because there are many bad things in the library. A lot of students put their bags or books on the library tables last night, so that they can avoid queue for library tables next day. They reap convenience, but it's not fair for other students.

1.2 Background Information

1.2.1 Domain Analysis document

- Introduction: This app is aimed at university students. There is a great demand for this type of application from university students, but there is currently no suitable system to match.
- General knowledge about the domain: We need to know the number of students in all our schools. Because all students may enter the library to study, we need to consider the pressures that library seating faces at different times. Especially when holidays come, seats in library will become very scarce.
- Customers and users: In this domain, our target users are university students. With the student number increasing of our university, finding library seats on holiday are becoming more and more difficult.
- The environment: Our university's admissions line increases year by year. At the same time, the number of students studying in the library is also increasing.
- Test version: As the program needs to be tested for a long time before it is officially put into use. Through using tests by members of our group, we were able to identify some of the program's shortcomings. So our program has three versions.
- Similar software: Through our observation, we have not yet found software with similar functionality. In other words, we are the pioneer in this domain.

1.2.2 Similar software

Although the library function is available in our own software developed by the Macau University of Science and Technology. But, when I clicked on this library icon, I found that this page only had the function of borrowing books.

圖書館



Therefore, through our observation, we have not yet found software with similar functionality.

In other words, we are the pioneer in this domain. At the same time, there are not a few universities like ours that are expanding their student intake. So there is a huge market in this area.

What can be borrowed is the management locker rental system that already exists at the Macau University of Science and Technology, whose interface is shown in the figure below.
(page 6)

圖書館

••• X

Q 檢索圖書、期刊、報紙等資源

高級搜索

我的借閱

我的預約



您目前沒有借閱的圖書

掃碼借書

1.3 Background

2021年09月15日

澳科大喜迎2021/2022學年六千餘新生

澳門科技大學於9月15日全面恢復面授方式教學，並喜迎2021/2022學年六千餘新生。本學年澳科大迎來新生有逾3400名預/本科生及近2900名研究生。並於當日上午八時舉行新學年的升旗儀式。



澳科大舉行2021/2022學年升旗儀式

With the development of information technology, life in society is becoming easier and easier. But at the same time, people are becoming more and more impatient. This has led to many students being unable to study in their dormitories. So more and more students want to study in the library. In recent years, the number of students enrolled in our universities has increased. But the University's campus area and infrastructure has not been expanded. So the number of seats in the library appears to be significantly inadequate. How can we maximise the efficiency of the library's operation under the limited conditions? That requires a rational management of library seats. To improve efficiency, the first thing that needs to be addressed is the prolonged occupancy of seats in library. Many seats are occupied for long periods of time, but very often these seats are not being used by anyone. So, this is a waste of library resources.

1.4 Requirements acquisition

Firstly, we understand that requirements acquisition is essentially a communication between the software developers and the clients. Customers ask software design developers for features they want or for suggestions on certain aspects. But requirements acquisition is not about proposing a detailed solution, it is about describing the customer's needs. So, we should think about how to get what our users want.

1.4.1 How to obtain requirements

There are many ways to obtain requirements, such as seminars, face-to-face meetings between clients and developers, in-person visits and so on.

- Seminar
- Face to face meetings between clients and developers
- In-person visiting

But in reality, all of the above methods are difficult to achieve. So, the remaining viable methods are questionnaires and virtual user ideas.

1.4.2 Questionnaires

Questionnaires are more convenient and actionable than other methods. Through the Internet, questionnaires can be disseminated quickly and on a large scale. As a result, this survey method can gather a large amount of user feedback and suggestions for the software. Here's the picture of questionnaire from tencent. As you can see, it has helped its' customer returned 2.47 billion questionnaires. That's an amazing data. As we all know, our country's total population are 1.6billion. From this part, we can see the ability of questionnaire. It has excellent efficiency.



腾讯问卷目前已经为 1900万 用户回收 24.7亿 份问卷

However, questionnaires also have their limitations.

The options in the questionnaire also limit the subjectivity of the users.

1.4.3 Virtual user idea

We bring ourselves into the role of the user and see our software from the user's point of view. Then we propose optimisation requirements for the software. As the main target group of our project are university students, and we are university students ourselves, so we will have a better understanding of the needs of university students.

1.5 Requirement display

- Clear pages
- Easy to operate
- Seats need to be booked in advance
- Need to arrive on time and scan the QR code to lock in your seat
- When reserving a seat, the system should show the number of seats remaining in the library and their distribution
- Alert when a seat becomes available
- Estimated time that the seat will be available
- Students need to sign out when leaving the library

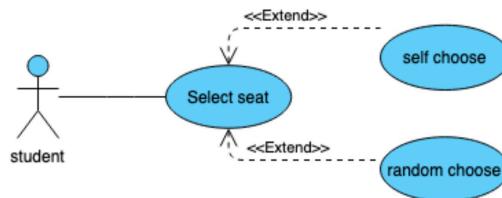
These needs were the result of our group's inquiry.

2 Design Document

2.1 Purpose

The purpose of this document is to describe the process of this program.

2.2 Modelling for student



Executor: Student

Pre-requisites: Students have logged into the library seat selection system via their username

Post-condition: Students complete their seat selection online

- Path:
- 1.Request to see all unselected seats
 - 2.System MUST-seat UI screen showing seating
 - 3.Students select their seats and confirm
 - 4.System returns results and notifies students
 - 5.System uploads and updates seat information

2.2.1 Expand

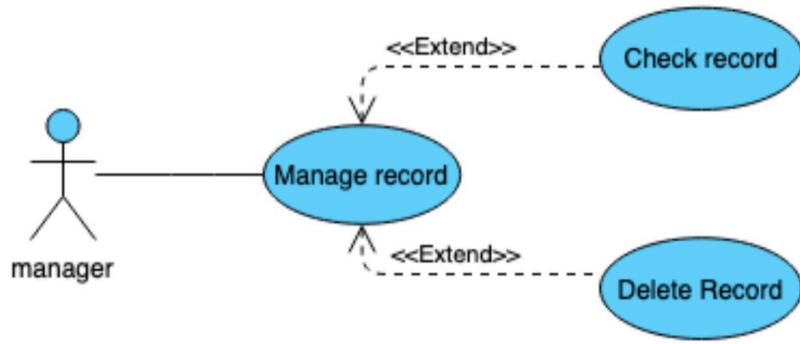
Random seat selection: 1.Students click on the random seat selection button

- 2.System Alert Results
- 3.Successful seat selection

4.The system indicates that the student has successfully selected a seat

Free seat selection : 1.Students choose an area by their own
2.Choose a seat of your choice in the selected area
3.Students confirm selection results
4.The system returns a confirmation result

2.3 Modelling for manager



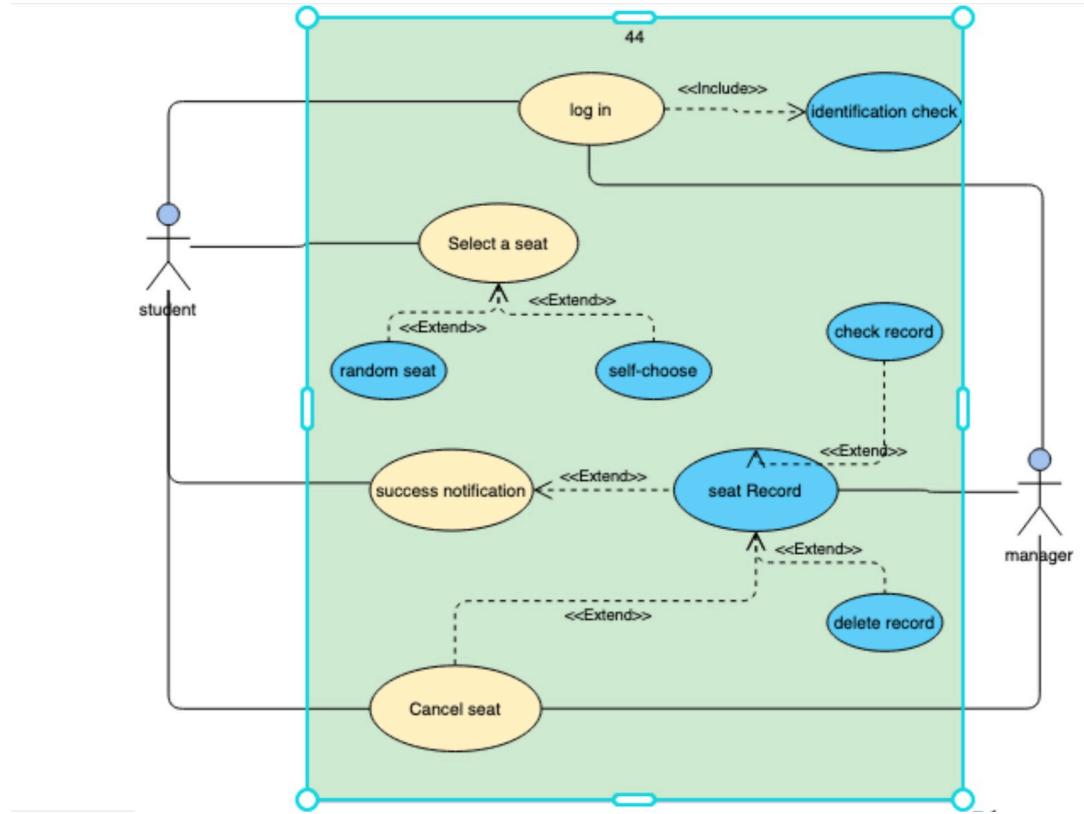
Executor: Administrators

Pre-requisites: Administrator login system to login page

Post-condition: Complete operations on relevant records

Path: 1.Administrator login system to login page
2.System display data
3. Administrators make enquiries about data
4. Administrators select data for editing
5. System update data

2.3.1 Expand



View records: 1. Administrator request to view data

2. Administrators view and update data

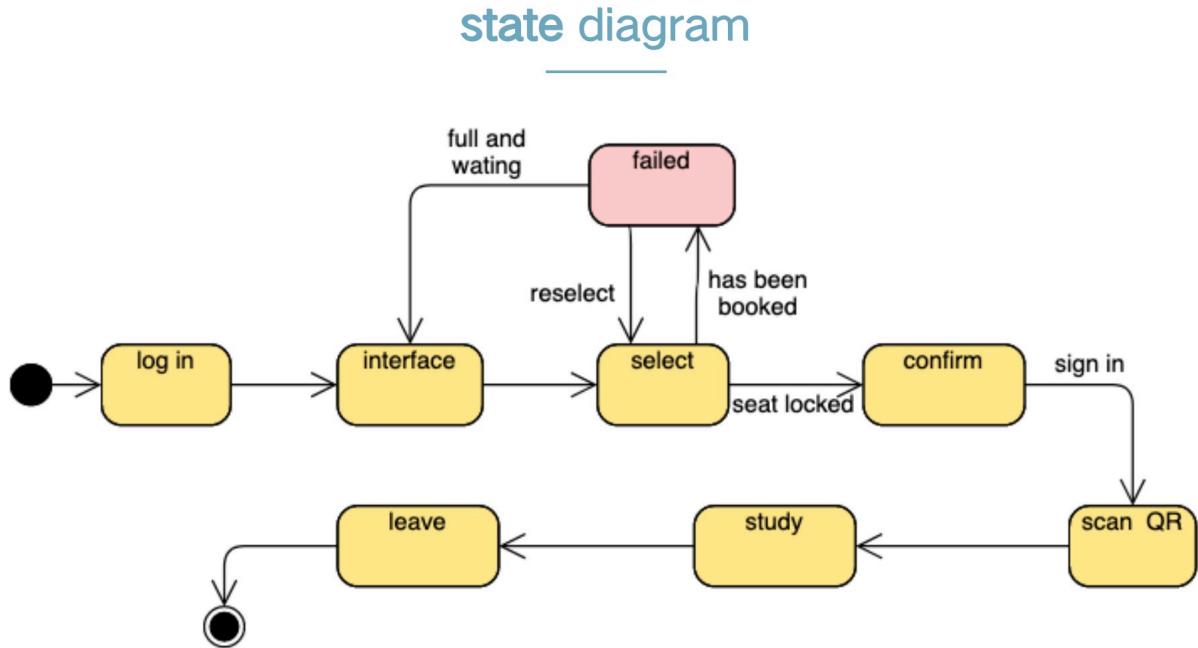
Delete records: 1. The administrator selects the data and chooses to delete it

2. The system prompts to confirm the delete option

3. Click confirm

4. The system indicates successful deletion

2.4 State diagram



This is our state diagram. Here's a brief introduction.

First, you need to log in. Then you will see the interface. So you start to select a seat. If you lock the seat successfully. Then you will confirm the seat. But if the seat has been booked, you need to reselect and you will get reminder "I am sorry the seats you select is full."

2.4.1 E-R diagram

The core of ER diagram contains 3 kinds of interrelated information: data objects, attributes of data objects and connections between data objects. Data objects are abstractions of complex information that must be understood by the software, which can be external entities, transactions, behaviors, etc. Attributes define the nature of the object. The way data objects are connected to each other, the links can be divided into three categories.

One-to-one connection (1:1)

For example, if a student can only choose one seat, then the seat and the student are one-to-one connections.

One-to-many connections (1:n)

For example, if an administrator has management of multiple seats, then the administrator and the seat are one-to-many connections.

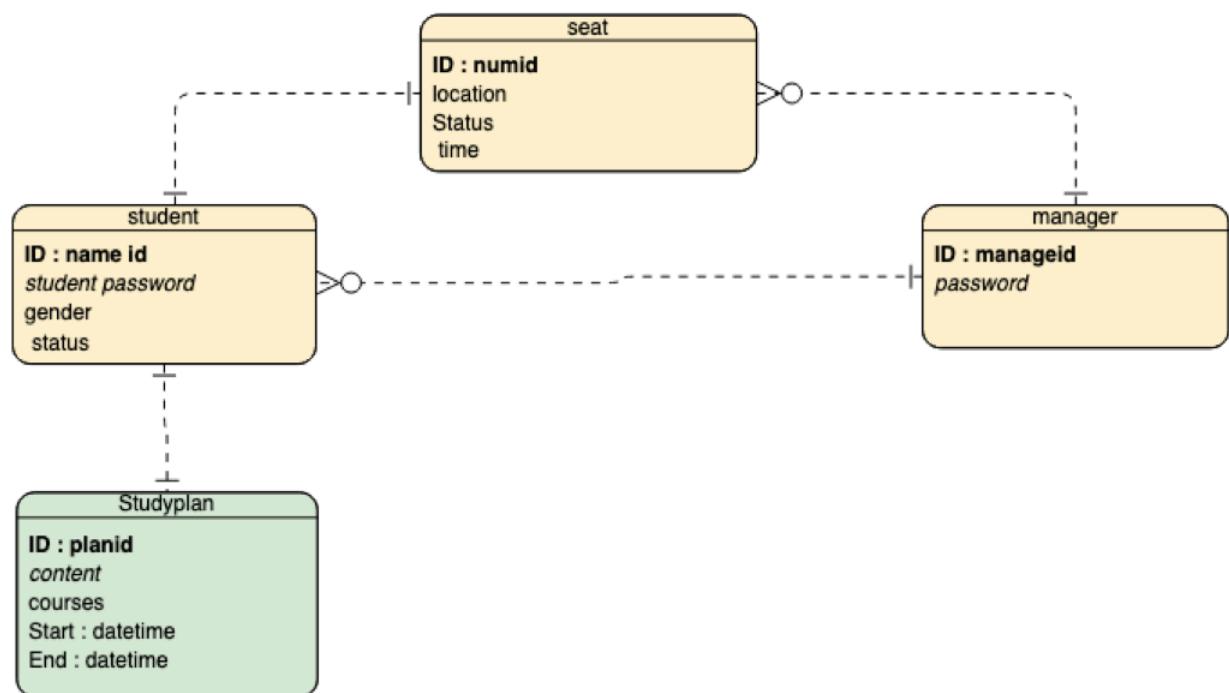
Many-to-many connection (m : n)

For example, a student can choose different floors and a floor can accommodate multiple students, then the student and the floor are many-to-many connections.

For students, in our system, a student can only choose one seat. But for administrators, one administrator can manage multiple seats. Also, in our database, one administrator can manage multiple students. In our initial concept, a student also has his or her own unique study plan.

The attributes of a student are mainly password, gender, reservation status, etc.; the attributes of a seat are mainly location (specific floor and coordinates), reservation time, etc.; the administrator mainly manages the password, etc.

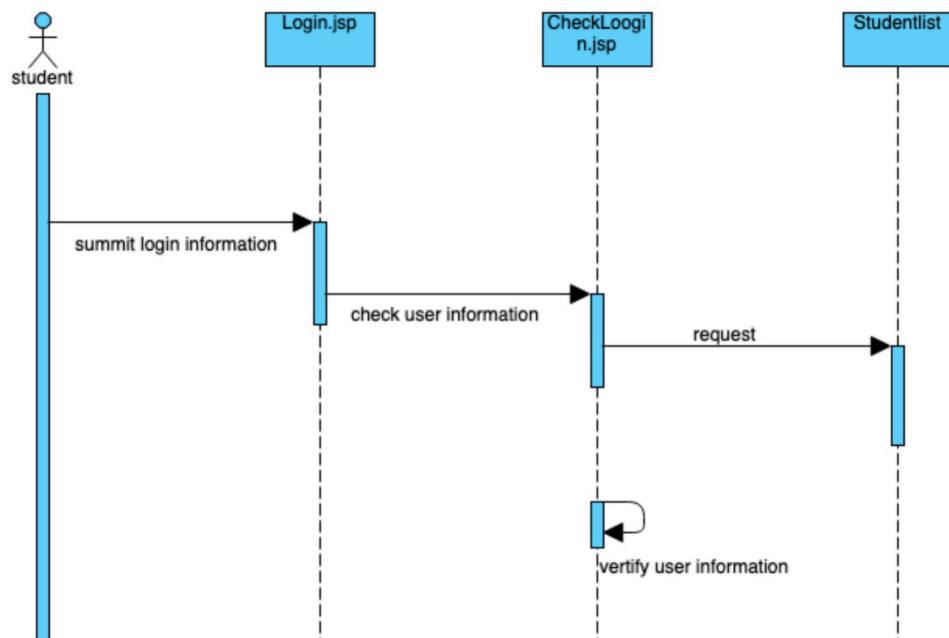
Entertainment Relationship Diagram



2.5 Sequence diagram

Here are the different sequence diagrams.

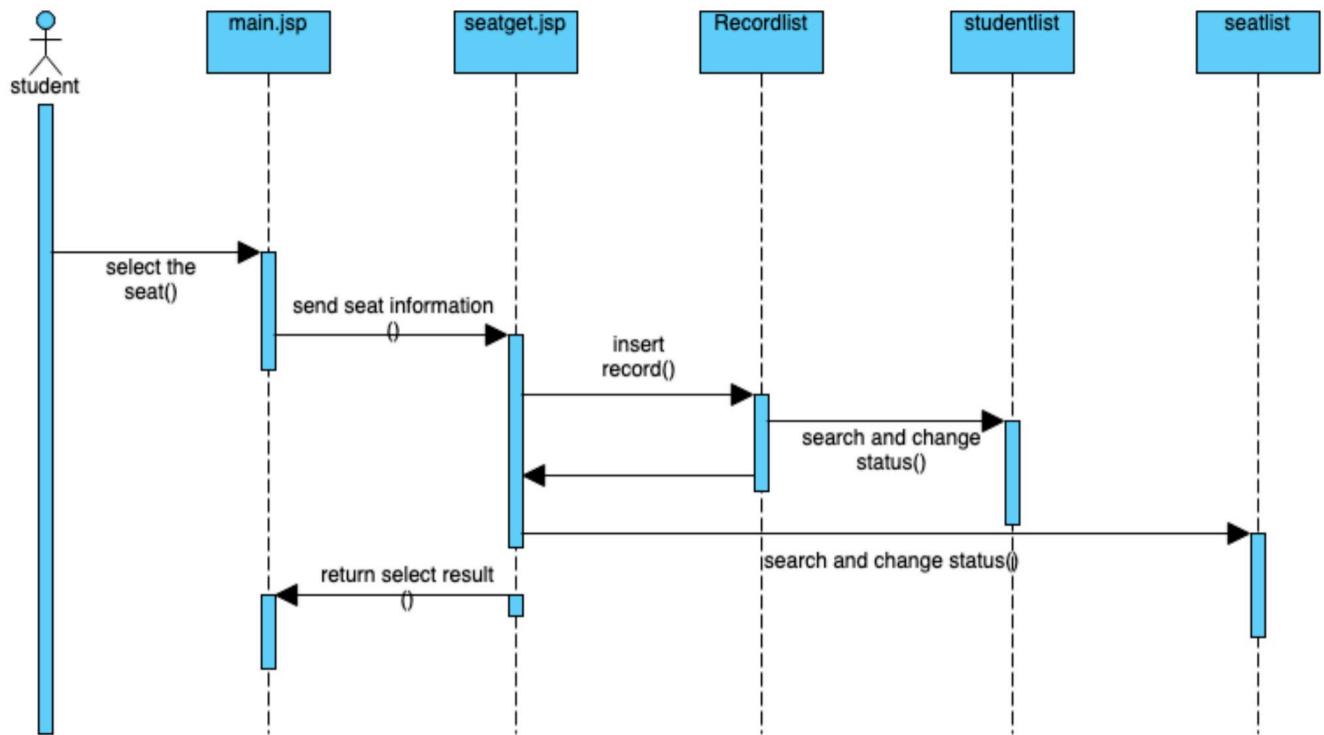
2.5.1 User login



This flowchart describes the user login process.

First user submit login information. Then the user check user information whether correct. If the information is correct, request to the Studentlist. But if the information is wrong, then the user need to verify user information.

2.5.2 Seat select

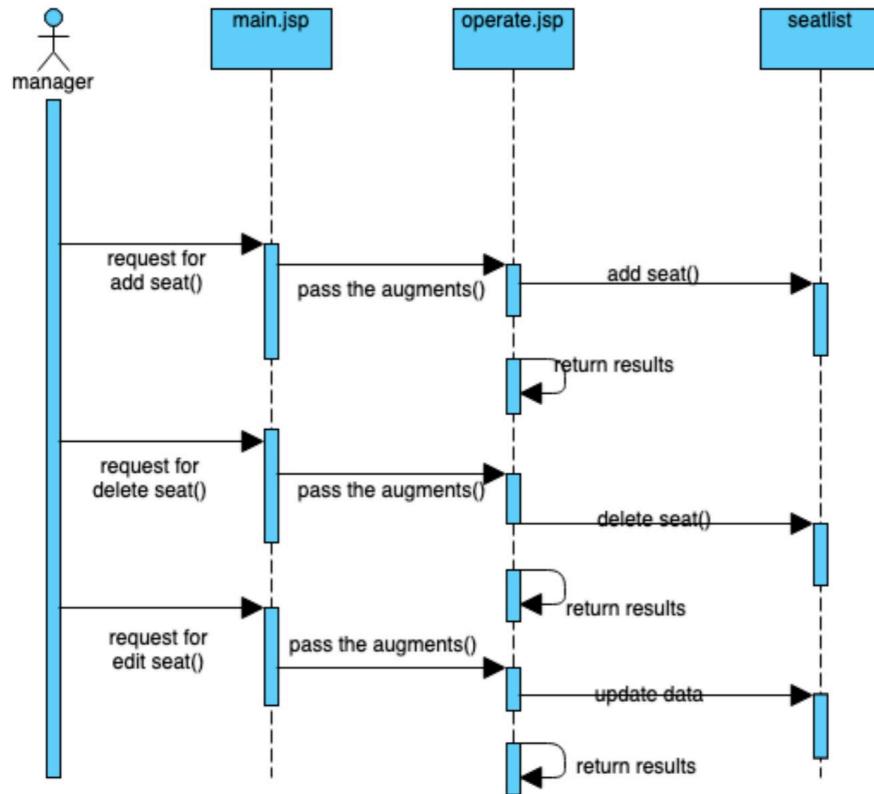


This flowchart describes the seat select process.

First, student need to select the seat. Then send the seat information. So the information will insert the record into the recordlist. But if the seat is not free.

Then student need to search and change the seat from seatlist. If the information record successfully, then return select result.

2.5.3 Manage data



This flowchart describes the manage data process.

If the manager request for add seat. Then he need to pass the augments. Send add seat to seatlist. After that, the relusts will return. If the manager request for delete seat. He also need to pass the augments. Then delete seat from seatlist. After that, the results will return. The last, if the manager request for edit seat. Then pass the augments. So he can update data to the seatlist. After that, the results will return.

3 Development Document

3.1 Analysis of the characteristics of project development

Our project is a Wechat mini program for school library seat selection, which has the characteristics of Internet product development, such as low version release cost, convenient use, rapid response to changes.

Traditional waterfall development:

The main problem with waterfall is that it reduces freedom due to its strict classification. Commitments made early in the project make it difficult and costly to adjust to changes in later requirements. A waterfall approach is generally not feasible in situations where requirements are unclear and can change over the course of a project.

Iterative development:

Iterative development, also known as iterative incremental development or iterative evolutionary development, is a software development process opposite to the traditional waterfall development, which makes up for some weaknesses in the traditional development method and has a higher success rate and productivity. In the iterative approach, the entire development operation is arranged as a series of small projects, known as a series of iterations. Each iteration includes requirements analysis, planning, completion, and testing.

Our product development method(Iterative development):

Development operations can be initiated before requirements are fully validated and parts of the system can be refined in an iteration. After the customer's response to refine the requirements, and start a new round of iteration. The advantages of this approach include

1. Reduce risk
2. Get early user reactions
3. Continue testing and integration
4. Use changes
5. Improve reusability

We also went through multiple iterations during the development process. The final version is the third version we've developed. As an official release our version number will also be named V1.0.0.

3.2 Team member roles

Product Owner: Wang Haocheng

closely partners with the business and the team to ensure everyone understands the work items in the product backlog

gives the team clear guidance on which features to deliver next

Ensure that the team is fully functional and efficient Acceptance development results

Development Team:Wu Hanyu, Gong Tao, Que Xuanyu, Pu Anyu, Mao Zunjie Cooperate to complete the development work

Understand user needs

Ensure the success of every development

Actively participate development planning, standup, development review, and the development retrospective

3.3 Development Process

3.3.1 Epic story

Prior to development, we discussed the details of the product in detail, simulating the needs of the product from a platform administrator and user perspective.

Administrator:

As an administrator, I want to be able to check the user's feedback on the small program and manage the blacklisted users.

User:

As a user, I hope this small program has a simple interface and simple operation, which can clearly show the remaining space on each floor of the library. And there are new seats are able to send a notice to remind advance booking seats.

3.3.2 User Story

To further target the needs of our users, we conducted a questionnaire survey of our students, further breaking down the epic story into smaller user stories so that we could develop a development plan for each stage.

Registration and login: in order to make the product more accurate service for students in school, our products can only be registered and logged in through students' student number .

Interface:As users, students hope that small programs can have a simple operation interface to facilitate operation.

Check the available seats: As users, most students hope to see the available seats on each floor of the library in real time, so as to reduce the time spent in searching for seats.

Seat reservation: As users, some students hope to reserve seats, which can improve the utilization rate of seats.

Notification: As a user, many students hope to be notified when other students leave the new seat, so that students who have not booked a seat can make a reservation as soon as possible.

Blacklist management: As an administrator, I hope to punish users who do not comply with the reservation rules, such as: after reserving seats, one person occupies more than one seat, and pull them into the blacklist.

Review user feedback: As administrators, we hope to see user feedback, further understand user needs and feedback, to improve our products.

Management of seat:As the administrator, I hope to manage whether the seats can be booked. For example, if there is a typhoon tomorrow, I can set the whole day tomorrow as unbooked.

3.3.3 Product Backlog

We have the initial product backlog based on the user story above.

标题	优先级	描述
interface	最高	As users, students hope that small pro...
Registration and login:	最高	in order to make the product more acc...
Seat reservation:	较高	As users, some students hope to reser...
Check the available seats:	较高	As users, most students hope to see th...
Management of seat	普通	As the administrator, I hope to manage...
Blacklist management:	普通	As an administrator, I hope to punish u...

3.3.4 Stage 1 interface

Development plan: In the production process we started to design the front end first, we chose to design the UI first. In terms of UI design, we give priority to simplicity and only keep necessary words and ICONS, so that users can operate easily and quickly. Then we set about writing a framework. In the course of learning online, we found a large number of completed frameworks on the web. But none of them fit perfectly with the layout of our school library. These templates are obviously not in line with our expectations. And the number of layers of the template framework is fixed, which can not be customized according to our needs. It is also inevitable that post-production will have problems corresponding to the back-end interface. So we decided to look at these different templates to see what some of the code could do. To reinvent the framework of our program.

Duration of this stage: 2 week

Task:

Preliminary design of each part of the interface

1. Login user interface
2. User interface
3. Interfaces of each floor
4. Reservation interface

How to achieve these tasks: Search the web for ICONS and pictures. Use Java in wechat developer tools to turn them into images or buttons for display in UI.

```
<view>
    <image class="logo" src="/images/icons/logo.png" ></image>
    <form class='login-form' bindsubmit="login">
        <view class="weui-cell weui-cell_input">
            <view class="weui-cell__hd">
                <view id="users" class="weui-label">用户名</view>
            </view>
            <view class="weui-cell__bd">
                <input class="weui-input" placeholder="学号、教工号、手机号"
                    name="username"/>
            </view>
        </view>
        <view class="weui-cell weui-cell_input">
            <view class="weui-label">密码</view>
            <input password type="text" placeholder="密码" name="password"/>
        </view>

        <view class="button-sp-area">
            <button class="weui-btn2" form-type="submit" type="primary" plain="true" >
                登录</button>
        </view>
    </form>
</view>
```

Show our work: Here we show the user login interface. The figure shows the information required for user login and the time period available for booking in the library

Other UI designs will be shown along with features in subsequent introductions.

Feedback:

After user experience, we hope to optimize font, layout and picture proportion to make the interface more beautiful.

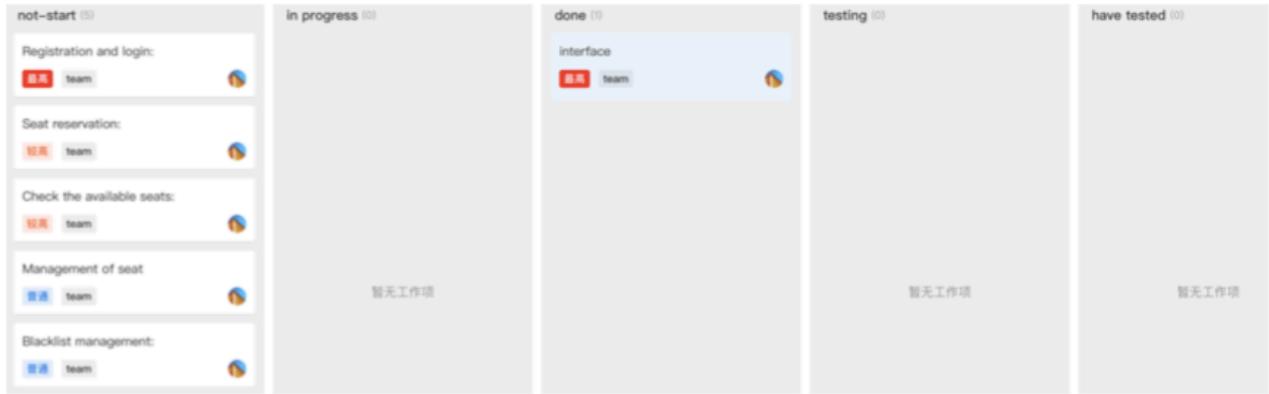
Summary:



UI design is relatively uncomplicated, but it's a lot of work. Here we have basically achieved our initial UI requirements, which are clean and uncluttered. But there is still room for improvement in the overall visual aesthetic.

3.3.5 Stage 2 Check the available seats

Development plan: With a preliminary framework in place, we set about implementing the various functions. The next step is to select the library and floor screen. Show the user the availability of seats on each floor and whether reservations are available. After our investigation, it is as authentic as possible according to the specific situation of our school library and



the number of seats on each floor.

Duration of this stage:

2 week

Task:

1.Show the available seats on each floor of the library. 2.Indicates whether a seat is available for reservation.

How to achieve these tasks:

Investigate the distribution and number of seats on each floor of the school library.We write a structure for each seat in the form of a seat number, and then determine whether the seat is available by determining the seat True or False, and determine how long the seat will be released after the reservation by means of a timer.

Show our work:

Figure 6 shows the interface for selecting the floor of the library or study room , and Figure 7 shows the remaining seats of that floor after selecting the floor.Users can view the available seats on different floors at different time periods

	Id	Building	Floor	Seat Type	Status	open_time
<input type="checkbox"/>	1	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	2	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	3	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	4	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	5	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	6	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	7	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	8	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	9	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	10	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	11	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	12	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	13	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	14	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	15	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	16	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	17	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	18	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	19	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	20	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	21	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	22	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	23	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	24	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	25	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	26	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	27	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	28	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input type="checkbox"/>	29	麦积区图书馆	1F	麦积区	已预约	2021-12-23
<input checked="" type="checkbox"/>	30	麦积区图书馆	1F	麦积区	已预约	2021-12-23

Feedback:

The function works normally, but some users report that it is difficult to find the exact position of the corresponding seat in the library only by the seat number.

Summary:

Through statistics, it can display the seats of each floor of the library, and also show whether each seat can be reserved. However, it is still difficult to achieve accurate positioning of various locations, and it is difficult and time-consuming to implement this feature. If we have time left at the end, maybe we can try to perfect it.

3.3.6 Stage 3 Registration and login

Development plan: The next important module is to implement the login function. The first is that students need to register an account. At the beginning, we also considered using wechat to log in. However, because we hope that the small program can serve the students in school

```

var app = getApp();
var util = require(' ../../../../utils/util.js');
Page({
  data: {
    date: [util.formatTime(new Date()).split(' ')[0],util.nextDate(new Date())],
    building: ['全部','图书馆','自习室'],
    floor: [['全部'],['全部','1F','2F','3F','4F'], ['1F']],
    type: [[[['全部']],[['全部','mac座位','电脑座位']],[['全部','数据库电脑座位','普通电脑座位','电源座位','普通座位']],[['全部','单独座位','电源座位','普通座位']],[['全部','电脑座位','电源座位','普通座位区']],[['全部','电脑座位区','电源座位区','普通座位区','单独座位区']]],[['全部']],[['全部','1F']],[['全部','2F']],[['全部','3F']],[['全部','4F']]],
    buildingnum:0,
    floornum:0,
    typenum:0,
    Datenum:0
  },
  //用户点击更改教学楼
  bindPickerChange: function (e) {
    this.setData({
      buildingnum: e.detail.value,
      floornum:0,
      typenum:0
    })
  }
},

```

more accurately, we still think it is more safe to register in the form of student number. This makes it easier to manage the database in the background, and it is also easier to connect to the school system.

Duration of this stage: 4 week

Task:

1. Realize the registration and login functions of users and administrators.

How to achieve these tasks:

Most of the small programs we usually use use wechat one-click login and registration. Relying on the wechat platform, there is a special function in the small program developer tool that can directly access the local wechat account data to achieve login. We adopted this feature at the beginning of the project. But then there is the problem of data interaction between different users. We didn't know at the beginning that the wechat applets developer tool didn't have a back-end database. Therefore, only the front end can not be realized. For example, after student A has reserved A seat, student B can not see that student A has reserved A seat. Such procedures are certainly incomplete. And we also want to implement more functions in

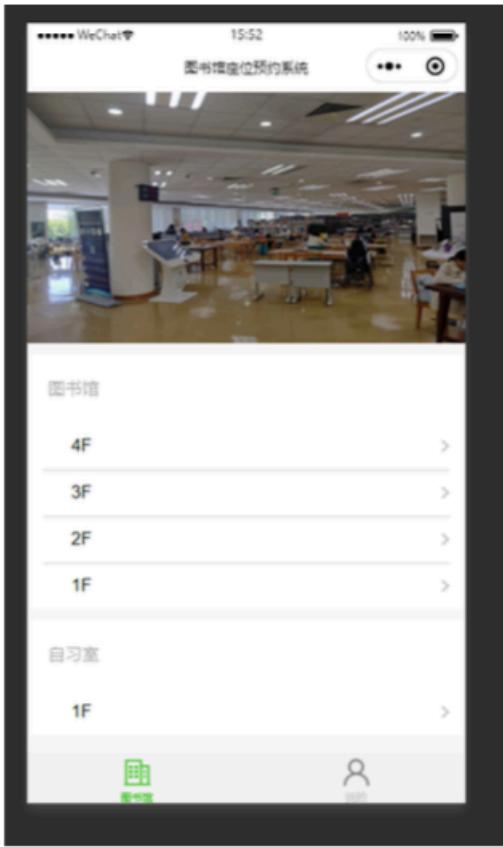


Figure 6

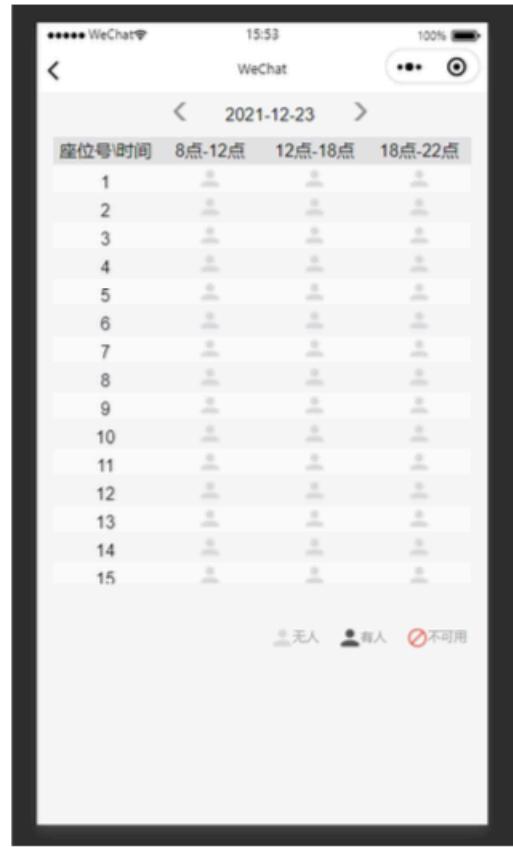
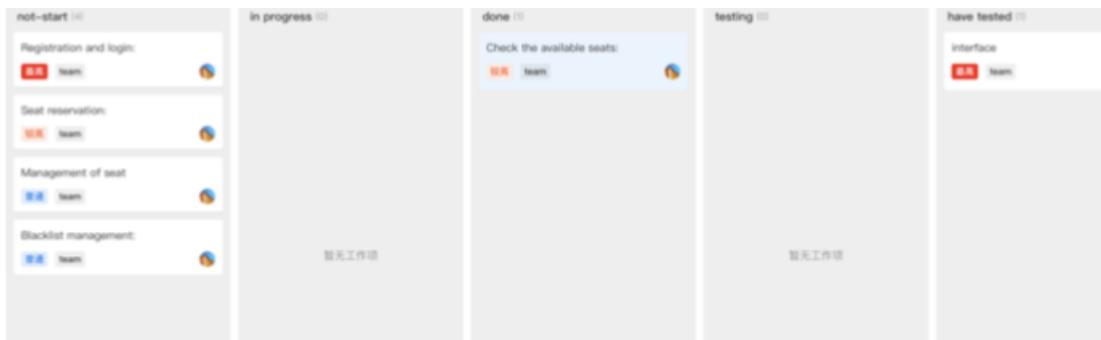


Figure 7

the future is not able to do. So we started learning about back-end and database implementations. We made a new registration system during the second improvement development phase of the project. Only wechat small program developer tool as the front-end development, using Wampsever as the back-end management tool. Our account has two types, one is the ordinary student account, the other is the administrator account. A normal student account allows for normal seat reservations, including modifications to personal information. The administrator account can be modified to add a student account and serve as an interface to manage the back-end database. Apache is used as the Web server software. Run in the server with PHP, call the local database mysql. It took us a lot longer than we expected, but we managed to make it work.

show our work:



```

        icon:'error'
    })
} else{
    if(e.detail.value.pwd != e.detail.value.repwd){
        wx.showToast({
            title: '两次密码不一致',
            icon:'error'
        })
    }else{
        wx.request({
            url: 'http://127.0.0.1/setuser.php',
            data:{
                username:e.detail.value.username,
                password:e.detail.value.pwd
            },
            success:(res)=>{
                // console.log(res);
                wx.showModal({
                    cancelColor: 'cancelColor',
                    showCancel:false,
                    content:res.data
                })
            }
        })
    }
}

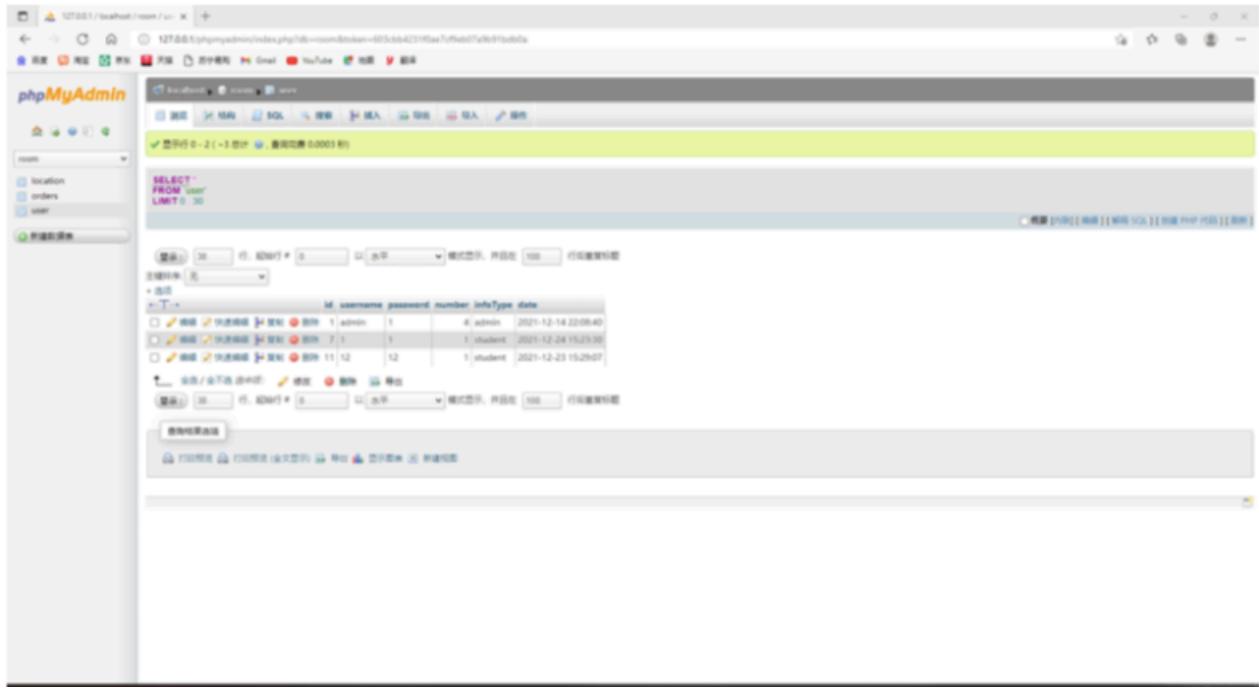
```

We registered an administrator account and two user accounts for testing.

Feedback:

In the first version, due to the lack of back-end interaction, the user's reservation can only stay local and cannot be seen by other users. In subsequent releases, interactions with the back-end were added so that each user's reservation could be synchronized.

Summary:



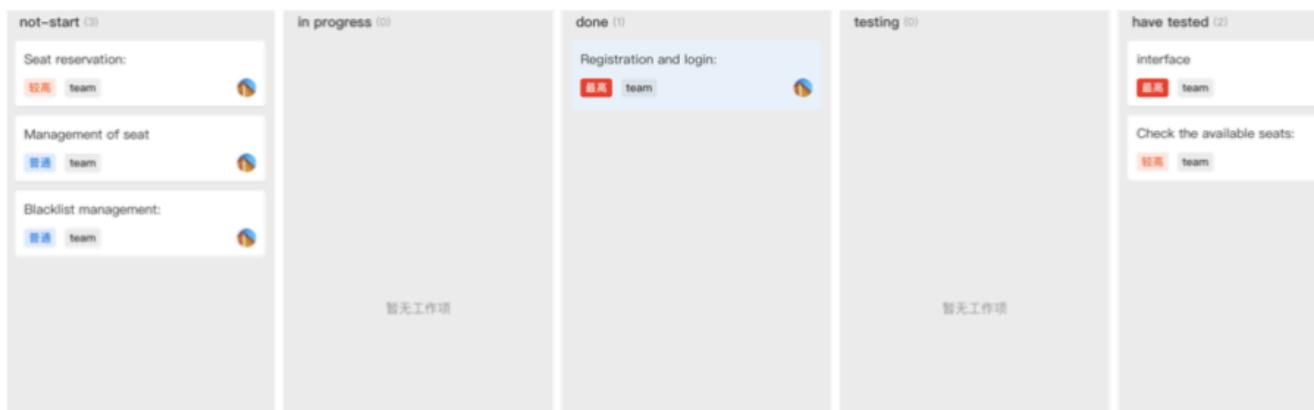
Students can register through the student number, at the same time, after the addition of the back-end, the small program can synchronize each user's appointment so that it is known by other users. In this version has basically completed the administrator and ordinary user registration login function.



Figure 10: Administrator interface



Figure 11: User interface



3.3.7 Stage 4 Seat reservation

Development plan: After completing the initial steps, this step will fulfill the purpose of our little program, which is to allow students to reserve seats in the school library without wasting time looking for them.

Duration of this stage: 2 week

Task:

1. Realize the function of reserving seats for users.
2. Lock the reserved seat.
3. Cancel seat reservation

How to achieve these tasks:

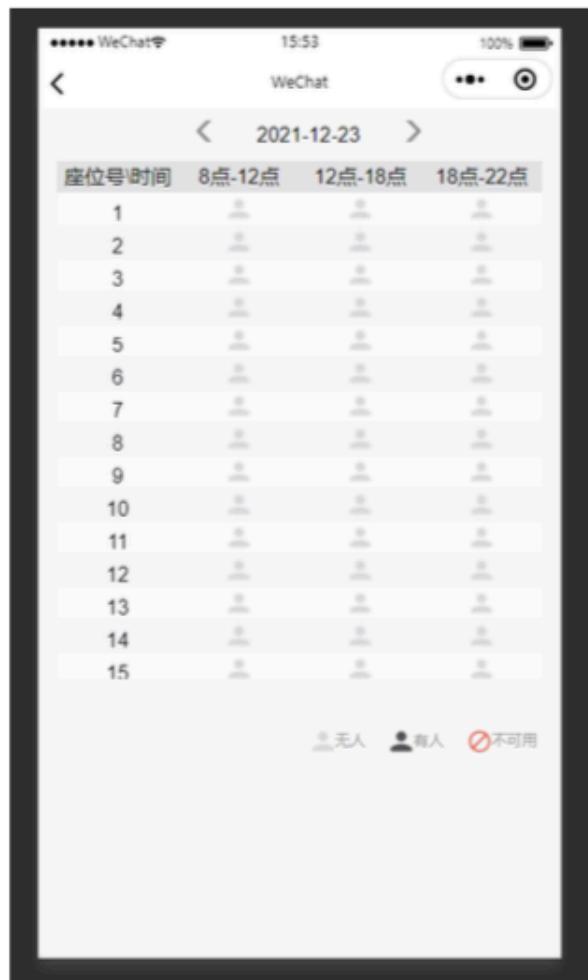
Learn the Related Java language by learning the code of other related programs and understanding the purpose of each piece of code. Finally in wechat developer tools will be implemented.

Show our work:

After the user clicks on the selected seat, the applet asks if it is sure to reserve that seat. Click OK and the reservation is successful. Shows that the seat will not be reserved by other users. The user cannot reserve other seats in the same time segment. Figures 14,15,16, 17 show this process.

```
//用户点击右日期
nextday() {
    this.setData({
        date: util.nextDate(new Date())
    })
    this.onShow();
},
//预定座位信息页面
Y: function (e) {
    wx.showModal({
        title: '提示',
        content: "确定预约这个座位?",
        success: (res) => {
            if (res.confirm) {
                wx.request({
                    url: 'http://127.0.0.1/order.php',
                    data: {
                        sit_index: e.currentTarget.dataset.item_index,
                        time_index: e.currentTarget.dataset.table_index,
                        building: app.globalData.building,
                        floor: app.globalData.floor,
                        sittype: app.globalData.type,
                        username: app.globalData.username,
                        date: this.data.date
                    }
                })
            }
        }
    })
},
```

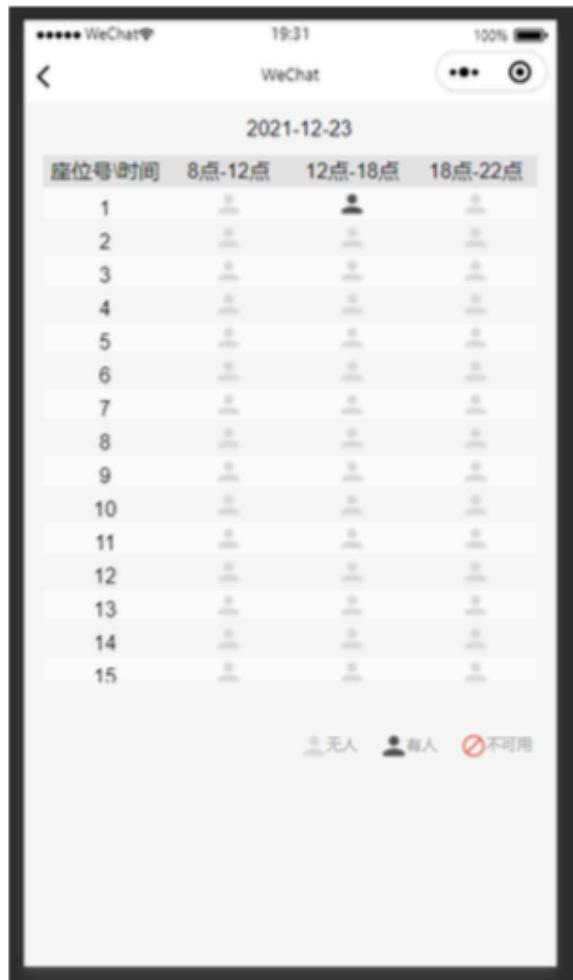
```
//座位已被预约
N: function (e) {
    console.log(e.currentTarget.dataset);
    wx.showModal({
        content: '该座位已被预约，下次趁早~(*￣▽￣*)o',
        showCancel: false,
    })
    this.onShow();
},
```



Figures14



Figures15



Figures16



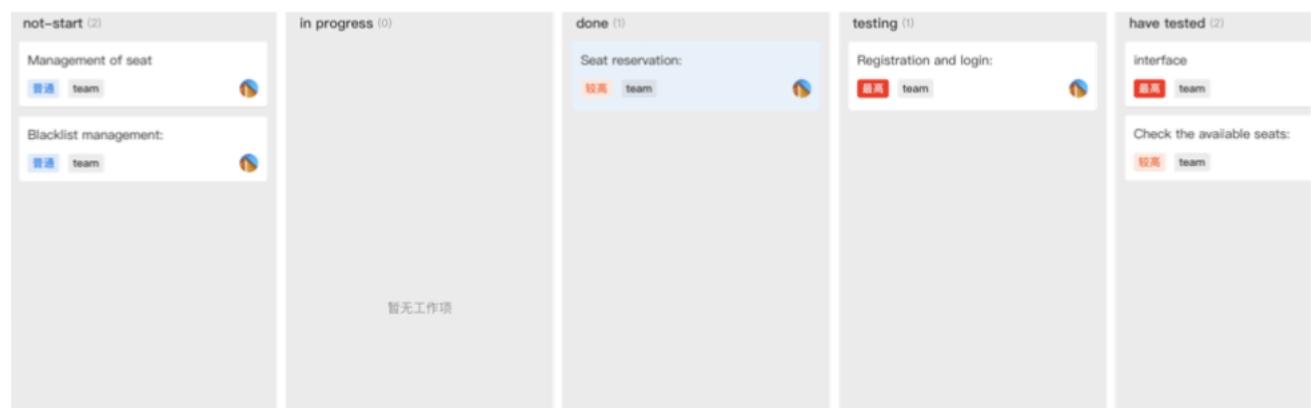
Figures17

Feedback:

The problem with the lack of a back end in the second release is explained in 3.3.6. With the latest version having a back end, the team was generally satisfied with what it could do.

Summary:

After making the backend, not only the user registration experience is improved, but also the reservation function can be realized. Users' appointments can now be seen by other users. A reserved seat will not be rereserved unless the person making the reservation cancels it. Anyway, our main goal, booking a seat, has been achieved. In the future, we will improve other functions of the small program, so that users can have a better use experience.



3.3.8 Stage 5 Blacklist management

Development plan: Because in the questionnaire survey, some students pointed out that sometimes someone occupies a seat in the library for a long time, but the person does not come for a long time. So we decided to create a blacklist feature for administrators. When a student finds that a reserved seat is not occupied for a long time, he can submit a report to the administrator. The administrator can blacklist the reported user, which prevents the user from using the reservation function.

Duration of this stage: 1.5 week

Task:

1. Blacklist used by the administrator

- 2.Blacklist users cannot use the reservation function
- 3.The user reservation function is restored after the masking period expires

How to achieve these tasks:

After being shielded, the user will be pulled into the database and carry a specific identification. When the user clicks the jump interface of seat, the program will judge whether the user carries the identification. If the identification is carried, the corresponding pop-up window will pop up, so that the user cannot continue to make the next appointment. If not, you can enter the reservation interface. And we also wrote a piece of code so that the program can calculate the user was banned time, and after reaching a certain banned time to remove the user's banned, restore the normal use of its reservation function.

```
N: function (e) {
  wx.showModal({
    cancelColor: 'cancelColor',
    content: '是否将该同学拉黑一天? ',
    success: (res) => {
      if (res.confirm) {
        wx.request({
          url: 'http://127.0.0.1/limit.php',
          data: {
            building: this.data.building[this.data.buildingnum],
            floor: this.data.floor[this.data.buildingnum][this.
data.floornum],
            sitstype: this.data.type[this.data.buildingnum][this.
data.floornum][this.data.typenum],
            time_index: this.data.time_index,
            sit_index: e.currentTarget.dataset.seatnum,
            date: util.formatTime(new Date()).split(' ')[0],
            nextdate: util.nextDate(new Date()),
            limit_date: util.limit_date()
          },
          success: (e) => {
            console.log(e.data);
            this.onShow();
          }
        })
      }
    }
  })
}
```

```
cancelColor: 'cancelColor',
content: '您当前已被管理员拉黑，解封时间为' + app.globalData.userdate
showCancel:false
```

```
function choose_time_index(date) {
    var now = new Date(); //当前日期
    var pass = new Date(date); //传来的日期
    var subtract = now - pass;
    if(subtract>=0){
        subtract = 1;
    }else{
        subtract = 0;
    }
    return subtract
}
```

Show our work:

Figure 21 and 22 respectively show the interface where the administrator blocks the user reserving a certain seat and the popup window when the blocked user enters the reservation interface.



Figure 21



Figure 22

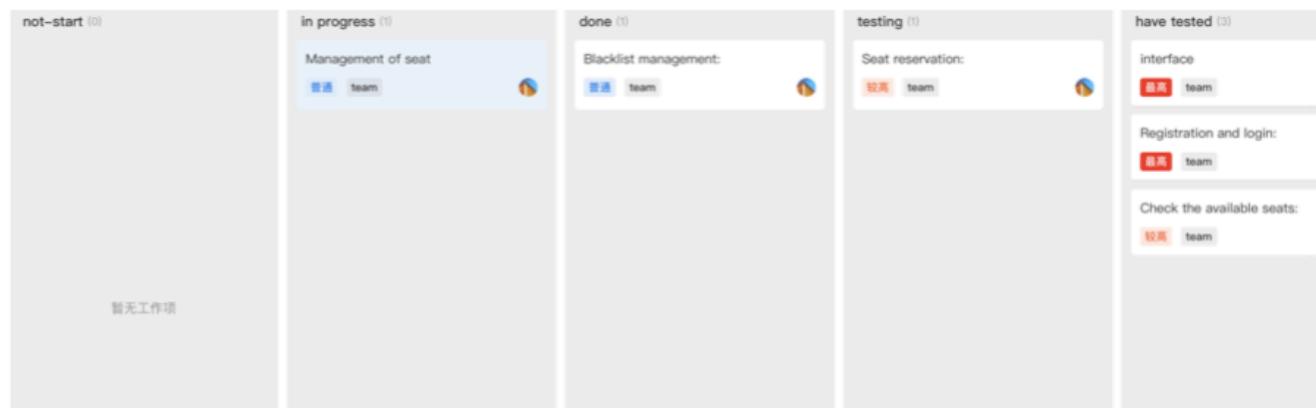
Feedback:

Both the user and our team believe that this function is necessary, which can effectively prevent some students from not coming after reserving seats to a certain extent, resulting in

the waste of seat resources.

Summary:

At this stage, we added a new function named blacklist for our program, which can make the blocked users unable to use the reservation function and judge the expiration time of the ban. This function can reduce the malicious occupying of seats in the library and improve the utilization rate of seats in the library.



3.3.9 Stage 6 Management of seat

Development plan: Enables administrators to manage all seats and make all or some seats unavailable during holidays, inclement weather, library closures, or maintenance of certain areas.

Duration of this stage: 1.5 week

Task:

- 1.The administrator can turn off/on the reservation function of some or all seats

How to achieve these tasks:

Make it unreservable by popup prompt and changing the seat status of the whole floor to False. At the same time, the icon of the seat has also changed into a non-reservation sign.

```
var message = "";
if(this.data.building[this.data.buildingnum] == "全部"){
    message = "是否要设置所有教学楼不可预约? ";
}else if(this.data.floor[this.data.buildingnum][this.data.floornum] == "全部"){
    message = "是否要设置" + this.data.building[this.data.buildingnum] + "所有楼层不可预约? ";
}else if(this.data.type[this.data.buildingnum][this.data.floornum][this.data.typenum] == "全部"){
    message = "是否要设置" + this.data.building[this.data.buildingnum] + this.data.floor[this.data.buildingnum][this.data.floornum] + "所有教室不可预约? ";
}else{
    message = "是否要设置" + this.data.building[this.data.buildingnum] + this.data.floor[this.data.buildingnum][this.data.floornum] + this.data.type[this.data.buildingnum][this.data.floornum][this.data.typenum] + "不可预约? ";
}
//展示提示信息
wx.showModal({
    cancelColor: 'cancelColor',
    content:message,
    success:(e)=>{
```

Show our work:

Figure 24 shows the interface when the administrator opens/closes the library seats. Figure 25 shows the seat icon and popup prompt the user sees after the seat is closed.

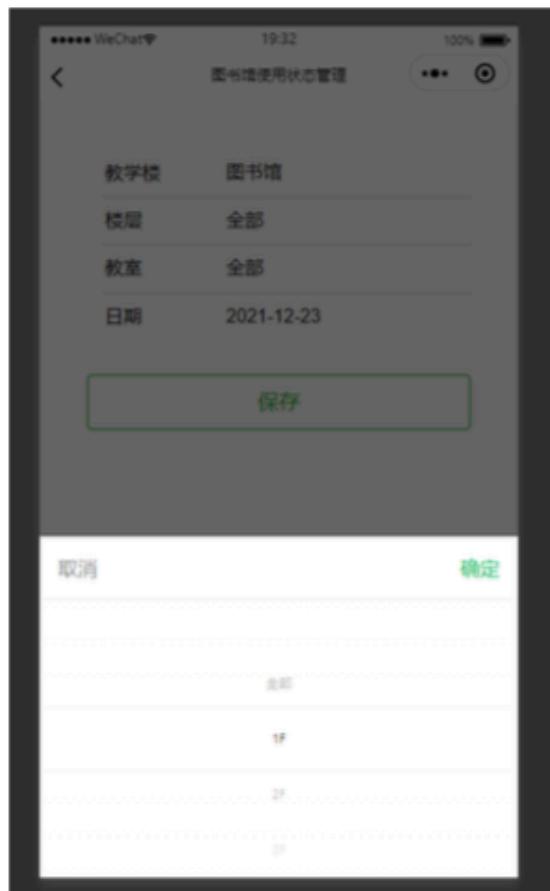


Figure 24



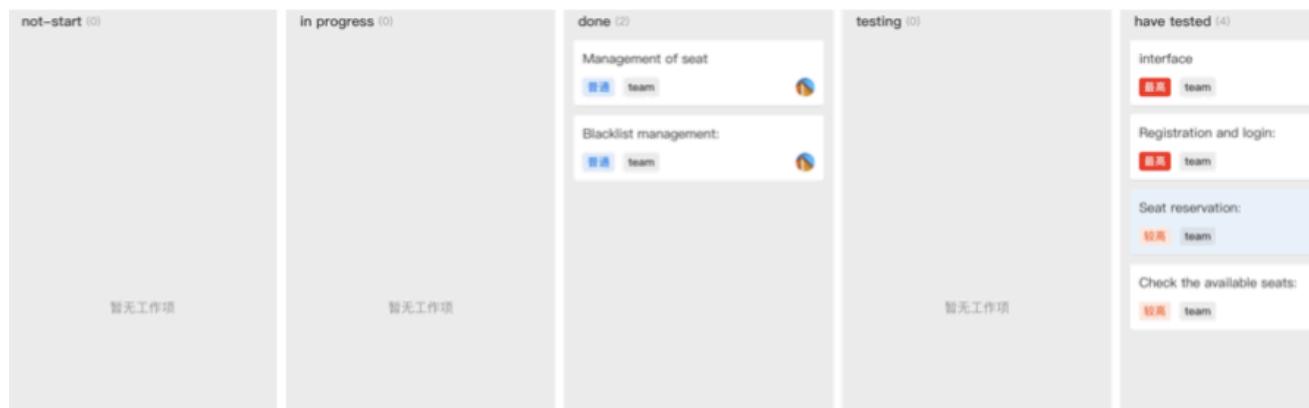
Figure 25

Feedback:

This feature is not widely used, but we still think it is necessary. It can make the fact that the library is closed known to students who have not received the notice. So that they don't have to waste time going back and forth because they don't know the closure information.

Summary:

After phase 4 was completed, we had implemented most of the functionality required by the requirements we captured. At the same time the program can also be fast, stable operation. In stages 5 and 6, we gave it more functionality. So that students can get a better reservation experience, administrators can better manage the program, the library seats can have a higher utilization rate.



4 Testing Document

4.1 Black box Testing

4.1.1 Introduction of Blackbox

Black box testing, as the name suggests, is a black box where the product to be tested is a black box, you only see the inputs and outputs, you do not see the logic implemented inside the box. Typically, this part is also a very important part of software development. Specifically, black box testing is also called functional testing, where you test to see if each function works properly. In testing, the program is seen as a black box that cannot be opened, and the program interfaces are tested without considering the internal structure and internal characteristics of the program at all, only checking that the program functions are working properly according to the requirements specification and that the program can properly receive the input data and produce the correct output information.

Black box testing is triggered by the relationship between input and output data from the user's point of view and attempts to find the following types of errors.



- (1) Incorrect or missing functionality
- (2) Interface errors
- (3) Database access errors
- (4) Performance errors
- (5) Initialisation and termination errors

and use case design methods include:

- (1) Equivalence class delineation
- (2) Boundary value analysis
- (3) Decision tables
- (4) Cause and effect diagrams
- (5) scenario approach

4.1.2 Boundary value analysis

Demonstration

Boundary value analysis is designed as a complement to equivalence class classification by selecting the boundary values of the equivalence classes as test cases. If the input conditions specify a range of values, the values that just reach the boundary of this range and the values that just exceed the boundary of this range should be selected as test input data. If the input conditions specify the number of values, the maximum number, the minimum number, one less than the minimum number and one more than the maximum number are used as test data. If the specification gives an input field or an output field. If the program uses an internal data structure, then the values on the boundaries of the internal data structure should be selected as test cases.

Generally it includes:

1. General boundary value analysis: generally taken as **Min**, **Min+**, **Normal**, **Max-**, **Max**.
2. Robustness boundary value analysis: in addition to the general boundary value analysis, **Min-**, **Max+** are also included.

why we use

The equivalence class division ignores certain specific types of efficient test cases, and boundary value analysis can fill in many of these gaps; according to a large number of test statistics,

many errors in programming occur at the boundaries of the input definition domain or output value domain, rather than in the middle of the input/output range. Therefore designing test cases for the boundary cases of the input and output equivalence classes can detect more errors and have a higher test return; boundary value data is essentially in the scope of a certain equivalence class and is indeed a redundancy (repetition) when testing, but for better test quality (boundary values are particularly bug-prone), boundary values have to be measured separately and an appropriate redundancy is acceptable.

4.1.3 Equivalence partition method

Equivalence Partitioning Method is also known as Equivalence class partitioning (ECP). It is a software testing technique or black-box testing that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that might require many arbitrary test cases to be executed before general error is observed.

In equivalence partitioning, equivalence classes are evaluated for given input conditions. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describes set of valid or invalid states. The simple diagram we have made here is shown in the diagram.

input Condition	Valid Equivalence class	invalid Equivalence class
x	integers	NOT integers
	positive	NOT positive

Boundary Value Chosen

The boundary values in our subject are then dominated by the number of seats.

Usually we need to pay attention to the following points.

- (1) If the input condition specifies a range of input values, then test cases should be designed for the boundaries of the range and invalid input test cases for situations where the boundaries have just been crossed.
- (2) if the input condition specifies a number of input values, then test cases should be designed for the minimum number of input values, the maximum number of input values, and for the case where there is one less than the minimum number and one more than the maximum number.
- (3) If the input or output of the program is an ordered sequence (e.g. a sequential file, a linear list or a table), special attention should be paid to the first and last elements of the sequence.
- (4) If an internal data structure is used in the program, the values on the boundaries of this internal data structure should be selected as the test case.

The important difference between the boundary value analysis approach and equivalence class division is that boundary value analysis examines states that are at or near the boundary of the equivalence division. We counted MAX seats available in the library

here is the example we made

Input	Effective value	Invalid Value
password	123456 (cover 1 3)	12356.3(cover 2 4)
user name	18098531001 (cover 1 3)	NG .2022(cover 2 4)

4.1.4 Cause and Effect Diagram

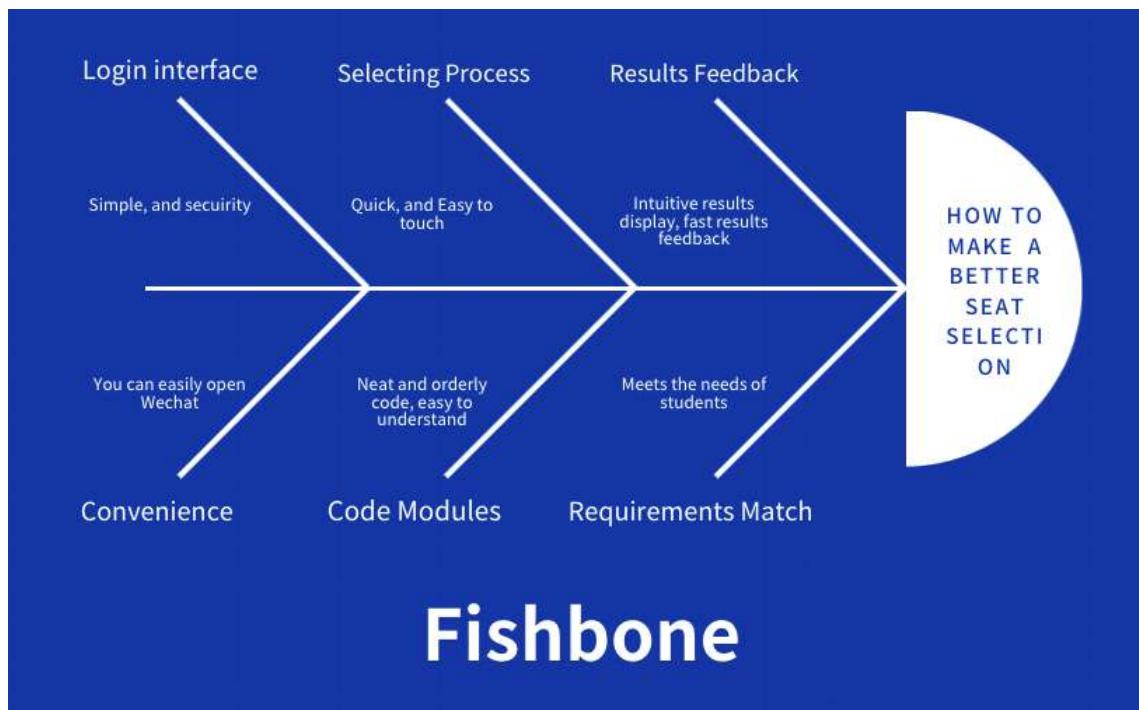
A cause-effect diagram is a visual tool used to logically organize possible causes for a specific problem or effect by graphically displaying them in increasing detail, suggesting causal relationships among theories. A popular type is also referred to as a fishbone or Ishikawa diagram. Cause-Effect can also be diagrammed using a tree diagram.

When diagnosing the cause of a problem, a cause-effect diagram helps to organize various theories about root causes and presents them graphically.

The C-E Diagram is a fundamental tool utilized in the early stages of an improvement team. The ideas generated during a brainstorming or affinity process are used to populate the diagram. Since the list of issues on a C-E may be very large, the team should use a prioritization or multi-vote technique to narrow the list of potential cause that they desire to investigate farther.

At the head of the diagram is the Effect that the team is investigating. The team brainstormed potential causes for this effect. The skeleton becomes the various potential causes and the headers are the column heads from the affinity diagram.

The six aspects we had to consider in the fishbone diagram were, in order of programmatic perspective, the user interface, the student seat selection process and feedback on results, and in order of administrative perspective, security, convenience, and modularity of the code. These are the essentials of how to make a good program.



4.1.5 Decision Table

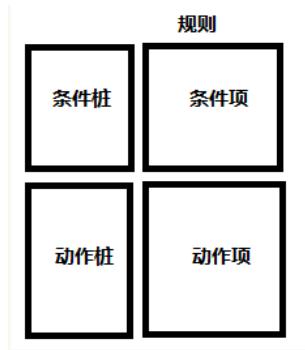
Decision table provides a handy way to explain business rules. In a decision table, rules are presented as columns, while the conditions that constitute the rules and the actions to take are listed as rows. The structured representation is easy to read and understand, making decision table an ideal tool to represent business policy, constraints and rules, regardless of their complexity.

Condition Stub: In the upper left hand side, all the conditions of the problem are listed. The order in which the conditions are listed is usually considered irrelevant.

Action Stub: In the lower left hand side, the possible actions specified by the problem are listed. There is no constraint on the order in which these actions are listed.

Condition Entry: In the upper right-hand column, the values for the conditions in its left-hand column are listed. The true and false values in all possible cases.

Action Entry: in the lower right-hand column, the action to be taken in the case of the various values of the condition entry.



Rule: A combination of condition values and their corresponding action combinations (i.e. the columns that run through the condition and action entries in the decision table) form a rule in the decision table. The number of condition combinations is the number of rules.

Parts of the decision tables are shown in the diagram.

Condition sub\rule	1	2	3	4	5	6	7	8	9	10	11
C1: Is the data completed?	F	T	T	T	T	T	T	T	T	T	T
C2: Is the Login interface showed?	-	F	F	F	F	T	T	T	T	T	T
C3: Is the Password formatted correct?	-	F	F	F	T	F	T	F	T	F	F
C4: Are the two passwords the same?	-	F	F	T	F	F	T	T	T	F	F
C5: Do you agree the wechat agreement?	-	F	T	F	F	F	F	T	T	T	T
A1: Library user Registered successfully	-										
A2: Data is incomplete	√										
A3: Login interface showed error	√	√	√	√							
A4: Incorrect password format	√	√	√		√			√		√	√
A5: two password entered not the same	√	√		√	√	√	√		√		√
A6: Wechat program doesn't run well	√		√	√	√	√	√				

4.2 Function Tests Cases

We have roughly described the program we have created through four methods: the equivalence class division method, the boundary value analysis method, the cause and effect diagram method and the decision table method, as well as in-depth research on our little program, we made the final functional test case. I divided it into two main categories, students and managers, while students were divided into 10 columns, namely serial number, module, function point, sub-function, use case name, precondition, operation process, expected result, actual result, and remarks. Administrators are divided into columns such as Login, View, Delete, and Blackout.

The modules for students are divided into registration, login, home page, seat selection screen, seat selection function, and confirmation. Function points and sub-function points are an introduction to the function under test and facilitate telling the tester exactly where the function under test is located. The expected results we set are the expected results before the test and the actual results are the results the tester actually gets after the test. The notes are the testers' expectations of the feature and help the developers to better improve the feature and better meet the needs of the users. The result of this functional test case is the result of all the development.

Although we encountered many problems in the actual development, for example, we first developed on the macbook and many features were not very adaptable. So in version 2.0 we wrote the database under windows. Our current version is ready for seat selection, but in later versions we will further optimise the intelligent seat selection process, i.e. adding personal preferences, in one step.

Here is the final functional test case for our group.

Sign-in case here is the table

module	function	sub-function	user case name	condition	operation	expected result	actual result
sign in	interface	interface	check interface	interface can be normally open	open sign-in interface	user can see standard interface	pass
		turning	click login	have "login" button	click login button	loading to login interface	pass
	Account sign in	username&password	enter username	username correct	click username frame, enter	successfully done	pass
				username not allowed	click username frame, enter	error	pass
			enter password	password correct	click password frame, enter	successfully done	pass
				password not allowed	click password frame, enter	error	pass
			confirm password	match with typed before	click password frame, enter	successfully done	pass
				doesn't match with typed before	click password frame, enter	error	pass
			Register	have done all the frame	click register	successfully done	pass
				have done all but some are not allowed	click register	error	pass

1	module	function	sub-function	username case
2				check interface
3				choose the building
4				choose the floor
5	Seat Selection	interface port	porting loading	choose the area
6				select seat
7				lock the seat
8				check the status

Ready to move to the next section, It is the last stage of technical testing, also known as delivery testing, which is carried out after functional testing and system testing of software products and before product release.

	condition	operation	expected result	actual result
1	interface been shown normally	click login and see the page	interface showed successfully	pass
2	building page has been flushed	choose the bolck N or I you prefer	block N and I showed	pass
3	floor page has been flushed	choose floor you prefer	floor 1-4 has been showed	pass
4	area page has been loaded	choose the computer/charge area	area page showed successfully	pass
5	seat color and status are normal	click the seat button	select seat successfully	pass
6	seat has not been locked	click the confirm button	seat confirmed	pass
7	have confirmed the seat	back to page and see the seat status	status is " seat confirmed"	pass
8				

4.3 White box Testing

This white box test is for the login.php file, which implements the function of login verification, passing in the correct account and password, output istrue=1 and output the details of the user. Pass in an incorrect account or password and output istrue=0. The code is as follows.

4.3.1 Code

```
1 <?php
2 // User Login
3 header('Content-Type: text/html;charset=utf8');// Set the text encoding for the page to be displayed
4 header('Access-Control-Allow-Origin: *');// can resolve cross-domain issues
5 $con = mysqli_connect("localhost", "root", "root"); // connect to the database
6 mysqli_query($con, "set names 'utf8'");// set the encoding of the database
7 $username = $_GET["username"];
8 $password = $_GET["password"];
9 $date = $_GET["date"];
10 echo "{";
11 if ($con) {
12     mysqli_select_db($con, "room");// select the database to use mydatabase
13     // query if the username and password are the same
14     $query = "SELECT * FROM user WHERE username = '$username' AND password = '$password'";
15     $result = mysqli_query($con, $query); // execute the query
16     if($row = mysqli_fetch_array($result)){//if the data is looked up
17         // spell out a string in json format to pass to the applet
18     }
19
20
21 Translated with www.DeepL.com/Translator (free version)
22     echo '"istru':1,';
23     echo '"username":';
24     echo $row['username'];
25     echo '","number":';
26     echo $row['number'];
27     echo '","infoType":"';
28     echo $row['infoType'];
29     echo '","date":';
30     echo $row['date'];
31     echo '","orderinfo":[';
32     // Array embedded in string, querying all appointments for this user today
33     $query1 = "SELECT * FROM orders WHERE username = '$username' AND date like '$date%'";
34     $result1 = mysqli_query($con, $query1); // execute the query
35     $i = 0;
36     while ($row1 = mysqli_fetch_array($result1)){
37         if($i == 0){$i = 1;}else{echo ',';}// used to determine the comma in the middle
38 }
```

```

39
40     Translated with www.DeepL.com/Translator (free version)
41         echo '{"id":';
42         echo $row1['id'];
43         echo ",""building"":";
44         echo $row1['building'];
45         echo ",""floor"":";
46         echo $row1['floor'];
47         echo ",""sittype"":";
48         echo $row1['sittype'];
49         echo ",""time_index"":";
50         echo $row1['time_index'];
51         echo ",""sit_index"":";
52         echo $row1['sit_index'];
53         echo ",""date"":";
54         echo $row1['date'];
55         echo "}";
56     }
57     echo ']';
58 }else{
59     echo '"istru":0';
60 }
61 }
62 }
63 echo "}";
64 mysqli_close($con);

```

4.3.2 Control flow Chart

Based on the structural characteristics of the source code to be tested, a control flow diagram is drawn, as shown below
 (The Chart is in page 50)

Calculate the loop complexity: $V(G) = P + 1 = 4 + 1 = 5$, where P is the number of decision nodes in the control flow diagram.

Determine all paths.

Path 1: ABG

Path 2: ABCEG

Path 3: ABCDFG

Path 4: ABCDHIJLHFG

Path 5: ABCDHIKLHFG

4.3.3 White Box Test Cases

(page 51)

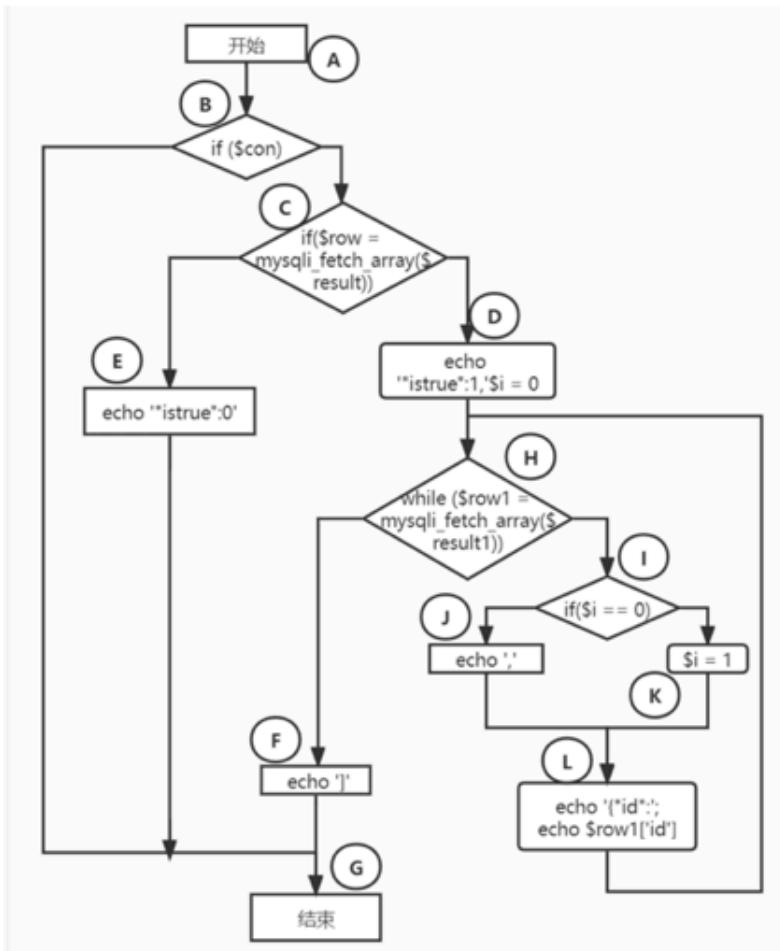


图 4-1 控制流程图

Table 4.1 White Box Test Cases

No.	Test input	Expected results	Coverage path
1	username=admin&password=1&date=20211224 (Database password change)	Prompt for database connection failure	Paths 1
2	username=admin&password=123&date=20211224(Wrong password)	{"istru":0}	Paths 2
3	username=admin&password=1&date=20211224(Correct account and password, no orderinfo data for this user)	{"istru":1,"username ":"admin","number":0,"infoType":"admin","date":"2021-12-14 05:44:31","orderinfo":[]}	Paths 3
4	username=admin&password=1&date=20211224(Correct account and password, the user has 1 orderinfo data)	{"istru":1,"username ":"admin","number":0,"infoType":"admin","date":"2021-12-14 05:44:31","orderinfo":[] (orderinfo Correct output of 1 data item)}	Paths 4
5	username=admin&password=1&date=20211224(正确的账号与密码, 该用户有 2 个 orderinfo 数据)	{"istru":1,"username ":"admin","number":0,"infoType":"admin","date":"2021-12-14 05:44:31","orderinfo":[] (orderinfo Correct output of 2 data items)}	Paths 5

```

Design the white box test case code using PHPUnit as follows:
<?php

require __DIR__.'/../vendor/autoload.php';
class LoginTest extends \PHPUnit\Framework\TestCase
{
    public function test_a(){
        $curl = curl_init();
        curl_setopt_array($curl, array(
            CURLOPT_URL => 'http://localhost:80/login.php?username=admin&password=1&date=20211224',
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => '',
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 0,
            CURLOPT_FOLLOWLOCATION => true,
            CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
            CURLOPT_CUSTOMREQUEST => 'GET',
        ));
        $response = curl_exec($curl);
        curl_close($curl);
        echo $response;
        $istru = json_decode($response, true);
        $this->assertEquals(1,$istru['istru']);
    }
    public function test_b(){
        $curl = curl_init();
        curl_setopt_array($curl, array(
            CURLOPT_URL => 'http://localhost:80/login.php?username=admin&password=123&date=20211224',
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => '',
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 0,
            CURLOPT_FOLLOWLOCATION => true,
            CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
            CURLOPT_CUSTOMREQUEST => 'GET',
        ));
        $response = curl_exec($curl);
        curl_close($curl);
        echo $response;
        $istru = json_decode($response, true);
        $this->assertEquals(0,$istru['istru']);
    }
}

```

```
15     CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
16     CURLOPT_CUSTOMREQUEST => 'GET',
17   );
18
19   $response = curl_exec($curl);
20   curl_close($curl);
21   echo $response;
22   $istruke = json_decode($response, assoc: true);
23   $this->assertEquals( expected: 1,$istruke['istruke']);
24 }
25
26 public function test_b(){
27   $curl = curl_init();
28   curl_setopt_array($curl, array(
29     CURLOPT_URL => 'http://localhost:80/login.php?username=admin&password=123&date=20211224',
30     CURLOPT_RETURNTRANSFER => true,
31     CURLOPT_ENCODING => '',
32     CURLOPT_MAXREDIRS => 10,
33     CURLOPT_TIMEOUT => 0,
```

» ✓ Tests passed: 2 of 2 tests – 50 ms

50ms

```
{"istruke":1,"username":"admin","number":0,"infoType":"admin","date":"2021-12-14 05:44:31","orderinfo":[]}  
{"istruke":0}
```

Time: 176 ms, Memory: 4.00 MB

OK (2 tests, 2 assertions)

Process finished with exit code 0

Figure 4-2 White box test results graph

4.4 Automatic Test

Performance Test

4.4.1 Jmeter Overview

Apache JMeter is a Java-based stress testing tool developed by the Apache organisation. Used for stress testing software, it was originally designed for web application testing, but has since been extended to other testing areas. It can be used to test static and dynamic resources such as static files, Java applets, CGI scripts, Java objects, databases, FTP servers, etc. JMeter can be used to simulate huge loads on servers, networks or objects, test their strength

from different stress categories and analyse overall performance. In addition, JMeter can do functional/regression testing of applications by creating scripts with assertions to verify that your application is returning the results you expect. For maximum flexibility, JMeter allows assertions to be created using regular expressions

4.4.2 Facing to Test

As the system is currently under pressure to accept the impact of business volume, it is facing pressure on system stability and maturity. The performance of the system will certainly become the focus of the issue, the massive amount of data "impact", the system can be stable at what level of performance, facing the increase in business, the system resistance to pressure and other such questions need to be answered through a more realistic performance simulation test, through the test and analysis of the system performance to improve the provision of some important Through testing and analysis, we can provide some important reference data for the improvement of the system performance, which can be used to improve and perfect the system in terms of hardware and software later on. The performance test was conducted on the login module and the information query module, the interface files are: login.php and showsit.php respectively.

4.4.3 Test Environment

The table is on the next page.

Table 4.2 Hardware environment

Computer machine models	Configuration information
HUAWEI MateBook X Pro	<p>Screen size: 13.9 inches</p> <p>Running memory: 16GB</p> <p>Spreadsheet name: HUAWEI MateBook X Pro</p> <p>Screen colour: 100% sRGB (typical)</p> <p>Battery capacity: 56Wh (rated capacity)</p> <p>Storage capacity: 512GB</p> <p>Resolution: 3000x2000</p> <p>CPU Model: 11th Generation Intel® Core™ i7-1165G7</p> <p>Processor</p> <p>Number of CPU cores: 4 cores</p>

Table 4.3 Software environment

Database server	MySQL 5.2	
Application servers	Operating system: Windows 10 flagship 64-bit	Operating system: Windows 10 flagship 64-bit
	Server: Apache	Server: Apache
	Browser: Internet explorer 11.0, Google Chrome	Browser: Internet explorer 11.0, Google Chrome
	PHP 7.0	PHP 7.0
Presses	Stress testing tool Jmeter 5.3	Stress testing tool Jmeter 5.3

4.4.4 Testing Process

1. Build a real running environment for simulated users.
2. Install the stress testing tool Jmeter5.
3. Use HTTP in Jmeter to create test scripts.
4. Use JMeter to organise the launch of the simulated load and collect test data as well as resource data of the test target machine and network.
5. Use the overall reporting component of Jmeter to analyse the test results.
6. Organise and analyse the test results and write a test summary report.

The individual scripts are designed as shown in the diagram below.

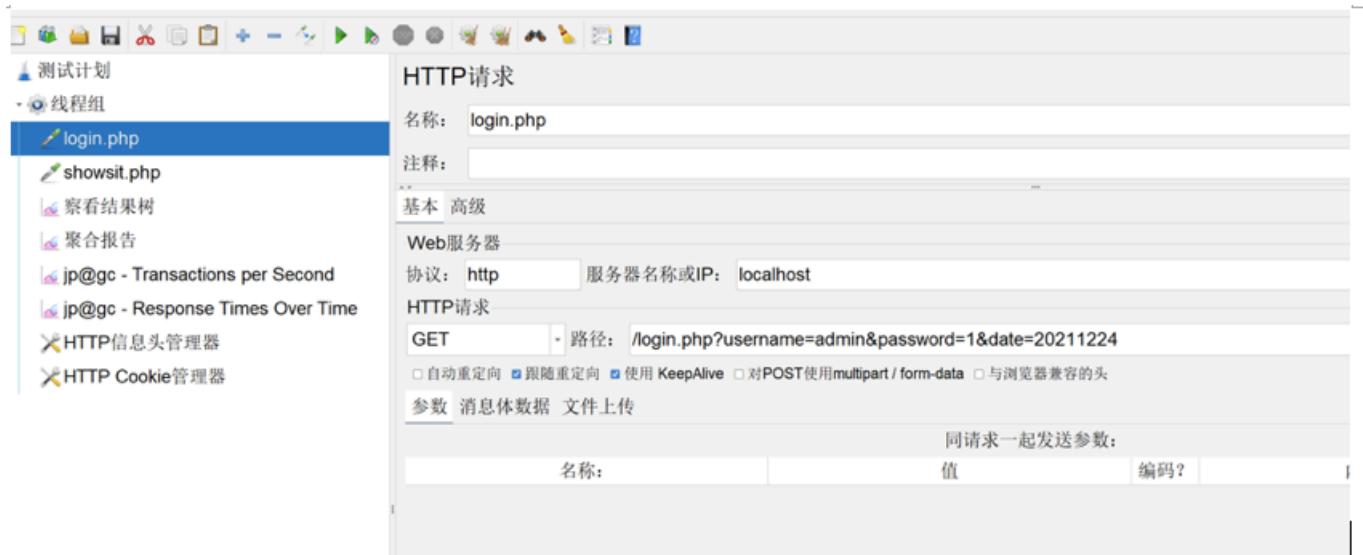


Figure 4-3 Jmeter login script

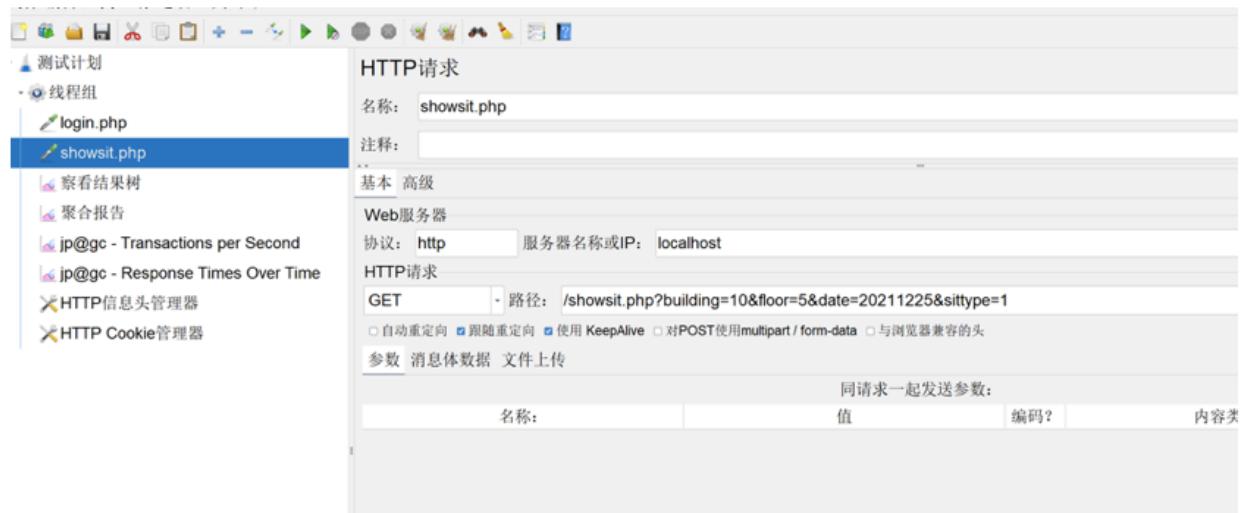


Figure 4-4 showsit query script

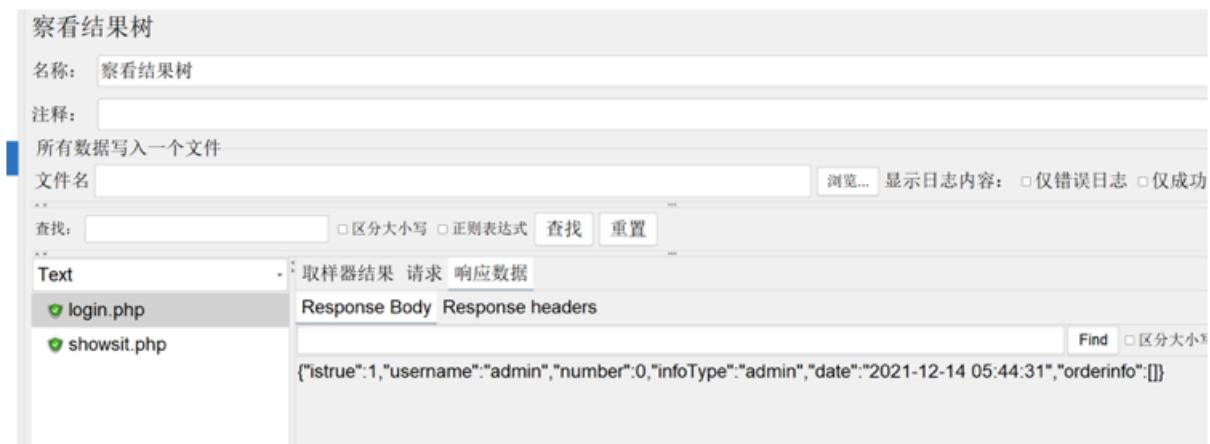


Figure 4-5 Script modulation

4.4.5 Test Design

Concurrent users less than 100, system login business response time within 3s, query response time within 5s, 100 major transactions processed in 1 minute, processing speed within 3s on average, no more than 5s.

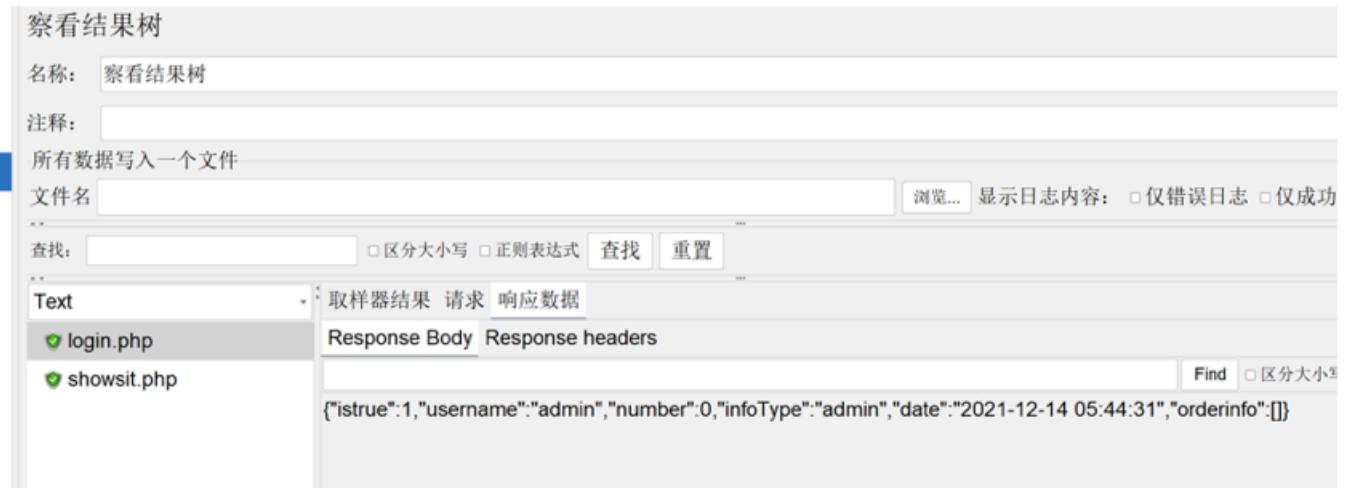


Figure 4-5 Script modulation

Identifier	1	Priority	高	Estimated execution time	2分钟
Use Case Name	Concurrent testing				
Test Purpose	Test the performance of multiple users stressing the system at the same time.				
Environment requirements	4 core CPU 8G RAM				
Test tool	JMeter				
Prerequisites	System login success				
Load pattern and load volume Desired performance metrics	Load pattern and load volume Desired performance metrics		Remarks		
100 concurrent user logins Average response time < 3S, 100% success rate	100 concurrent user logins Average response time < 3S, 100% success rate				
100 users querying concurrently Average response time < 3S, success rate 100%	100 users querying concurrently Average response time < 3S, success rate 100%				

4.4.6 Result

Function Login and query function	Function Login and query function	
Objective To test the performance of multiple users operating the system under pressure at the same time.	Objective To test the performance of multiple users operating the system under pressure at the same time.	
Method Record code containing the following function: Login	Method Record code containing the following function: Login	
Number of concurrent users and transaction execution		
Login	Number of virtual users passed	100
	Average transaction response time (s)	0.665

	Maximum transaction response time (s)	3.179
	Exception rate	0%

Enquiry	Number of virtual users passed	100
	Average transaction response time (s)	0.473
	Maximum transaction response time (s)	4
	Exception rate	0%

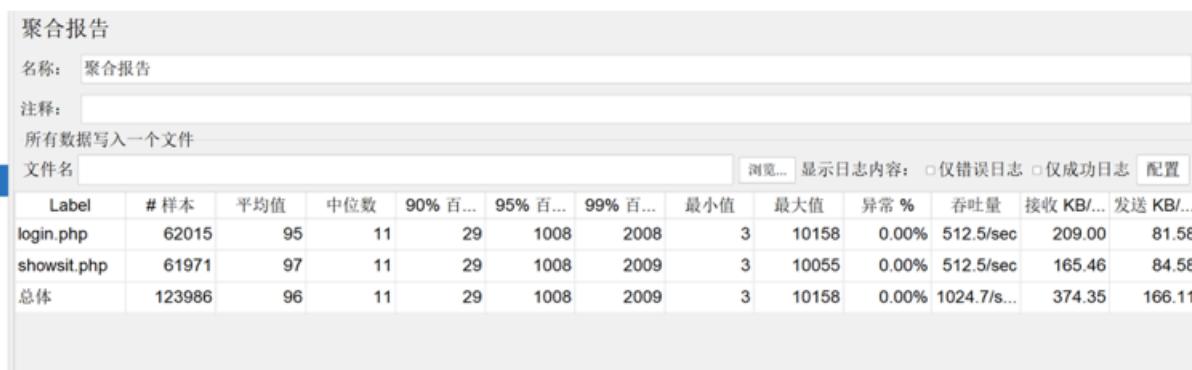


Figure 4-6 Aggregation Report

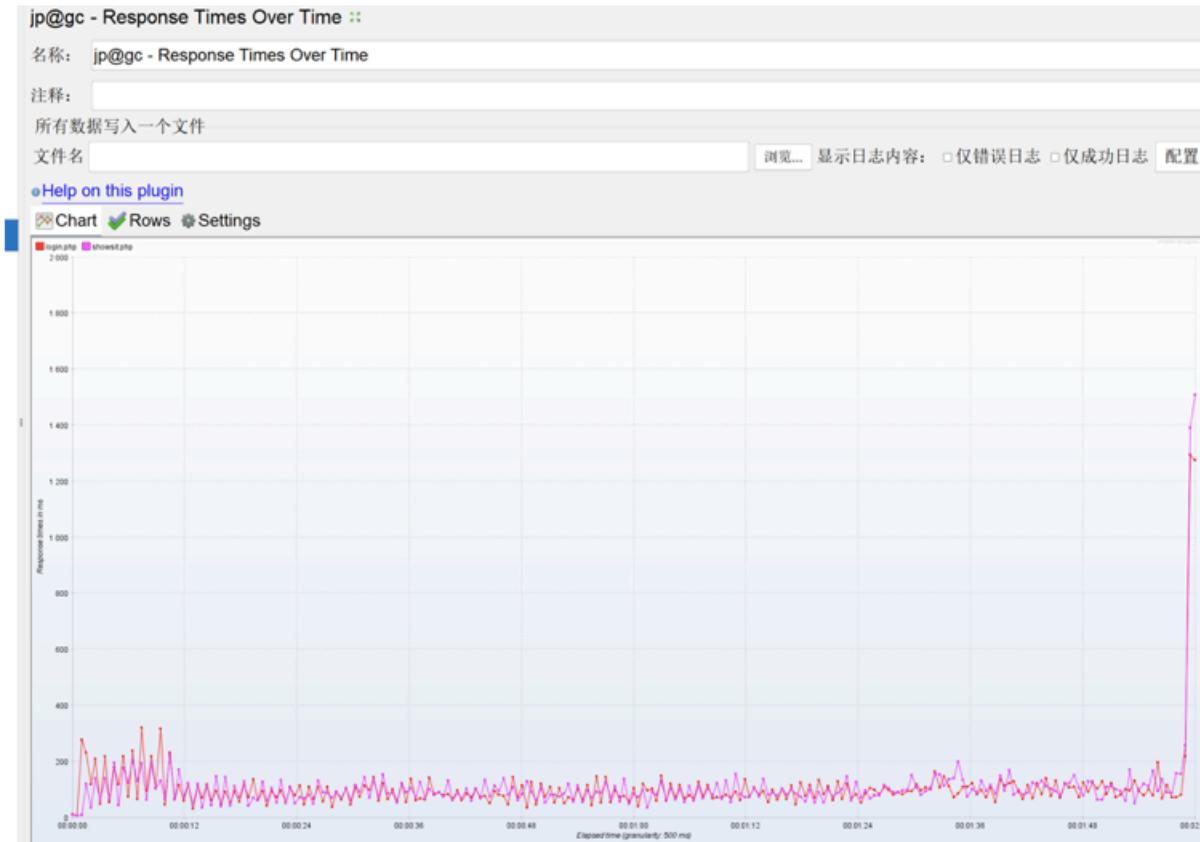


Figure 4-7 Response time trend graph

According to the Jmeter test report, the average response time of the performance indicators for login and query were all less than 3 seconds, with a success rate of 100%; the response time and TPS trend graphs were also relatively stable, and the system performance met the expected indicators.



5 Acceptance Test

5.1 Key Program Simulation Steps

This section is mainly for our field demonstrations of the software. This screen below is the login screen of the applet.



After we log in to our administrator account, we now see an administrator login screen. The administrator's authority lies mainly in the registration of library violations, library usage status changes, and administrator account management.



We create an account here, In this section, it is mainly a registration screen for us. We can enter the username we want, usually a school number or school staff account, and enter the password we want.

if we do not fill in anything, or fill in the account number or password, it will prompt "incomplete".



Similarly, we can't have an empty username in the main screen, otherwise it will prompt.

The password cannot be empty, if it is empty, it will prompt for a password error.

And if we type password twice and they are different, our program still report a error.



Before further demonstration, our program is fully customized according to the number and type of seats in the library of Macau University of Science and Technology, with the following parameters.

And our reservation interface is shown below.

block\ type	Charging seat	mac seat	computer	normal	database computer
Block N-1		30	45		10
Block N-2	56		20	100	
Block N-3			56	174	
Block N-4	56		58	100	
Block I	66		10	170	

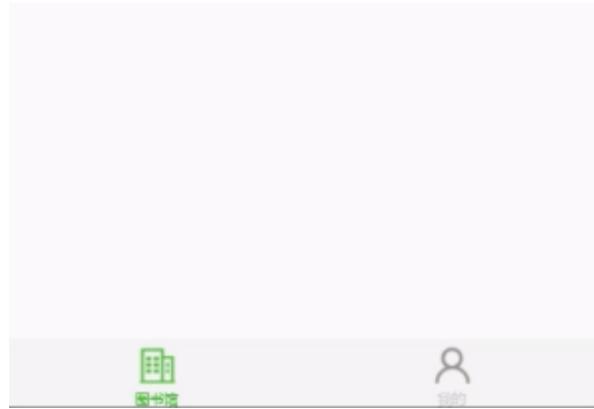


Here we can check we choose Block N or Block I to study.

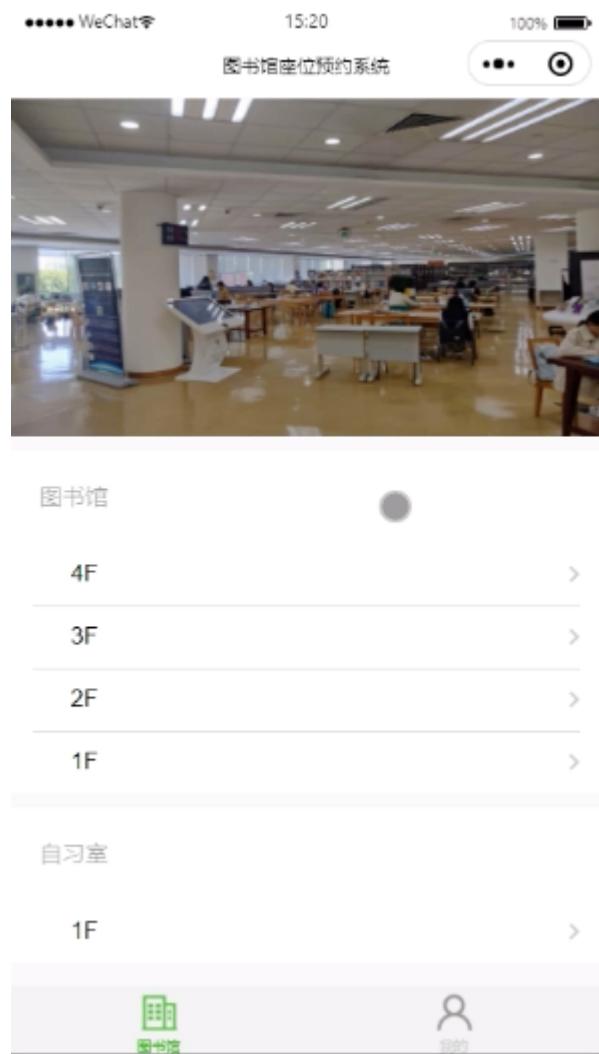


图书馆

自习室



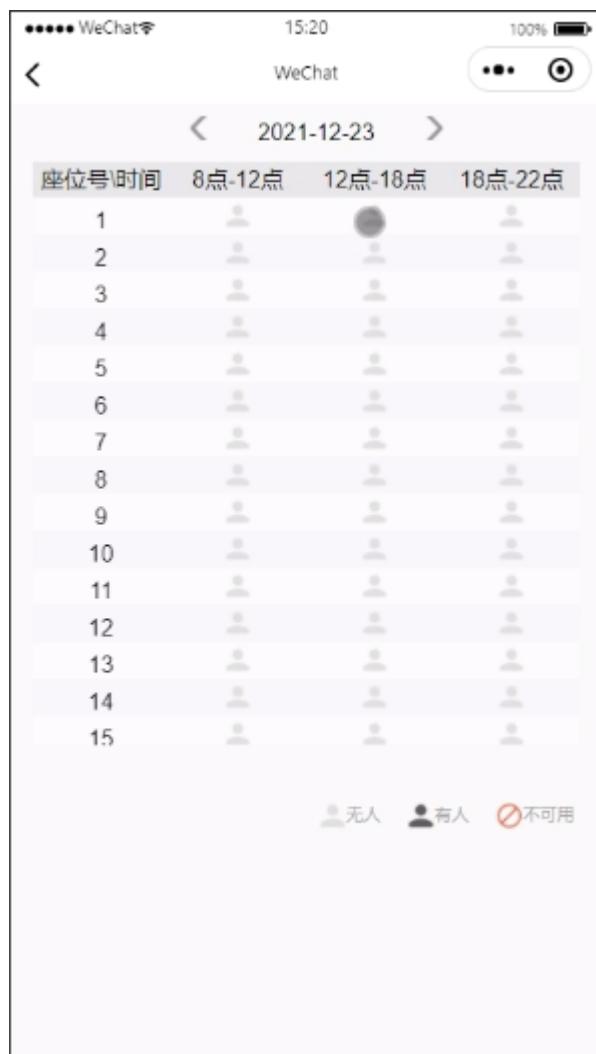
meanwhile, suppose we choose the Block N, we also can select the sepecific floor you want to study in



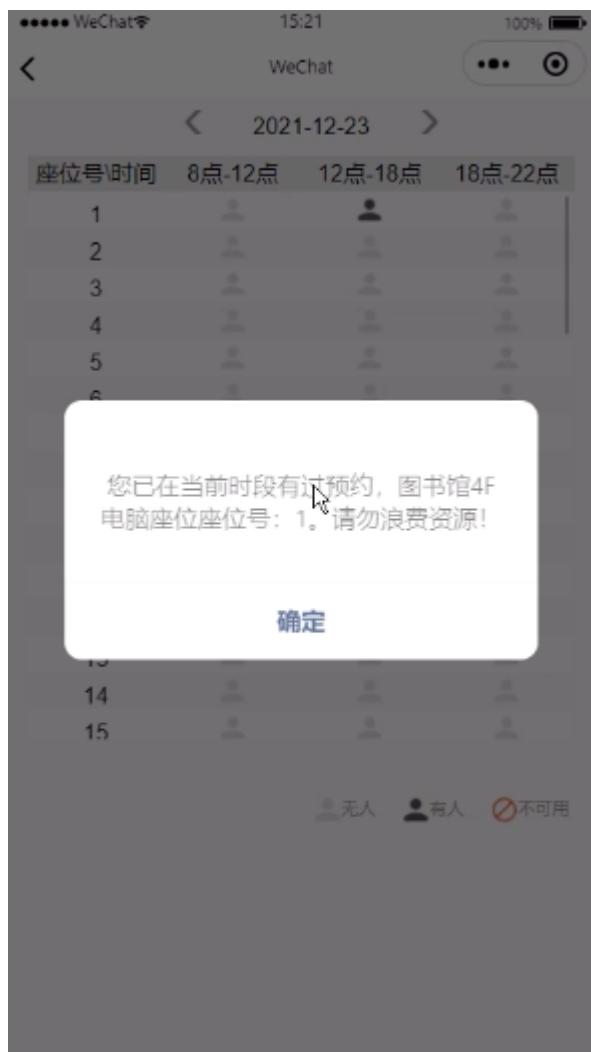
Our seating types are also divided into many different types, which also depends on our preliminary field research and demand acquisition, students can choose their favorite seats according to their preferences.



The next screen we see is an appointment screen. Students can click to select the seat of their choice.



If I want to use the same user to book a different seat for the same time slot, our system will prompt that we can't waste resources.



when we return to home page, we can see there already have our personal reserve information of seat.



Manager Operation Steps What we see now is a violation disciplinary system in the administrator interface. The administrator has permission to black out a classmate.



If a student is blacked out by the administrator, then he cannot continue to reserve a seat and is prompted with the following message.



Administrators also have the authority to choose whether to open a reservation for tomorrow, or to change the status of library use.



In our version 3.0 we have added a database, which means that for a student, if a seat has been reserved and this information has been recorded in the background, then he cannot continue to select that seat.



5.2 Database Interface

In the third version of our program, we successfully created the database of our library seat selection system, which I think is very necessary. Here the librarian can directly select to delete an account, and if a new account is registered, it will also be directly displayed here.

phpMyAdmin

localhost > room > user

浏览 结构 SQL 搜索 插入 导出 导入 操作

显示行 0 - 3 (-4 总计 5 , 查询花费 0.0003 秒)

```
SELECT *  
FROM `user`  
LIMIT 0 , 30
```

显示: 30 行, 起始行 # 0 以 水平 模式显示, 并且在 100 行后重复标题
主键排序: 无

	id	username	password	number	infoType	date
<input type="checkbox"/>	1	admin	1	4	admin	2021-12-14 22:08:40
<input type="checkbox"/>	7	1	1	1	student	2021-12-24 15:23:30
<input type="checkbox"/>	11	12	12	1	student	2021-12-23 15:29:07
<input type="checkbox"/>	12	123	123	0	student	2021-12-23 15:29:10

全选 / 全不选 选中项: 修改 删除 导出
显示: 30 行, 起始行 # 0 以 水平 模式显示, 并且在 100 行后重复标题

查询结果选项 打印预览 打印预览 (全文显示) 导出 显示图表 新建视图

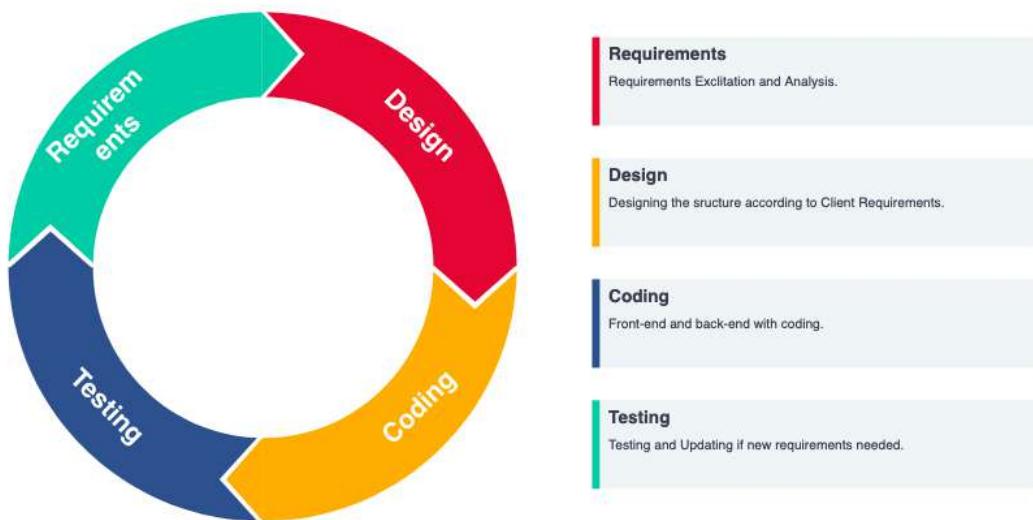
	id	username	password	number	infoType	date
<input type="checkbox"/>	1	admin	1	4	admin	2021-12-14 22:08:40
<input type="checkbox"/>	7	1	1	1	student	2021-12-24 15:23:30
<input type="checkbox"/>	11	12	12	1	student	2021-12-23 15:29:07
<input type="checkbox"/>	12	123	123	0	student	2021-12-23 15:29:10

6 Management and Maintenance Document

After the software product has been developed and delivered to the user, it enters the operation and maintenance phase of the software. This phase is the final stage of the software lifecycle, and its basic task is to ensure that the software remains operational for a considerable period of time.

The amount of work required to maintain the software is significant and, on average, the maintenance costs for large software can be up to four times the development costs.

The main aim of software engineering is to improve the maintainability of software, reduce the amount of work required to maintain it and reduce the total cost of the software system. Software maintenance is the process of modifying software after it has been delivered in order to correct errors or meet new needs. Software maintenance can be defined concretely by describing four activities that may be carried out after the software has been delivered for use.



6.1 Maintenance methods

1. The first type of maintenance is corrective maintenance. During the life of any large program, users are bound to find program errors and report the problems they encounter to the maintenance staff. The process of diagnosing and correcting errors is referred to as corrective

maintenance.

2. The second type of maintenance is adaptive maintenance, the activity of modifying software to fit properly with a changing environment, which is both necessary and frequent.
3. The third type of maintenance is completeness. During the use of the software, users often suggest new features or modifications to existing features, and may also suggest general improvements. In order to meet these requests, refinement maintenance is required. This maintenance activity usually accounts for the majority of software maintenance work.
4. The fourth type is preventive maintenance, where software is modified in order to improve future maintainability or reliability, or to provide a better basis for future improvements.

6.2 Process of Maintenance

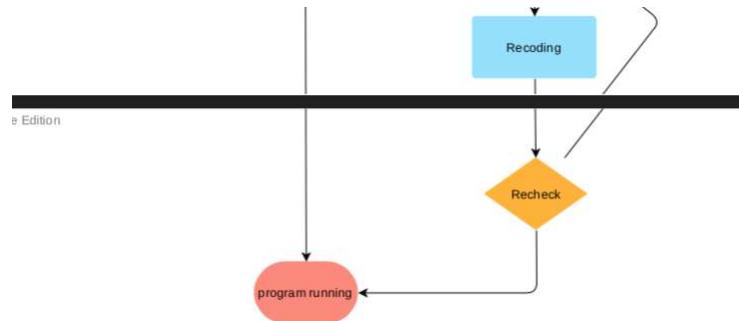
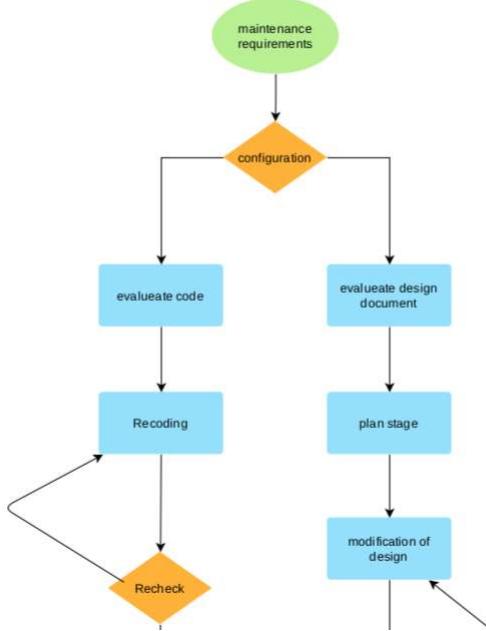
The maintenance process is essentially a modified and compressed software definition and development process, and in fact the work associated with software maintenance begins well before a maintenance request is made.

A maintenance organization must first be established A reporting and evaluation process must then be defined And a standardized sequence of events must be defined for each maintenance requirement

In addition, a record-keeping process should be established for maintenance activities, and review criteria should be defined.

For our MUST library seat selection program, the sepecific items are listed as follows

1. Account security. The main body of our account is done in WeChat applet, and the account security will be guaranteed.
2. Library update. The maintenance of our seat selection system will change with the changes of the library. For example, if some new seats are added to the library, or a batch of seats are temporarily not open, maintenance, etc., we will make data adjustment in our back-end and front-end. Also, our management account will have the function to choose whether to open the reservation or not.
3. Optimization of code. This is also something that our team is very concerned about. We will upload our code to github, and then we can edit it by multiple people, constantly over time we will also ensure the fluency of the code.



6.3 Timeline of Project

Our group completed such a project after almost 4 months. Each person contributed their time to help the team to complete such a work.

For example, our 1.0 did not work on the macbook, but we managed to solve the problem on the windows system when we wrote the back-end test. We also tried to add some new things and started from practicality and simplicity, and have been trying to think how to make the student users get a better experience.

Here is our timeline of project

