



Macau University of Science and Technology

Faculty of Innovation Engineering

COES PLUS
An Enhanced Course Selection System

Luo Yemao 2009853G-I011-0041

Luo Wei 2009853G-I011-0070

Tian Yunfu 2009853V-I011-0048

Lin Longjun 2009853F-I011-0014

2209 SE251 D1

December 2022

Contents

1 Requirements Document	4
1.1 Problem	4
1.2 Background Information	4
1.2.1 Evolutionary Model	5
1.3 Environment and System Models	5
1.3.1 Use Case Diagram	5
1.4 Functional Requirements	5
1.5 Non-Functional Requirements	8
1.5.1 Efficiency	8
1.5.2 Security	8
1.5.3 Reliability	8
1.5.4 Scalability and Maintainability	9
2 Design	10
2.1 Purpose	10
2.2 General Priorities	10
2.2.1 System Security	10
2.2.2 System Performance	11
2.2.3 Diversity of Function	12
2.2.4 Aesthetics	12
2.3 Outline of The Design	12
2.3.1 Project architecture	12
2.3.2 Principle of Design	13
2.4 Major Design Issues	14
2.4.1 Back End	14
2.4.2 Front End	14
2.5 Class Diagram	14
2.6 Sequence Diagram	17
2.6.1 User Login	17
2.6.2 Select & Drop Courses	17
2.6.3 Open & Close Class Grabbing Period	18
2.7 ER Model	19
3 Development Document	21

3.1	Development method	21
3.1.1	Product development method	21
3.1.2	Pros	21
3.2	Team member roles and duties	21
3.3	Development Process	22
3.3.1	Epic Story	22
3.3.2	User Story	23
3.3.3	Product Backlog	23
3.3.4	Sprints & Burn Down Charts	24
4	Testing Document	38
4.1	Black-box testing	39
4.1.1	Equivalence Partition	39
4.1.2	Boundary Value Analysis	41
4.1.3	Cause and Effect Diagram	42
4.1.4	Decision Table Testing	43
4.2	White-box Testing	44
4.2.1	Static White Box Testing	44
4.2.1.1	Code Review	44
4.2.1.2	Code Walk-through	45
4.2.1.3	Static Testing Tools	45
4.2.2	Dynamic White Box Testing	46
4.2.2.1	Control Flow Diagram	46
4.2.2.2	Independent Path Coverage	47
4.2.2.3	Condition Coverage Method	48
4.2.2.4	Testing Cases of Login System	48
4.3	Automatic Testing	50
4.3.1	Visual Bug Testing	50
4.3.2	SQL Injection Testing	50
4.3.3	Back-end Concurrent Testing	52
4.3.4	Back-end Performance Testing	52
4.3.5	Back-end Unit Testing	53
4.3.6	Back-end Automatic testing of interface	54
5	Management and Maintenance	55

5.1	Software Management	55
5.2	Management Model	56
5.2.1	Egoless Team Organization	56
5.2.2	CMMI	56
5.2.3	PERT Chart	57
5.2.4	Gantt Chart	58
5.2.5	Earned Value Chart	59
5.3	Software Maintenance	59
5.3.1	Corrective Maintenance	60
5.3.2	Adaptive Maintenance	60
5.3.3	Perfection Maintenance	60
5.3.4	Preventive Maintenance	60

1 Requirements Document

1.1 Problem

For the current COES system of M.U.S.T., We have collected following problems:

- 1) The system will crash and be out of service when a large number of requests come in at the same time.
- 2) Administrators arrange various faculties' classes in a difficult way.
- 3) Available seats for a class are not accurate and will suddenly turn to 0.
- 4) Too many users access and the server returns too much data, resulting in insufficient bandwidth and response code 400 or even no response.
- 5) When the user is logging in, the website will redirect the user to a new page to enter a verification code which may cause the user to lose connection during redirecting.

1.2 Background Information

Obviously, the occurrence of these problems is hard to meet the demand of thousands of enrolling. For the above problems, and with a faith to solve them, we are motivated to develop a system that could have a good performance while enrolling. Additionally, we'd like to implement some other functions like secure login and commenting for teachers. Here we analyzed some flaws of the present COES system, which is also what our product needs to complete:

- 1) There's a long delay in response when enrolling on COES, thus it's hard to meet the demand of most of the students
- 2) The overall architecture of the system does not meet the requirements of high concurrency scenarios and cannot achieve high availability in certain circumstances.
- 3) The system is developed from an early age, resulting in backward technology and unreasonable business logic which lead to long responding time and high-performance costs in responding to each request.
- 4) In some special scenarios, such as the beginning of course selection, the system's access request traffic will suddenly soar, causing the service to fail to respond normally. But at other times. The demand for the system's performance is not that much. It does not need a server with incredibly high performance to handle its business.

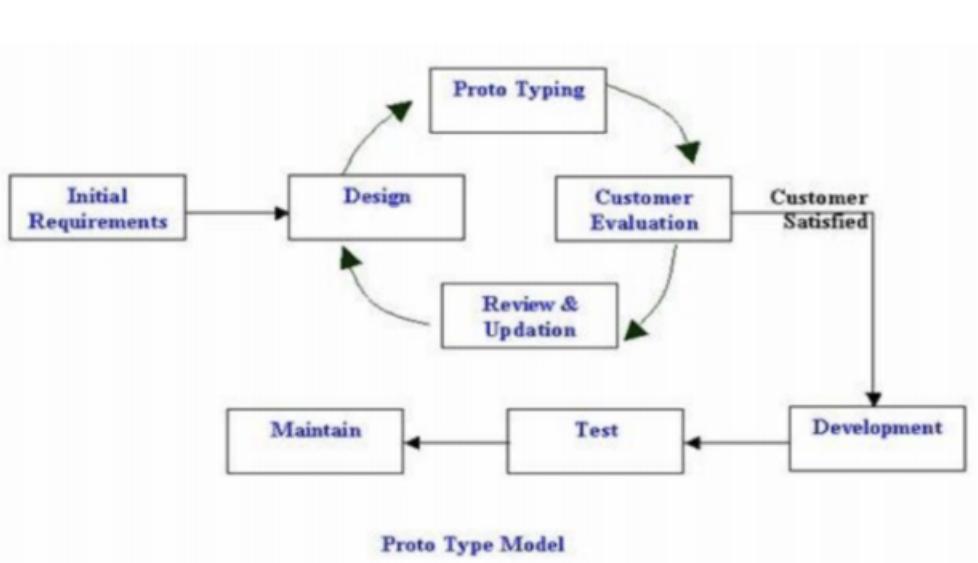


Figure 1: Evolutionary Model

1.2.1 Evolutionary Model

Obviously, we cannot define all the requirements of our COES plus in advance. That's the reason why we use the evolutionary model. For implementation, We need to communicate core requirements to users during the development process so that they are able to provide effective feedback when we see the core requirements implemented to support the final design and implementation of the system. When the core system is put into operation after development, users try it out first, complete their work, and put forward new requirements to refine the system and enhance the system's capability. Software developers implement an iterative process of development based on user feedback.

1.3 Environment and System Models

The software needs to run on Linux (Ubuntu or CentOS) and Windows. School administrators, students, and teachers mostly use Google and Edge browsers, so front-end pages are required to render in both browsers. As for the server in university ,it supports for multiple distribute micro services (JDK 1.8) . For databases, university systems support MySQL and no-SQL data storage.

1.3.1 Use Case Diagram

Use Case Diagram:

1.4 Functional Requirements

- Requirements for Admin
 - 1) Adding, modifying, and deleting course information.

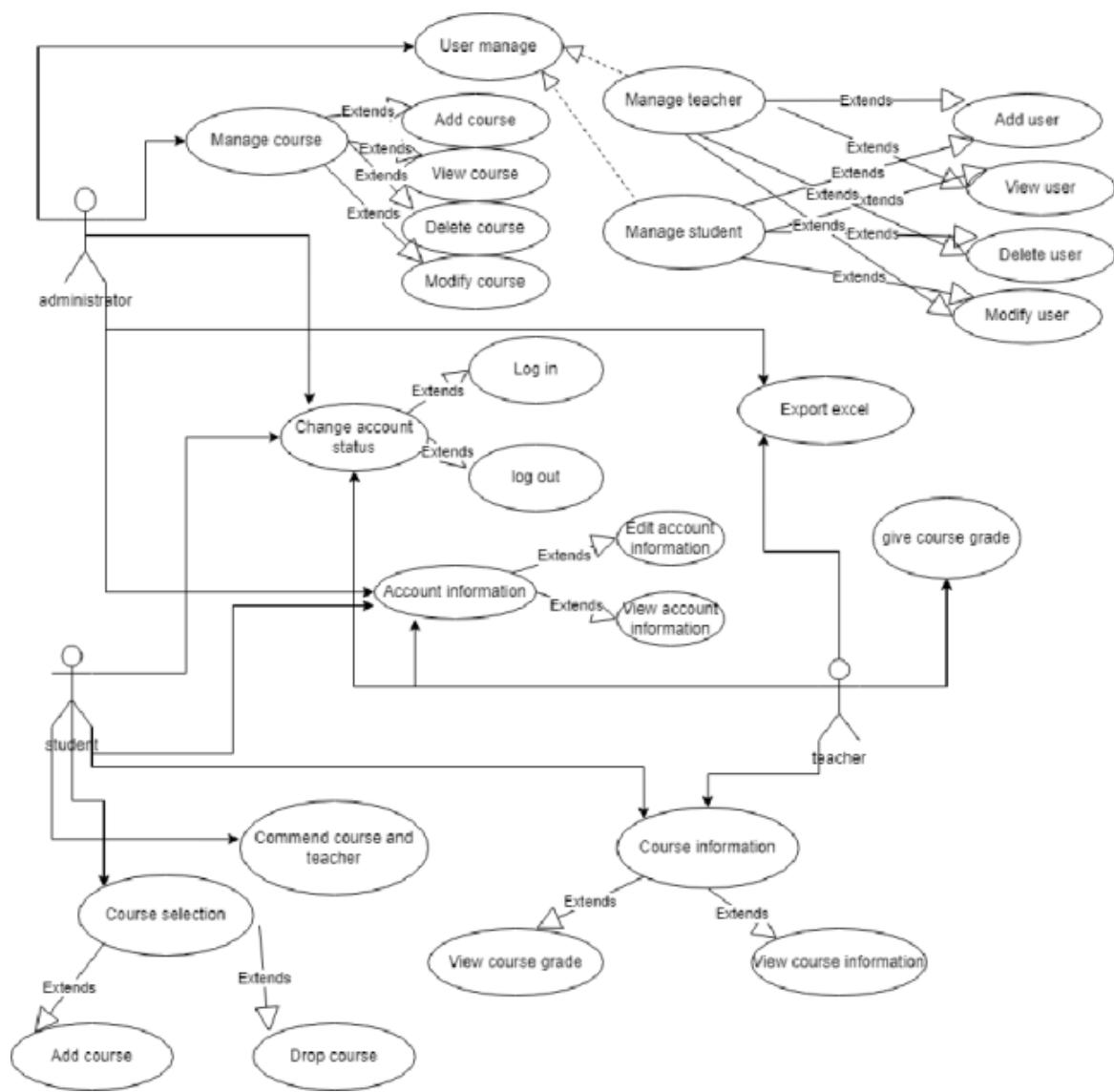


Figure 2: Use Case Diagram

- 2) Adding, modifying, and deleting teachers' information.
- 3) Adding, modifying, and deleting students' information.
- 4) Adding, modifying, and deleting faculty information.
- 5) Adding, modifying, and deleting semester information.
- 6) Control the enrolling time of a semester.
- 7) Sending a verifying email when users or administrators would like to change their password.
- 8) Monitor user traffic in real time.

- Requirements for Client

- 1) Users need to log into their accounts by entering their college ID and password.
- 2) Student users can modify their personal information.
- 3) Student users can add, or cancel courses they like at enrolling time.
- 4) Student users can see the courses they have chosen.
- 5) Students users will receive a confirmation window whenever students users chose a course.
- 6) Students users can comment for their teachers.
- 7) Teacher users can modify their personal information.
- 8) Teacher users can see the courses they teach.
- 9) Teacher users can add, delete, and modify the courses they are responsible for.
- 10) Teacher users can grade the class students on a course.
- 11) Teachers users can take a view of the comments of themselves written down by the students.

- Other Functional Requirements

- 1) The system must identify the conflicts of course time with those chosen whenever student users chose a course, and send an alert window if conflict.
- 2) When searching for courses, it can be searched by course number or course name, or even the instructor's name, fuzzy search should also be supported.
- 3) When needing to add or query some information such as students, teachers, and courses, the system should support importing from an Excel file and outputting as an Excel file.
- 4) In order to deal with potential security risks, the system should record the user's last login IP address and can limit specific IP addresses. When risks are detected, the system should automatically detect them and take certain actions to prevent danger from happening.
- 5) In order to comply with local legal requirements, the system will force users to change their passwords once a year.

- 6) The system should have a certain notification function, such as sending an email to students to remind them before the course selection starts, and sending an email notification to students after the course selection is completed. Administrators can also batch-select students to send customized messages.
- 7) In order to facilitate students to know more about teachers and courses, the system should allow students to evaluate teachers and courses, so that other students who want to choose this course can make a choice in the future.

1.5 Non-Functional Requirements

1.5.1 Efficiency

The system must be able to meet the high concurrent requests of students during course selection, process these requests correctly, and fulfill business logic perfectly. And it must have high fault tolerance performance, can quickly and reliably self-heal in the event of a crash, and quickly restore stable services.

1.5.2 Security

Because our system is a school's educational administration system, which stores a lot of information about students, teachers and other staff. Once information leakage occurs, it will not only cause huge losses to users, but also seriously affect the reputation of the school, so we have a very high demand for system security.

1.5.3 Reliability

Not only the logic of the system needs to be improved, but the overall architecture of the system should also be reasonably designed, which means that not only a reliable logic processing module is required, but also an excellent system architecture needs to be designed, which includes high reliability and can provide High throughput database module or cluster. And how can these requests be equally sent to each logic processing module using an outstanding load balance method.

Since the system will be deploy as multiple distributed micro services. The traditional way of saving tokens in local cache can not be used. A distributed session storage method need to be used in order to satisfy the requirement of letting each micro service checking tokens carried by different requests.

In terms of UI design, for the background management system, the UI design should be clear and concise. It should not be too fancy. All the quick operations of the administrator should take priority. As for the main COES system, the UI design should give priority to functions. Similarly, it should not design to display too many resources that occupy high bandwidth, such as pictures or animations. What's more? Considering that many students are using the system for the first time, the system should have sufficient prompt functions, and there should be detailed information prompts for users' failure operations.

1.5.4 Scalability and Maintainability

As for the scalability and maintainability of the system, the distributed micro service development logic is adopted, which makes it easier to expand the performance of the system. It is also good for the version iteration of the system. Since the system is serviced by multiple micro service modules, the system can adopt gray-scale publishing during the iterative upgrade, which can reduce the risk of iterative versions and prompt the user experience. Similarly, the micro service architecture can also embrace cloud native operation and maintenance solutions, such as the k8s system, making continuous delivery more convenient and fast.

Above is the current survey of our project, and in the future time, we will work more on the requirements collection so that we can improve our product further.

2 Design

2.1 Purpose

The purpose of this document is to design the functional design specification of the software. The design documents describe the main systems of the project and refer to the requirements (traceability) implemented by the design.

COES System

- 1) Students are able to look up the course which could be chosen by them as well as its information(course time, classroom, teacher, comment)and select their ideal course in the system.
- 2) Teachers are able to look up the course which they teach and its information which may include students in class, course information. Also, teacher should be able to grade each student in class.

ADMIN System

- 1) Administrator is a role who can manage all users information (course information and personal information) and do CRUD operations to entities in database.

2.2 General Priorities

2.2.1 System Security

Because the system involves large amount of users' privacy information such as phone numbers, names and sex. Once the information leakage takes place in the system , it would be a considerable lost to school reputation and a huge threat to all users. We would choose the most reliable data transportation and storage and take great responsibility to protect the most secret information which is users' credential. In order to do it, we know considered of using MD5 encrypt method to encrypt users' credential. Hopefully it can prevent destroyer from making trouble in the system deliberately.



Figure 3: MD5 Encryption

Beyond physical protection such as encryption. We also find it's important to design the system security protection logic. We will record admin's login IP address. When it changes, certain method will be used to verify the status of this admin account. At

the same time, we also encrypted the exported excel information form and sent the password to the designated administrator's mailbox to ensure the security of the information. At the ADMIN side, we also added a watermark to the display day of the data information, effectively preventing information disclosure and providing a proof of accountability.



Figure 4: Log In Warning



Figure 5: Export Data Warning

Figure 6: Watermark

2.2.2 System Performance

Performance mainly focuses on system stability and the overall QPS the system can handle. In order to enable more students to log in the system at the same time and complete their goal which is select courses without difficulty. We should consider to design the system in an efficient way. Hence we need to consider both business processing module and storage module to make them both efficient and reliable.

In order to accomplish this task. We plan to design the system's business processing module to be distributed and be made up of multiple micro services. A distributed

architecture can make the system to be flexible and easily upgrade it's performance with lower cost. As for the storage module. First we plan to store small data which access frequently in memory and store other data in disk. This can help increase IO throughput of the storage module. When the bottleneck come's to SQL server which deal with data in disk. We try to solve this problem by separate read and write. As we all know write operation in SQL server is much slower than read operation causing by table lock when executing update and other operations. Hence the data storage module's performance will improve if we separate read and write operations in different SQL server. Other method such as adding index in database will also be taken.

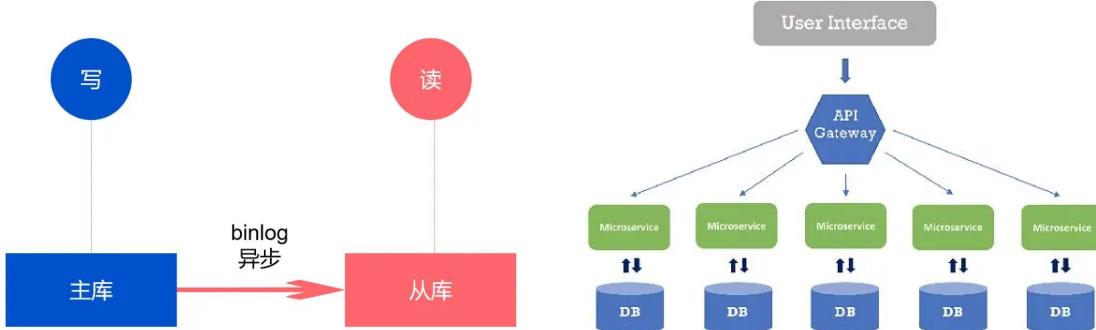


Figure 7: Dual Database Mode

Figure 8: Micro Service Architecture

2.2.3 Diversity of Function

To let the process of course looking up information and selection be more convenient ,we would add some function, like fuzzy query. To facilitate the management of student accounts it is necessary to let admin export his or her query data in the form of excel files.

2.2.4 Aesthetics

For the reason that pages with too many components will slow down the loading speed of the web page and take too many bandwidth, we will only add a proper amount of web components to make the web page beautiful on the premise of ensuring the speed of students' course selection, so that students can have a comfortable visual experience as well as a efficient speed when choosing courses.

2.3 Outline of The Design

2.3.1 Project architecture

The design of the entire COESPLUS follows the service-oriented architecture pattern. We split the entire system into COES system and ADMIN system for development.

These two systems together provide services for three users. This design effectively decouples the system business and regulates the growing complexity. It also makes it easier for us to carry out agile development.

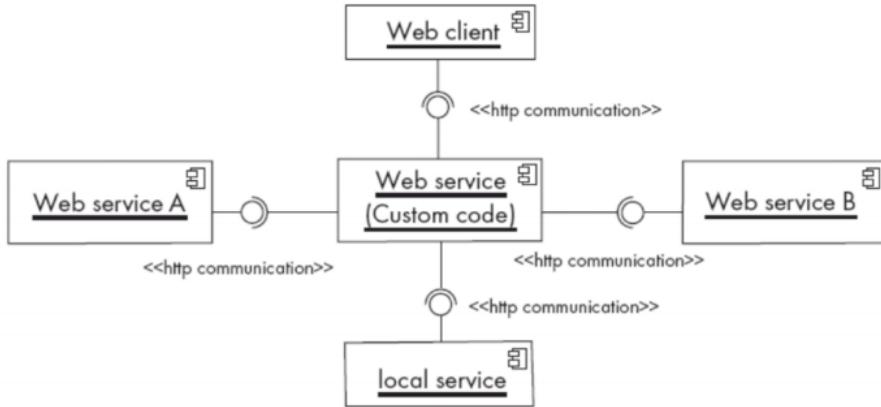


Figure 9: Service-Oriented Architecture

2.3.2 Principle of Design

- 1) Divide and conquer: A large logic is composed of many small and refined logic, so that a large task can be divided into many small tasks, which not only facilitates us to develop a complex function, but also makes the logic more complete and clear.
- 2) Decoupling where possible: Code decoupling can reduce the dependencies between programs, so as not to cause a large-scale crash of the system. At the same time, when an error occurs, it can locate the location of the error code more quickly, so that it can be repaired accurately and quickly. Improve the efficiency of development, and at the same time avoid the appearance of large blocks of logic, which can help subsequent developers to become familiar with the code more quickly and participate in development as soon as possible.
- 3) Keep the level of abstraction as high as possible: We abstract all fields in each table in the database into attributes in the same class, so that we can easily add, delete, check and modify data, simplify code, and improve development efficiency.
- 4) Design for Portability: We divide different services into different business segments. The control layer, business layer and data persistence layer are used to process HTTP requests, logic requests and database operation requests respectively. Which means if we want to rebuild a certain module or function. There is no need to rewrite all source code. Instead, we can just simply change the implementation of a certain interface.
- 5) Design for flexibility: We design the service to be fully automated. Users only need to inform the control layer of the request, and services such as the business

layer will be completed automatically without the user's reconfirmation. This can make the user's interaction with the page more concise and clear.

- 6) Increase re-usability where possible: The components we use in UI can be divided into simple logic, so that its re-usability can be largely increased.
- 7) Reuse existing designs and code where possible: We have selected an open source background management framework which supports secondary development. This will help to reduce a large amount of time and cost.

2.4 Major Design Issues

2.4.1 Back End

- 1) We find that the course selection system only undertakes the task of students' course checking during the non course selection period. At this time, the overall load of the system is low and the performance demand is weak. However, if the load of the system rises sharply during the course selection period and the performance demand is strong, how to design a system with elastic performance has become a major problem. If the performance of the system cannot scale elastically, it will cause a lot of performance waste in most non course selection periods.
- 2) It is very important to ensure that the number of students who choose courses does not exceed the set value. There are many ways to achieve this, but how do we balance the safety and performance of course selection? If we limit the write data from the database level, it will greatly affect the system performance, so we need to find a reasonable degree to limit.

2.4.2 Front End

- 1) When the system traffic suddenly increases, every bit of bandwidth is particularly important. If we design the interface too fancy and use too many pictures, each user will use too much bandwidth to load the page, which will greatly affect the access experience of other users. Therefore, how to balance the beauty of the interface while ensuring the browsing experience is a big problem.

2.5 Class Diagram

Based on our three-tier MVC architecture, we provide three types of diagrams, including all classes in the control layer, all classes in the service layer, all classes in the data persistence layer, and all entity classes.

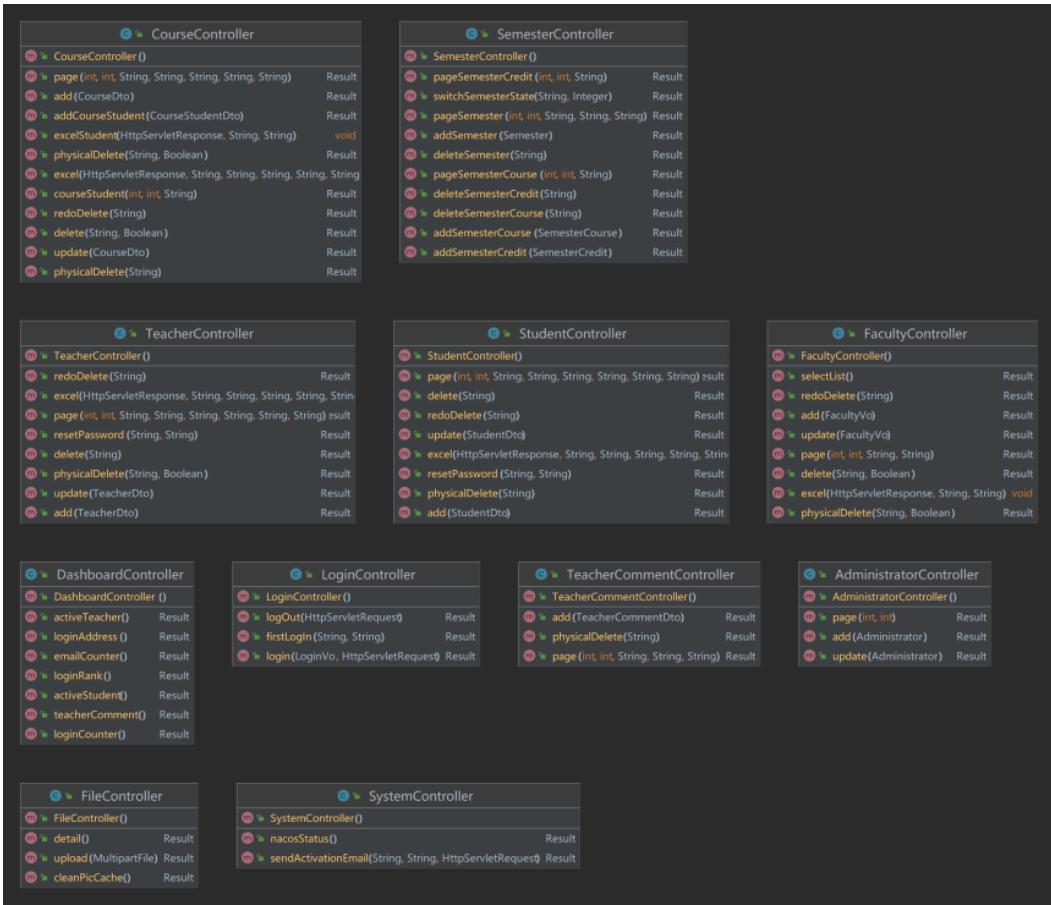


Figure 10: Class Diagram 1

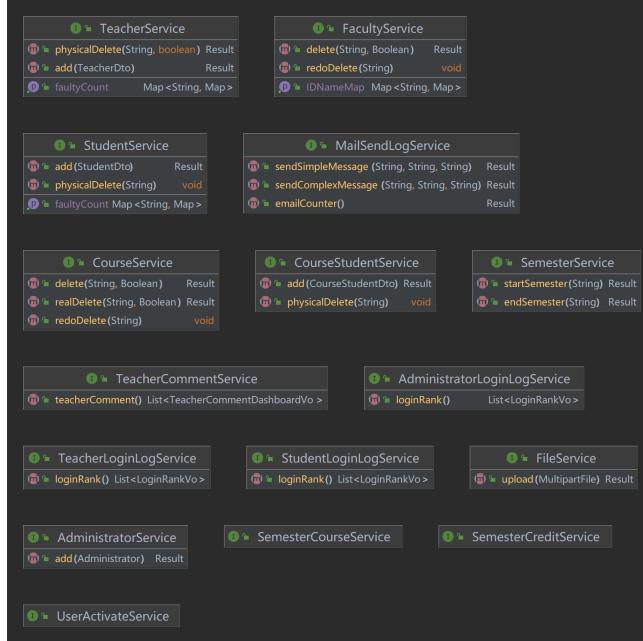


Figure 11: Class Diagram 2

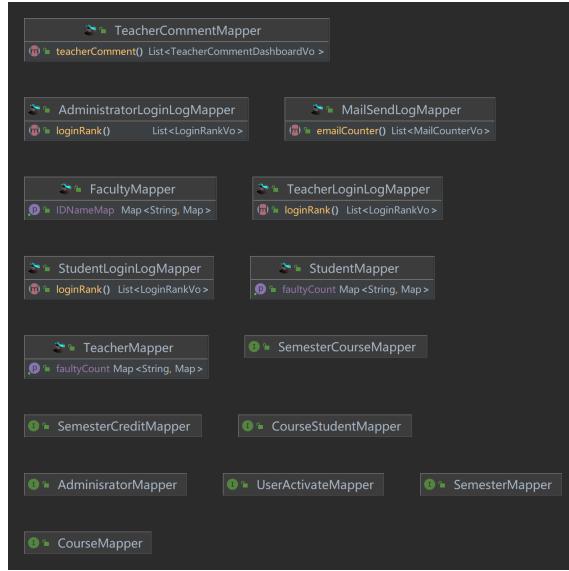


Figure 12: Class Diagram 3



Figure 13: Class Diagram 4

2.6 Sequence Diagram

Sequence diagram is a UML interaction diagram. It shows the dynamic collaboration among multiple objects by describing the time sequence of sending messages between objects. It can represent the sequence of use case behavior. When a use case behavior is executed, each message in it corresponds to a class operation or trigger event in the state machine that causes the transition. We choose to draw the sequence diagram of user login, administrator login, student selection of courses, student drop courses, administrator starting the semester and management ending the semester.

2.6.1 User Login

These are the sequence diagrams for user login. First, the user enters the account password and the browser sends the login request to the corresponding control layer interface. The control layer interface will call the corresponding service class method to query whether the account password entered by the user is correct from the database. If it is correct, the user token will be returned and the user information corresponding to the token will be stored in thread-local. Otherwise, the login error and error information will be returned. If the login is successful, it will also judge whether the user's last login IP is consistent with the current login IP. If there is any change, it will send an email to remind the user.

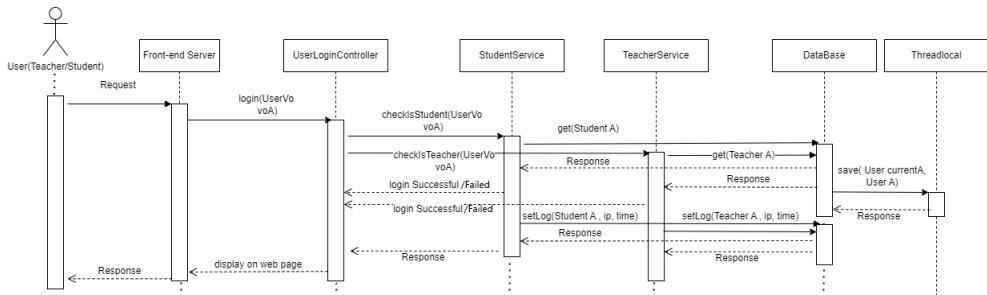


Figure 14: User Log In

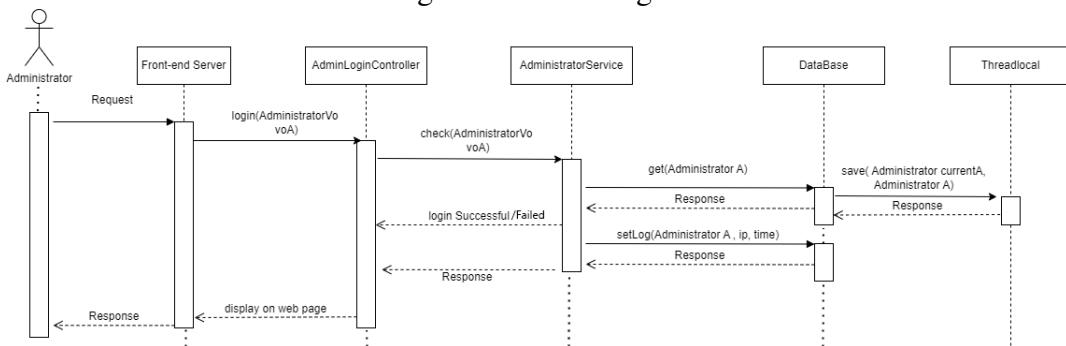


Figure 15: Admin Log In

2.6.2 Select & Drop Courses

These are the sequence diagrams for students to select courses and drop courses. First, the user clicks the corresponding drop or selection button on the web page, and the

browser sends the corresponding request to the control layer. First, it will go through a layer of interceptors to determine whether the user has logged in. If yes, it will obtain the logged in user information from the thread-local, and then call the subsequent method to the service layer through the user ID in the user information.

If the user clicks the course selection button, it will first verify whether the user's credit limit is reached, and if not, try to obtain the semaphore. If it is successful, the course selection information will be written into the database and a prompt will be returned. If it fails, the user will be prompted for the reason for failure.

If a failure occurs in the process of writing to the database, the failed rollback will be automatically triggered to avoid data inconsistency. If the user clicks the drop out button, a semaphore will be released first and the course selection data will be erased from the database. If it is successful, a success prompt will be returned; otherwise, the reason for failure will be returned. At the same time, failure rollback will be automatically triggered if an error is sent.

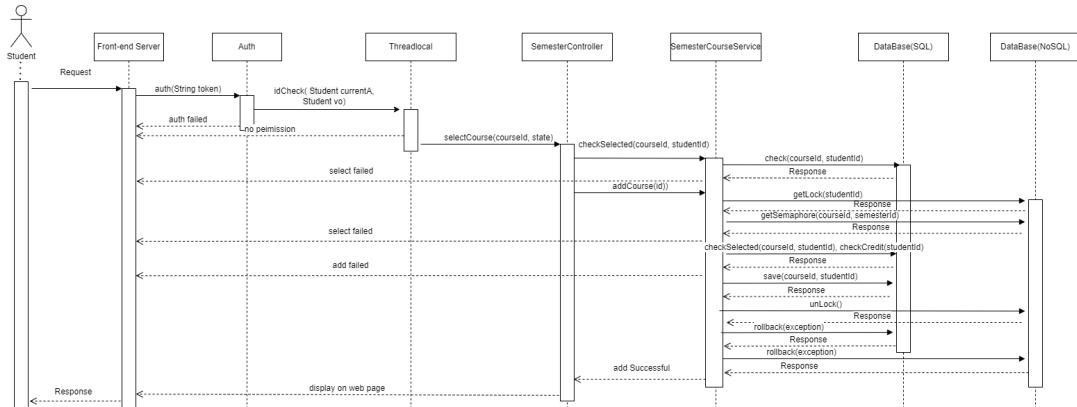


Figure 16: Student Drop Course

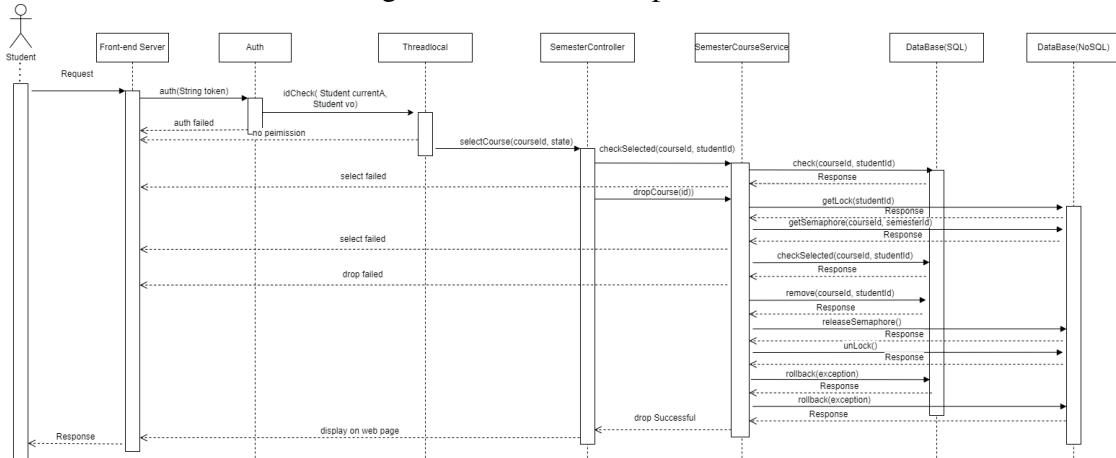


Figure 17: Student Select Course

2.6.3 Open & Close Class Grabbing Period

These are the sequence diagrams of the administrator's opening and closing of the class grabbing period. First, when the administrator clicks the start or end lesson grabbing button, the browser will send a request to the control layer interface. After re-

ceiving the request, the browser will first pass the request through the interceptor. If it passes, the next step will be carried out. Then, the semester ID parameter carried by the request will be transferred to the service layer method to try to change the semester status.

If the semester is started, the system will first query from the database whether the credit limit of all colleges has been set. If yes, it will check whether there are courses to choose from. If yes, the system will store the course balance in Redis in the form of semaphores. If everything is completed successfully, a success prompt will be returned, Otherwise, an error message is returned and the failure rollback is automatically performed. If the end button is clicked, the course balance will be read from Redis and the corresponding course balance in the database will be updated. After these two steps are completed, the success information will be returned, otherwise the error information will be returned and the failure rollback will be automatically performed.

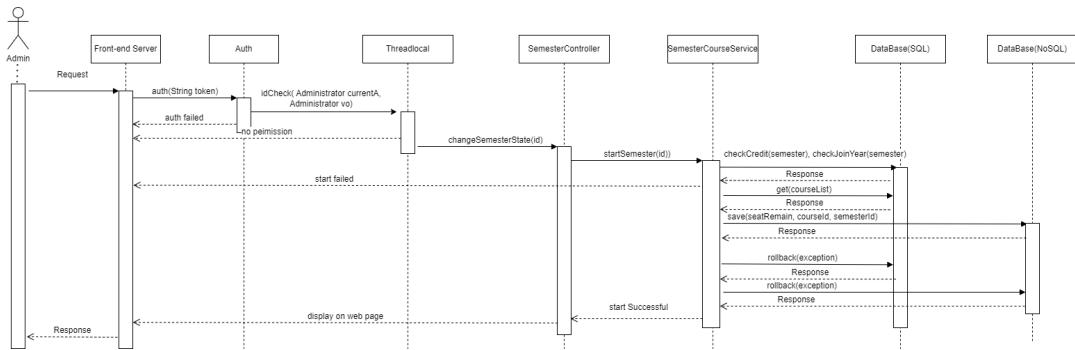


Figure 18: Open Class Grabbing Period

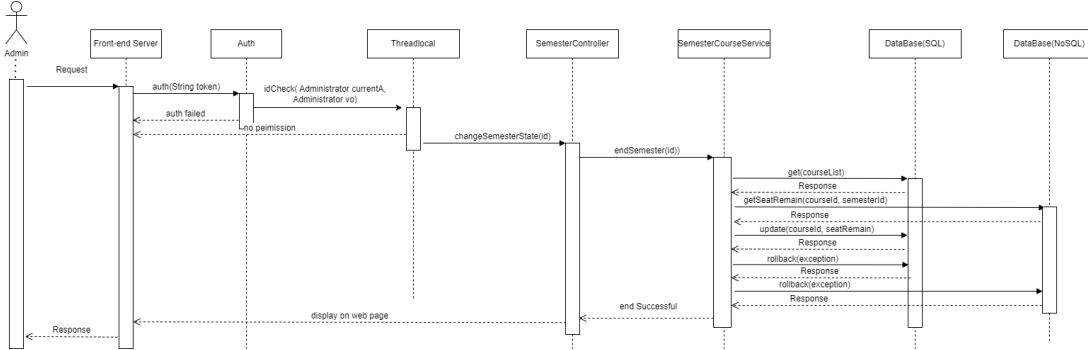


Figure 19: Close Class Grabbing Period

2.7 ER Model

In order to design a reliable, efficient and stable database, we first designed the ER model of the database. We plan to use a total of 15 tables to store information. At the same time, we have set unique indexes and non null constraints for each primary key ID, and we have also set common indexes for the foreign keys of each intermediate table. This can effectively speed up queries when the data volume is very large. At the same time, we also set some redundant fields, such as isDeleted. This can achieve the effect of virtual deletion, providing the possibility of data recovery after accidental deletion.

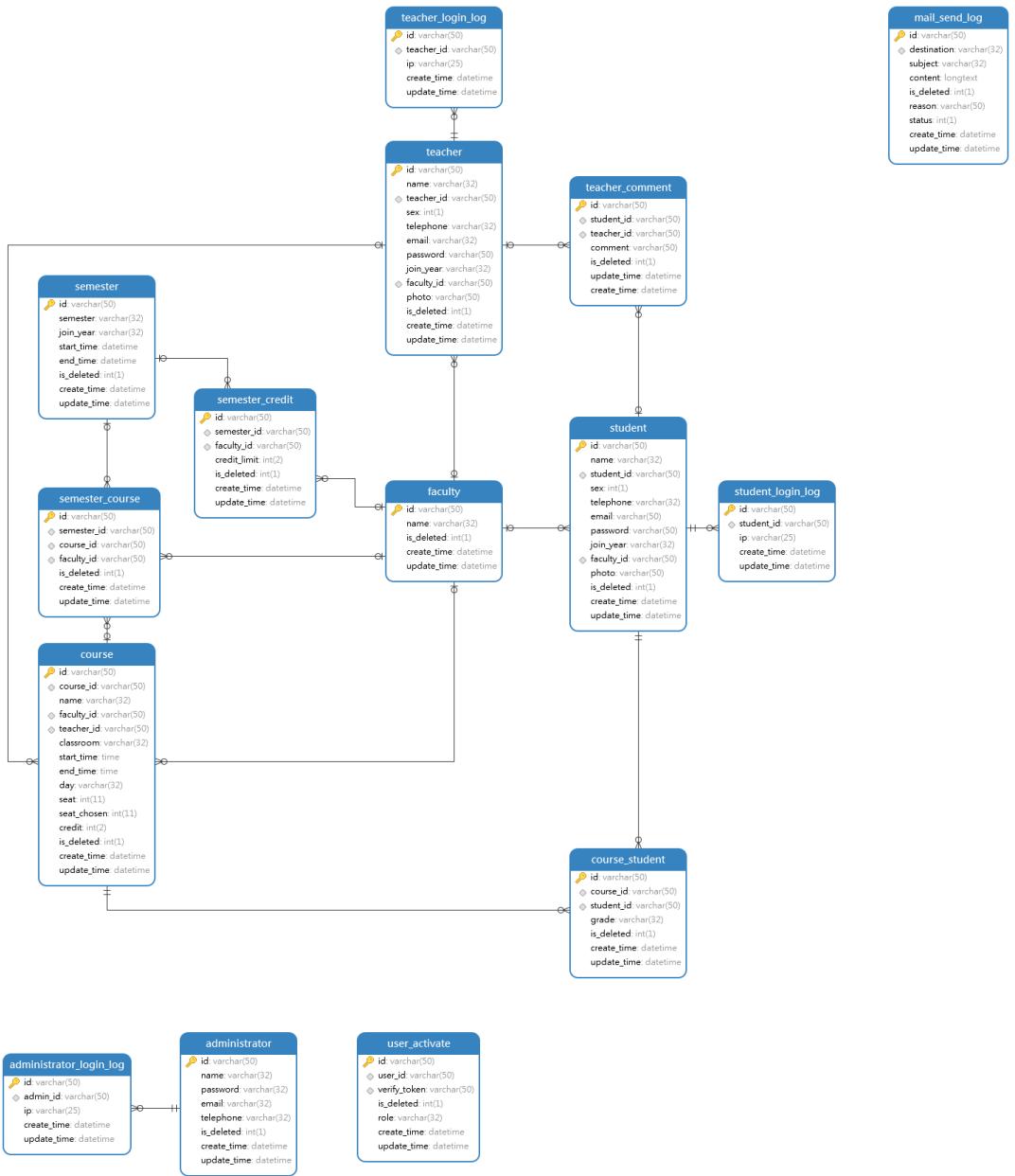


Figure 20: ER Model

3 Development Document

3.1 Development method

3.1.1 Product development method

Our objective is to design a brand-new enrolling system, which implements a lot of additional functions compared to the original system. Thus, we hope to handle them one by one, more exactly, we'd like to develop and test our system under Agile Development.

3.1.2 Pros

- Large project is divided into manageable amounts.
- System developments are coded and tested during each sprint review.
- Works well for fast-moving development projects.
- The individual effort of each team member is visible at any time.

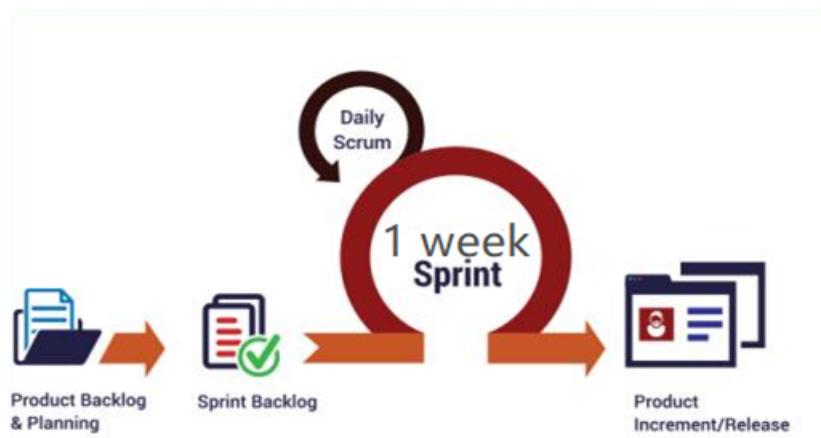


Figure 21: Sprint planning and arrangement

3.2 Team member roles and duties

- **Product Owner: Luo Yemao**

- 1) Close partners with the business and the team to ensure everyone understands the work items in the product backlog.
- 2) Give the team clear guidance on which features to deliver next.
- 3) Decide when to ship the product with a preference towards more frequent delivery.
- 4) Ensure that the development team delivers the most value to the business.

- **Scrum Master: Luo Wei**

- 1) Check to resolve impediments and distractions for the development team, insulating them from external disruptions whenever possible.
- 2) Schedule the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and the sprint retrospective.
- 3) Teach Scrum teams to work more efficiently and produce high-quality products through coaching and mentoring.
- 4) Help Scrum teams understand and adopt self-management.

- **Development team: Luo Yemao, Luo Wei, Lin Longjun, Tian Yunfu**

- 1) The TEAM is responsible for converting the product backlog into potentially shipable increments of functionality at each iteration.
- 2) The TEAM is self-managing, has actual autonomy, and must be culturally compatible, based on motivating people's initiative and avoiding external interference.
- 3) The TEAM has the full right to decide how to convert requirements into product functions, such as whether to do design, what algorithm to use, how to do defect prevention, etc.
- 4) Neither the PO nor the SM has the right to direct the TEAM to realize the requirements, but the TEAM must promise to deliver The paid function is what the PO expects.

3.3 Development Process

3.3.1 Epic Story

According to different users, our epic story is divided into three parts:

- 1) For the system administrator, I need to be able to clearly manage students, teachers, courses and schedule the start and end of enrollment. At the same time, I need the system to ensure 100% data security. On this premise, the system should be able to provide some additional functions, such as data export.
- 2) As far as the teacher is concerned, I need to be able to clearly inquire about the courses I teach, the time and place of classes, and the number of students. At the same time, the system should allow me to grade my students. Finally, I also hope that the system can export some materials I need, such as student information, course information, etc.
- 3) As a student, I was fed up with the poor performance of the old system. At that time, I often could not log in to the system normally. Even after I logged in to the system, I could not choose the course I wanted. For me, it would be more useful to pray to God than waiting in front of the computer on time. So I hope that the new system can have excellent performance and help me quickly choose the course I like.

已完成的问题							在问题“导航器”内查看
关键字	概要	问题类型	优先级	状态	Epic	问题计数 (1)	
SEPROJECT-22	后台项目	↑ Epic	↑ Medium	完成		1	
SEPROJECT-23	学生项目	↑ Epic	↑ Medium	完成		1	
SEPROJECT-26	老师项目	↑ Epic	↑ Medium	完成		1	

Figure 22: Epic Story

3.3.2 User Story

In order to meet the needs of the above three characters, we divided the three parts of epic story into many user stories to facilitate the arrangement of sprints. So as to better complete the development task as soon as possible.

- 1) Manage Faculty: As an admin. I would like to have the ability to create, remove, update and search certain faculty.
- 2) Manage Course: As an admin. I would like to have the ability to create, remove, update and search certain course.
- 3) Manage Account: As an admin. I would like to have the ability to create, remove, update and search certain faculty.
- 4) Account Functions: As a user of the system. I need to log in to the system and log out from the system as well. When there is a threat to my account. I would like the system to inform me. Further more, I need to reset my account in case I forget my password.
- 5) Search Functions: As a student or a teacher. I need to query my chosen or assigned courses. As well as the connected information of the course.
- 6) Add Courses: As a student. I need the right to choose courses, so that I can choose the courses I like.
- 7) Grade Student: As a teacher. At the end of a semester. I have to grade my student. Hence the system need to help me to do so.

3.3.3 Product Backlog

We have decided two parts of project in order to satisfy these three epic stories. First epic story fits in with our first part which is PROJECT_ADMIN which is mainly focused on the management platform. The second and third epic stories fit in with our second part which is PROJECT_COES which is mainly focused on the teacher and student interface.

1. PROJECT_ADMIN

- 1) Manage Faculty
- 2) Manage Course
- 3) Manage Account

4) Account Functions

2. PROJECT_COES

- 1) Account Functions
- 2) Search Functions
- 3) Add Courses
- 4) Grade Student

The screenshot shows a Jira Software Kanban board titled 'COES项目 抢课功能 老师给学生打分'. The board has three columns: 待办 (To Do), 处理中 (In Progress), and 完成 (Done). Each column contains several tasks, each with a checkmark and a small icon.

待办	处理中	完成
SEPROJECT-86 管理员设置抢课开始页面 COES项目	SEPROJECT-85 老师需要给学生打分且学生要先完成抢课 COES项目	SEPROJECT-92 导出XCEL按钮 COES项目
SEPROJECT-90 老师打分页面 COES项目		SEPROJECT-96 学生抢课页面 COES项目
SEPROJECT-91 导出优惠EXCEL COES项目		
SEPROJECT-89 学生抢课接口 COES项目		
SEPROJECT-87 管理员设置抢课开始 COES项目		

Figure 23: Sprint's Kanban (example of sprint 8)

3.3.4 Sprints & Burn Down Charts

- Sprint1 – ADMIN_STRUCTURE_CONSTRUCT

序号	标题	归属	归属未完	优先级	状态
SEPROJECT-9	留学生态和数据			Medium	待办
SEPROJECT-8	抢课后台数据存储模块			Medium	待办
SEPROJECT-7	设计学生优惠券			Medium	待办
SEPROJECT-6	对开源社区数据的技术选型			Medium	待办
SEPROJECT-5	评估Vue3及一些配套的小库			Medium	待办
SEPROJECT-4	更新README.md			Medium	待办

Figure 24: Sprint 1

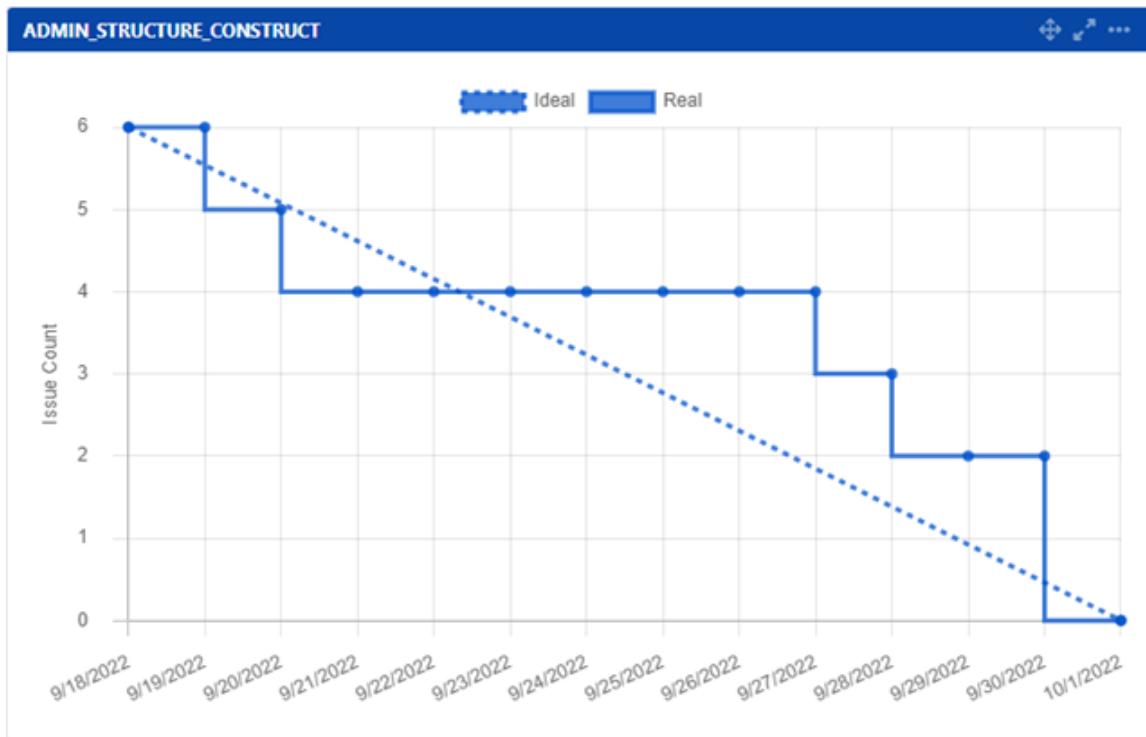


Figure 25: Sprint 1's burn down chart

Planning Meeting	
Topic	Build up basic structure of the project
Time	30 minutes
Discussion	Build the front-end and back-end basic framework, discuss the project architecture and complete the architecture construction.
Conclusion	The front-end framework chooses the naive-admin open source framework and conducts secondary development, and the back-end framework adopts a completely self-developed framework for development.

Review Meeting	
Topic	Assess the rationality of the project architecture
Time	30 minutes
Discussion	Whether the software project division is reasonable, and whether it can accomplish the three goals proposed in our design document.
Conclusion	The performance of the software architecture is reasonable, but the division is too complicated, so we canceled the separate authentication module and notification module. The whole project is divided into two modules, ADMIN and COES.

Retrospective Meeting			
Topic	Review sprint 1		
Time	30 minutes		
Discussion	Whether the tasks assigned by each project member are completed, and whether the amount of tasks assigned to each member is reasonable. Whether the division of labor among members is reasonable.		
Conclusion	The sprint1 task was completed very well, and the workload of the members was relatively balanced.		

- Sprint2 – ADMIN_QUERY

后台项目功能模块 04/10月/22 - 10/19月/22			
关键字	功能	归属模块	优先级
SEPROJECT-13	studentTeacher用户使用教师端口开发	待办	↑ Medium
SEPROJECT-12	学生与教师的交互的接口开发	待办	↑ Medium
SEPROJECT-16	权限Administrator管理	待办	↑ Medium
SEPROJECT-11	学生Springboot的mybatisplus	待办	↑ Medium
SEPROJECT-4	的学生基本操作	待办	↑ Medium
SEPROJECT-14	教师与学生的数据	待办	↑ Medium
SEPROJECT-15	course_student关联带函数	待办	▲ Low
SEPROJECT-3	教师VUE-3-后台的数据	待办	↑ Medium
SEPROJECT-20	用户的添加与删除	待办	↑ Medium
SEPROJECT-24	验证码头部数据对	待办	↑ Medium
SEPROJECT-25	请求学生与后台接口的数据	待办	↑ Medium
SEPROJECT-6	后台管理设计参考	待办	↑ Medium
SEPROJECT-17	administrator登录模块接口开发	待办	↑ Medium
SEPROJECT-10	公告管理页面	待办	↑ Medium
SEPROJECT-21	公告管理模块增加新条件	待办	↑ Medium
SEPROJECT-23	管理员后台需要的数据进行分页查询	待办	↑ Medium

Figure 26: Sprint 2



Figure 27: Sprint 2's burn down chart

Planning Meeting			
Topic	Build the basic query function of the ADMIN project		
Time	30 minutes		
Discussion	Complete the front-end query interface of the ADMIN project for colleges, courses, teachers, and students, and the corresponding query interface for the back-end.		
Conclusion	The query should be made into a paged query structure, and the query interface should support fuzzy query of various information, such as querying the corresponding student according to the first three digits of the student number.		

Review Meeting			
Topic	Assess the rationality of the project query functions		
Time	30 minutes		
Discussion	Evaluate the query functions completed by the front-end and back-end, and evaluate the interface aesthetics, operation convenience and query performance.		
Conclusion	The pre-conceived functions are all completed, the page is more beautiful, and the access time of the back-end interface is also in line with expectations.		

Retrospective Meeting			
Topic	Review sprint 2		
Time	30 minutes		
Discussion	The main task of this sprint is to complete the query function of the ADMIN terminal for courses, teachers, students, and colleges, and complete the above fuzzy query and paging query functions.		
Conclusion	The members of the development team completed their respective tasks very well. Although they encountered some difficulties, they did not give up and finally solved the technical problems. Finally, after evaluation, the results presented meet expectations. Bravo!		

- Sprint3 – ADMIN_EXPORT_EXCEL_CRUD_DEBUG

Sprint3 任务列表及完成度统计表			
任务ID	任务描述	完成度	状态
SEPROJECT-03	教师信息录入与查询	Medium	完成
SEPROJECT-44	增加教师成绩管理模块	Medium	完成
SEPROJECT-43	课程信息录入与查询	High	完成
SEPROJECT-27	教师评价功能实现	Medium	完成
SEPROJECT-45	增加教师评价模块	Medium	完成
SEPROJECT-46	Administrator CRUD	Medium	完成
SEPROJECT-41	FacultyCourses信息查询模块	Medium	完成
SEPROJECT-32	教师评价功能实现	Medium	完成
SEPROJECT-27	StudentInfoTeacher用户数据管理模块	Medium	完成
SEPROJECT-30	教师评价功能实现	Medium	完成
SEPROJECT-29	教师评价功能实现	Medium	完成
SEPROJECT-01	教师评价功能实现	Medium	完成
SEPROJECT-26	管理后台需要对数据进行Excel格式的导入和导出功能模块实现	Medium	完成

Figure 28: Sprint 3



Figure 29: Sprint 3's burn down chart

Planning Meeting	
Topic	Finish export query data to excel file and fix some bugs
Time	30 minutes
Discussion	The user wants to export the query data into an excel file, and we also need to fix some bugs found during the last sprint test.
Conclusion	There is basically no essential difference between the function of exporting excel and the function of querying. You should first fuzzily query the data from the database before exporting excel. As for bugs, the task of fixing bugs is assigned based on the principle that whoever develops is responsible.

Review Meeting	
Topic	Assess the rationality of the project export functions and bug status
Time	30 minutes
Discussion	Whether it is safe to export data, whether additional security measures need to be added to ensure data security, and let testers conduct regression tests to check bug fixes.
Conclusion	First of all, all the bugs have been fixed normally, and after discussion, we decided to set a random password for each exported excel, and send these passwords to the email address of the corresponding export administrator.

Retrospective Meeting		
Topic	Review sprint 3	
Time	30 minutes	
Discussion	Whether the task of adding a password to excel is completed, and whether this can effectively ensure data security.	
Conclusion	First of all, we think it is very good for us to think about data security, because the top of our development priorities is data security, and secondly, we think there is no good way to limit human-intentional data leakage, so for excel Adding a password is our best effort.	

• Sprint4 – ADMIN_AUTHORIZATION_EXTRA_FUNCTIONS

Sprint4 – ADMIN_AUTHORIZATION_EXTRA_FUNCTIONS		开始时间	结束时间	状态
SEPROJECT-53	完成	2022-10-26	2022-10-27	Medium
SEPROJECT-47	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-51	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-48	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-58	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-49	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-50	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-49	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-40	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-55	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-62	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-62	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-56	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-51	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-57	待办事项	2022-10-26	2022-10-27	Medium
SEPROJECT-54	待办事项	2022-10-26	2022-10-27	High
SEPROJECT-41	待办事项	2022-10-26	2022-10-27	Medium

Figure 30: Sprint 4



Figure 31: Sprint 4's burn down chart

Planning Meeting	
Topic	Discuss the authentication of user login and modifying password in ADMIN
Time	30 minutes
Discussion	The user need to modify their password so that the system will keep personal information safe. And different user role use different functions, so the system should identify who they are and what they can do.
Conclusion	The authentication should use tokens to identify the status of users, and the token should be saved in database with expiration.

Review Meeting	
Topic	Show the method realize the authentication of users
Time	30 minutes
Discussion	We use redis to store dynamic tokens with role-prefix and Id to identify different users, and use MySQL to store the static verify-tokens with Id to verify whether the user is firstly login. If so, the first-login user should modify their password. The dynamic tokens is set with a expiration.
Conclusion	The system should identify whether the token is in redis when a user logs in. If the user logs in repeatedly, the token will not change before it expires. We also should limit the number of the user logins within a specified time.

Retrospective Meeting	
Topic	Review sprint 4
Time	30 minutes
Discussion	What we did and didn't go well with actions to make the next sprint better, and whether the system is feasible when multiple users log in at the same time or the same user logs in multiple times
Conclusion	What we commit for this sprint does do a great job in practice, the division of labor among members is also reasonable.

- **Sprint5 – COES_STRUCTURE_CONSTRUCT_BASIC_FUNCTIONS**



Figure 32: Sprint 5



Figure 33: Sprint 5's burn down chart

Planning Meeting	
Topic	Discuss the structure of user terminal, COES and realize the function of counting the logging in users as well as the number of active users in ADMIN.
Time	30 minutes
Discussion	According to the user's needs, what functions the client needs, and what information does the database need to store. Front-end and back-end programmers need to communicate with the framework. In ADMIN system, the administrators need to know the status of users.
Conclusion	Refer to the requirements document, determine the required APIs, and the required database tables. Then to build up the frame. The users status should depends on the login and logout in COES.
Review Meeting	
Topic	Show the structure of user terminal, COES, and the function of counting the logging in users as well as the number of active users in ADMIN.
Time	30 minutes
Discussion	The system allows teachers and students to search their course information. Students can also select or delete their course, give comments and so on, while teachers can search for students involved in the course, and set their grade,etc. The student grades and comments should be stored in data base. The logic of login and logout functions of teachers and students is similar with the administrators'. So we conclude user status from dynamic tokens which is recorded when login and deleted when logout.
Conclusion	We summarized the APIs above and the tables needed in the structure, used folders to group for the framework. And the functions of ADMIN have been realized.

Retrospective Meeting	
Topic	Review sprint 5
Time	30 minutes
Discussion	What we did and didn't go well with actions to make the next sprint better, and whether the frame is suitable as well as whether the users status we get is correct.
Conclusion	What we commit for this sprint is clear and reliable, the division of labor among members is also reasonable. The functions did a good job.

- Sprint6 – COES_AUTHORIZATION_BASIC_FUNCTIONS



Figure 34: Sprint 6



Figure 35: Sprint 6's burn down chart

Planning Meeting			
Topic	Discuss the authentication of users in COES and the basic functions.		
Time	30 minutes		
Discussion	The logic of the authentication is similar with the one in ADMIN(sprint 4), but we should add the interceptors to intercept users' information to identify whether they have the right to use some function. And function of searching selected courses for students should be realized in this sprint.		
Conclusion	We should complete the logging in and out, the interceptors, and selected courses for students. The interceptors get users' tokens when logging in.		

Review Meeting			
Topic	Show the authentication of users in COES and the basic functions.		
Time	30 minutes		
Discussion	The function of logging in and out is similar with the one in ADMIN(sprint 4). The interceptors intercept users' tokens to get users' id. Students can get their selected courses information from database by matching their id.		
Conclusion	Every requests will take the tokens in its header, so functions which need authorization can be used correctly.		

Retrospective Meeting			
Topic	Review sprint 6		
Time	30 minutes		
Discussion	What we did and didn't go well with actions to make the next sprint better, and whether the students can get information correctly.		
Conclusion	It is a good idea that the interceptors can intercept users' information, and every requests can be read to get userId, which not only can direct student or teacher to their different interfaces, but also can identify different users to get their own information.		

• Sprint7 – COES_DEBUG DESIGN_ENROLLMENT



Figure 36: Sprint 7



Figure 37: Sprint 7's burn down chart

Planning Meeting	
Topic	Debug front-end and back-end and design students' enrollment
Time	30 minutes
Discussion	Front-end should switch to connect back-end from mock while it may have some problems. The students' enrolment is one of the most important functions in the system.
Conclusion	Front-end and back-end programmers should communicate to solve the problems when debugging. And we should determine how to design the students' enrollment.
Review Meeting	
Topic	Solve problems when debugging and show the design of students' enrollment
Time	30 minutes
Discussion	The real returns from back-end sometimes differ from which mock simulated. We communicate and modify the codes in front-end and back-end so that the system has worked well. The enrollment is designed like this: We store a time table set from administrator. The different rows stores different time periods and which grade students are able to select in the time. And another table will store the course can be selected with the id of time table, the remain number and faculty constraint. The student want to select courses will be identified by grades. What he/she can select is limited by his/her faculty. The request to select will be stored in redis for high efficiency. The system will return successful or not by the remain number.
Conclusion	The debugging can be effective by communicating. The design of enrollment uses both MySQL and redis.

	Retrospective Meeting
Topic	Review sprint 7
Time	30 minutes
Discussion	What we did and didn't go well with actions to make the next sprint better, and whether the enrollment design can be optimized.
Conclusion	The communication between front-end and back-end programmers make us know each other's work better. But we don't have a optimization of the enrollment design.

- Sprint8 – COES/ADMIN_ENROLMENT_GRADE

The screenshot shows a project management interface for 'COES项目' (COES Project). At the top, there are tabs for '待办功能' (Pending Functions), '老师给学生打分' (Teachers giving grades to students), and '问题' (Issues). Below the tabs, there are icons for '添加' (Add), '编辑' (Edit), '删除' (Delete), and '...'. A search bar is present. The main area displays a list of tasks:

- SEPROJECT-05 老师需要给学生打分且学生要能完成检测 (checked)
- SEPROJECT-06 管理员设置招课开始页面 (checked)
- SEPROJECT-09 老师打分页面 (checked)
- SEPROJECT-92 导出EXCEL按钮 (checked)
- SEPROJECT-08 学生选课页面 (checked)
- SEPROJECT-91 导出信息EXCEL (checked)
- SEPROJECT-09 学生给便接口 (checked)
- SEPROJECT-07 管理员设置招课开始 (checked)

To the right, a grid shows the status of these tasks across different team members, each represented by a small profile icon and labeled 'COES项目'.

Figure 38: Sprint 8



Figure 39: Sprint 8's burn down chart

	Planning Meeting
Topic	Discuss the teacher-giving-grade and students' enrollment.
Time	30 minutes
Discussion	Teachers need to give grades to student as a basic function. Administrators need to set the enrollment start time and end time. Students need to select courses or delete what they have chosen. They all need interfaces in front-end also.
Conclusion	This sprint we will realize the function of the teacher-giving-grade and students' enrollment in COES as well as the supervision of enrollment in ADMIN.

Review Meeting	
Topic	Show the method realize the teacher-giving-grade and students' enrollment
Time	30 minutes
Discussion	We add an entry to the course-student table in order to store grades. In addition, we create two tables to record the start and end time of enrollment, and another decide which course can be selected by students from which faculty. For enrolment, we use redis transaction and semaphore as lock for adding or deleting a course.
Conclusion	The operation we use in redis really grantee the digital security and is necessary.

Retrospective Meeting	
Topic	Review sprint 8
Time	30 minutes
Discussion	What we did and didn't go well with actions to make the next sprint better, and whether the system is feasible when multiple students enrolling at the same time.
Conclusion	What we commit for this sprint does do a great job in practice, the division of labor among members is also reasonable.

- Daily stand-up meeting (excerpt)

		Daily Stand-Up Meeting		
Date	Name	Done	Undone	Trouble
10.15.2022	Luo Yemao	17	23	
	Lin Longjun	20	25	
	Luo Wei	24		
	Tian Yunfu		10	Unable to display the data returned by the back-end.
10.21.2022	Luo Yemao	33,42,43	26	
	Lin Longjun	44,46	27,37	Have trouble reading EasyExcel official doc.
	Luo Wei	29,30		Try to figure out how to store token received from login function and add it as a header field to every request.
	Tian Yunfu	31,32	45	
10.28.2022	Luo Yemao	40,47,48	41,53	Try to configure our mail server using our own domain.
	Lin Longjun	60,61	62	
	Luo Wei	50,52	54,55,56	Have trouble in coding the global response code interceptor
	Tian Yunfu	49,51	57,58,59	
11.11.2022	Luo Yemao	72	70	
	Lin Longjun	74,75	76	
	Luo Wei	71	73	Need to use another third party api to display the return data as charts. Which takes time to learn.
	Tian Yunfu	77		
11.18.2022	Luo Yemao	83,84	78	
	Lin Longjun	79	81	How to ensure the performance and safety of the logic of controlling students to grab classes is very difficult and requires a lot of thinking.
	Luo Wei	93		
	Tian Yunfu	82		

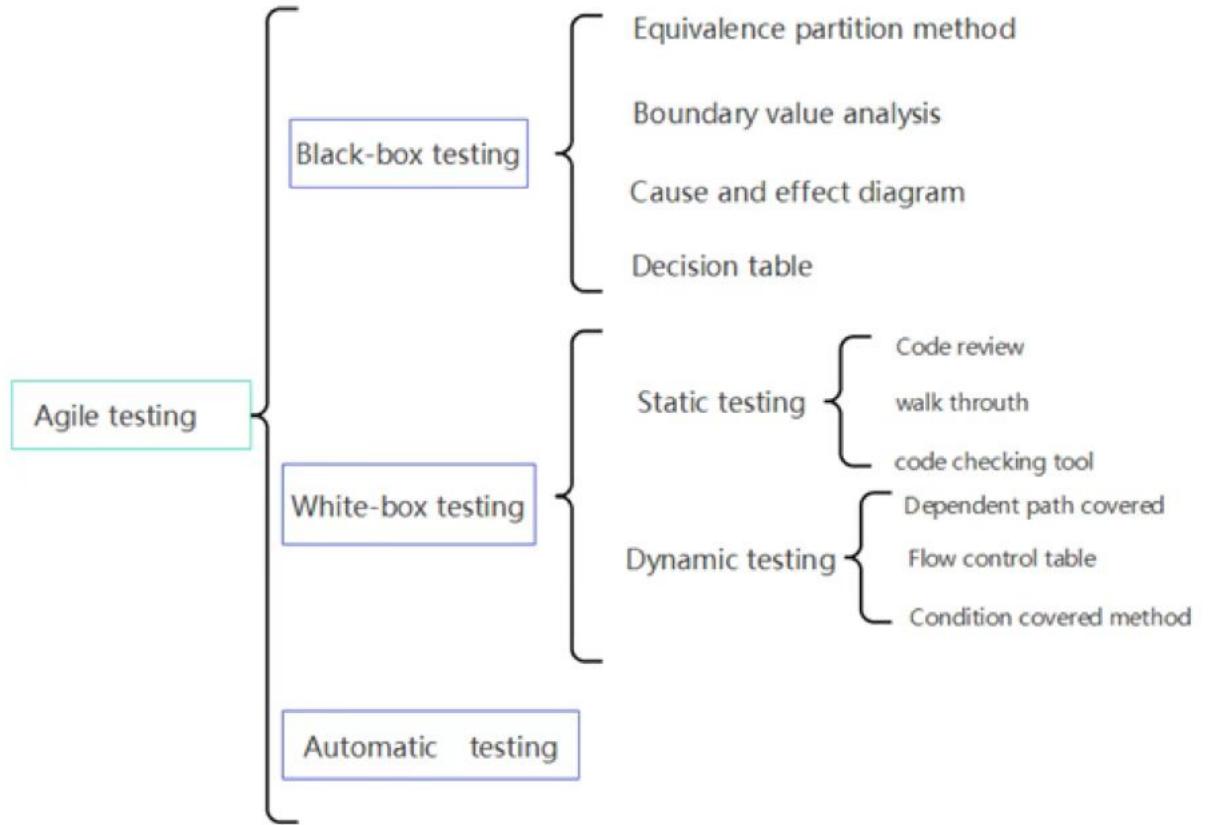
4 Testing Document

Software testing is a process of software execution, and it aims to discover and correct the mistakes in the software to improve the reliability, correctness, and safety of the software . Guided by defect prevention, testing is to demonstrate that the software meets the design requirements, finds defects, and prevents defects. How to execute software testing may be a hard problem in the software life cycle. After confirming the requirement of users and discussion in the software QA engineers group, we chose the test method of agile testing. We chose agile testing for several reasons:

- 1) The project is equivalent to parallel development and testing, and the overall project time is faster.
- 2) Clear division of work tasks and high work efficiency.
- 3) It is found that the BUG can be solved in a short time, and the impact on the test of related modules is relatively small.
- 4) Be able to communicate with developers in a timely manner, and test throughout, reducing the amount of BUG.

After the development started on September 20th, we started testing and continued until the final presentation on December 13th.

I divided the agile testing section into three sub-headings, they are black box testing, white box testing, and Automatic testing. Black box testing is all dynamic testing, and white box testing is a combination of static testing and dynamic testing. In the black box test, I used four methods to get the final functional test cases. They are the equivalence class division method, boundary value analysis method, causality diagram method, and decision table method. In the software COESPLUS, the function of rolling course, points-scoring, and login is making great effects on the whole system, so in black-box testing, we pay the most attention to those three parts. And in the white box static testing, we conducted in the way of code reviewing, code walk-through, and code checking tools. In the white box dynamic testing, I used three methods to design the testing case. They are condition covered method, path covered method, and flow control table. The following figure can see our test catalog more clearly:



4.1 Black-box testing

4.1.1 Equivalence Partition

Equivalence class division is an important and commonly used black box testing method, which reasonably classifies the inexhaustible testing process, so as to ensure the integrity and representatives of the designed test cases. The equivalence class division method is to divide all possible input data, that is, the input of the program into several parts (subsets), and then select a small amount of representative data from each subset as test cases. For the reason, it is impossible to implement exhaustive testing. We need to select some representative data from a large number of possible data as test cases. After category division, the representative data of each category plays an equal role in the test to other values in this category.

When designing test cases, we divide equivalence classes based on requirements description and list equivalence tables to determine test cases. Here we test the page query function of the student table on our Admin system. In this part, we divide the input condition into 4 parts: student name, student ID, the faculty of the student, and student join year. These four parameters have different rules and limits, in detail, we made such table:

Input Equivalence Class	Valid Equivalence Class	Invalid Equivalence Class
Student Name	1. string excluding special character or number	2. string containing special characters
	3. fill	4. string containing numbers
	5. student exists	6. Student not exist in data base
Student ID	7. 18 number and letter characters including '-'	8. string not containing numbers or letters
	9. 9 th and 14 th character should be '-'	10. string length greater than 18 11. the '-' character is in disorder
	12. in list input	13. illegal input 14. input out of the faculty list
Faculty of student	15. string in year style	16. string containing other characters
	17. string only containing 4 numbers	18. string length over 4 characters
	19. join year should before present year	20. join year is later than present year

Figure 40: Equivalence Partition

You can see the division of valid equivalence classes and invalid equivalence classes in detail. Following are the testing cases based on equivalence classes.

Order number	Input parameter	Coverage Equivalence Class	Expected Output
1.	黎杰宏	1, 3, 5	Display the info of all students named 黎杰宏
2.	Linda Foster; ; 2022	1, 17, 19, 21	Display the info of student 黎杰宏 who join the university at 2022
3.	卢秀英; ; 法学院 ; 2020	1, 3, 5, 13	Display the info of all students named 卢秀英 in 法学院 who join at 2020
4.	Marry	6	Not Such Student
5.	@!@; 2109202C-E111-6073 ;	2, 6, 7, 8	Please input correct student name

Figure 41: Equivalence Partition Test Case

4.1.2 Boundary Value Analysis

Many program errors occur on the boundary of the input or output range. Therefore, setting test cases for various

boundary conditions can find many program defects. Boundary value analysis is different from equivalence class division: Boundary value analysis does not randomly pick one from a certain equivalence class as a representative, but each boundary of this equivalence class must be used as a test condition; boundary value analysis not only considers Input conditions but also consider the test conditions generated by the output space. Based on the principle of boundary value analysis, we tested the function of points-scoring.

Score input

Test requirements: input a student's score n to judge which grade is suitable.

Analysis: Determine effective area and invalid area; Critical point: 0, 50, 100

A score less than 0 or 100 would be judged to invalid value, 0 to 50 is fail, and 50 to 100 is pass.

Number of case	Value	Expectation of output
1	-1	Invalid value
2	0	Fail
3	1	Fail
4	59	Fail
5	60	Pass
6	61	Pass
7	99	pass
8	100	Pass
9	101	Invalid

Figure 42: Boundary Value Test case

4.1.3 Cause and Effect Diagram

The equivalence class division method and boundary value analysis method both focus on the input conditions but do not consider the various combinations of input conditions and the mutual constraints between the input conditions. In this way, although various input conditions may be wrong and have been tested, the combination of multiple input conditions that may be wrong has been ignored.

If various combinations of input conditions must be considered during testing, the number of possible combinations will be astronomical. Therefore, a form suitable for describing a combination of multiple conditions and correspondingly generating multiple actions must be considered for testing cases. Design, which requires the use of cause and effect diagrams.

In the causality diagram, the cause is the input and the result is the output. Based on the principle of Cause and Effect diagram, we design the following diagram for the function of login and setting course selection by administrators.

Login

Reason:

- C1:Missing username input
- C2:Missing password input
- C3:password does not match username
- C4:password does not match username

Result:

- e1:Display information "Please input username"
- e2:Display information "Please input password"
- e3:Display information "User name or Password is wrong"
- e4:Login in system

In the function of login, only if the user input username and password, where the password matches its username, the system would give the permission to login.

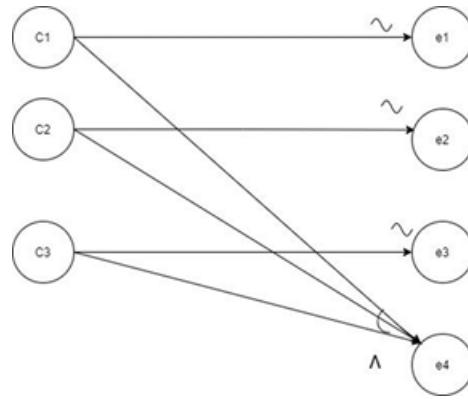


Figure 43: Cause and Effect Diagram of Login

Administrator set course selection

Reason:

C1:Input faculty name

C2:Input student grade

C3:Input start time

C4:Input end time

Result:

e1:Display information "Please input faculty name"

e2:Display information "Please input student grade"

e3:Display information "Please input the start time of class selection"

e4:Display information "Please input the end time of class selection"

e5:Setup successful

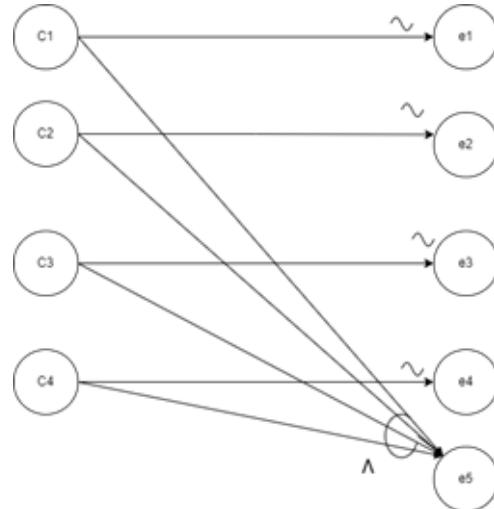


Figure 44: Cause and Effect Diagram of Set Course Selection

4.1.4 Decision Table Testing

A decision table is a tool for analyzing and expressing the execution of different operations under multiple logic conditions. Among all the black box testing methods,

the testing based on the decision table is the most rigorous and logical testing method. The advantage of this method is that it can enumerate all complex problems according to various possible situations, concisely, and avoid omissions. Therefore, a complete set of test cases can be designed using the decision table.

According to the Cause and Effect diagram, we designed a decision table for the function of login and setting course selection.

Decision tables(login)					
	1	2	3	4	
Input user name	Y	N	Y	Y	
Input password	Y	-	N	Y	
password match username	Y	-	-	N	
Display information "Please input user name"		✓			
Display information " Please input the password"			✓		
Login	✓				
Display information "User name or Password is wrong"				✓	

Decision tables(administrator set course selection)					
	1	2	3	4	5
Input faculty name	Y	N	Y	Y	Y
Input student grade	Y	-	N	Y	Y
Input start time	Y	-	-	N	Y
Input end time	Y	-	-	-	N
Display information "Please input faculty name"		✓			
Display information "Please input student grade"			✓		
Display information " Please input the start time of class selection				✓	
Display information " Please input the end time of class selection					✓
Setup successful	✓				

Figure 45: Decision Table

4.2 White-box Testing

4.2.1 Static White Box Testing

4.2.1.1 Code Review

Code review is a process of increasing the quality of code in the way of rechecking the code by others in the life cycle of software. Code review requires members consciously and systematically to check other group members' code. We would hold a code review in the retrospective meeting in every sprint. The code developer would introduce the code written by him and the leader would make comments on Gitlab or point out the mistake in the face. After passing the review by the leader the code would be permitted to be merged into the master branch.

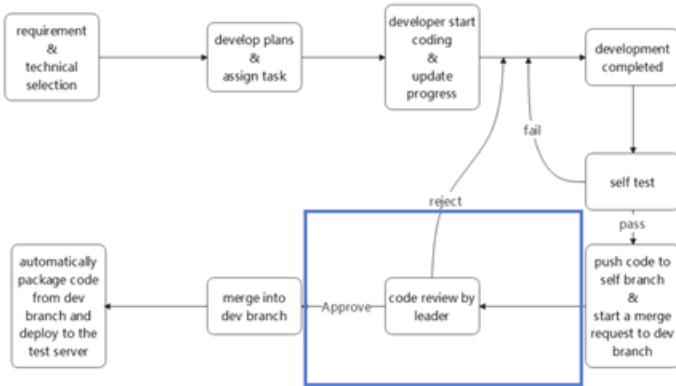


Figure 46: Process of Code Review

4.2.1.2 Code Walk-through

In the process of software development, we would hold code walk-throughs irregularly and the walk-through in fronted-end and back-end is separated. The code would be introduced by developers and inspected by other developers in the group to find the error, including the type of data definition, the effect field of variables, and the method of conveying parameters.

罗叶茂 @YemaLuo 2周前 大佬们好，关于这个主题 2星期内
最近更新 by 林珑君 2星期前

[coesplus-coes/src/main/java/com/coesplus/coes/controller/teacher/TeacherCourseController.java](#) 0 → 100644

```

79 +         });
80 +         log.info(currentPage + ":" + pageSize + ":" + courseVoList);
81 +         page.setRecords(courseVoList);
82 +         return Result.ok(page);
83 +     } catch (Exception e) {
84 +         log.error(e.getMessage(), e);
85 +         return Result.error(e.getMessage());
86     }
87 }
88 /**
89 * 教师设置学生成绩
90 */
91 @PatchMapping()
92 public Result giveGrade(@RequestParam("courseId") String courseId, @RequestParam("studentId") String
studentId
93 , @RequestParam(value = "isDeleted", required = false) String isDeleted, @RequestBody String
grade) {
  
```

罗叶茂 @YemaLuo · 2星期前
看不懂，为什么要isDeleted，isDeleted一定要为1，难道管理员禁用了课程教师还能修改吗？最好这三个都用requestbody传，而且代码中的传参方法和api文档不一致。需要修改！

Figure 47: Walk-through

4.2.1.3 Static Testing Tools

In our development we use Eslint to automatically check the code grammar error

and use Alibaba code Guideline to standardize the coding style



The screenshot shows a code editor with a static code analysis tool overlay. The code is a Vue.js component with the following structure:

```
</n-car>
</n-carou>
</div>

<!--访问量 | -->
<VisiTab />
<!-- -->
</div>
</template>
<script lang=>
    import { ref, onMounted } from 'vue';
    // import { getConsoleInfo } from '@/api/dashboard/console';
    import VisiTab from './components/VisiTab.vue';

    onMounted( hook: async () => {

```

An ESLint error is displayed at the top right: "ESLint: 'ref' is defined but never used. Allowed unused vars must match /^.+u.(@typescript-eslint/no-unused-vars)". Below the error message are some keyboard shortcuts: 移除未使用的元素 Alt+Shift+Enter, 更多操作... Alt+Enter.

Figure 48: Static Code checking Tool

4.2.2 Dynamic White Box Testing

Static white box testing is the process of carefully reviewing software design, architecture, and code in an orderly manner without executing software, so as to find out software defects, sometimes called structural analysis.

As for the advantages, The primary reason is to find software defects as early as possible to find software defects that are difficult to find or isolate by dynamic black box testing. It is very valuable to let the test team focus on the review of software design at the beginning of the development process. Another reason is to provide ideas for black box testers to design and apply test cases when accepting software for software testing.

4.2.2.1 Control Flow Diagram

The control flow diagram is an abstract representation of a procedure or program and an abstract data structure used in the compiler, maintained internally by the compiler, representing all paths traversed during the execution of a program. It represents the possible flow direction of all basic block execution in a process in the form of a diagram, and can also reflect the real-time execution of a process.

To better understand the internal logic, we design the flow control table and we take the function of login and making comments as examples. Because the line of each function's code is far away, we use the description of the function instead of the number of the code line.

flow chart of login system

the ring complexity of the login flow chart is:6

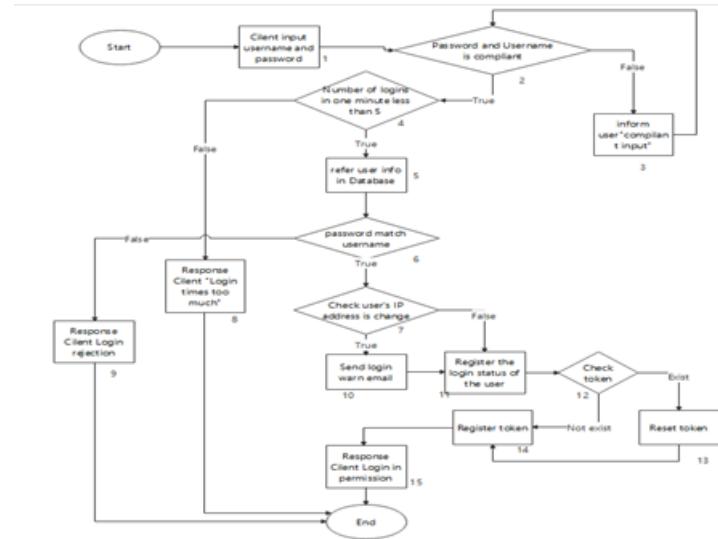


Figure 49: Flow Chart of Login

flow chart of comment function

the ring complexity of the comment function is:4

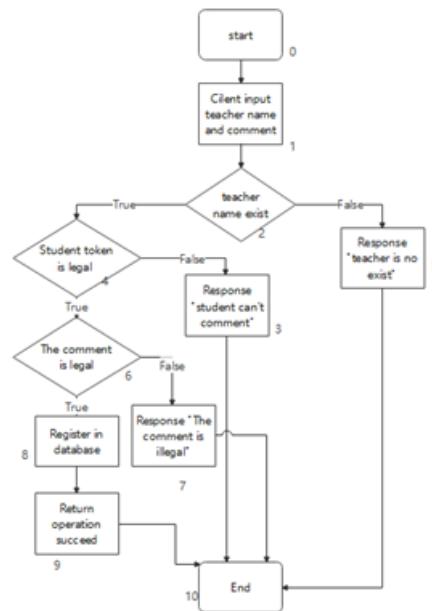


Figure 50: Flow Chart of Making Comment

4.2.2.2 Independent Path Coverage

Independent path testing is based on the program control flow diagram, by analyzing the loop complexity of the control structure, and deriving the basic executable path collection, so as to design the corresponding test case method. The designed test cases should ensure that the statements of the program cover 100% and the conditions cover 100% in the test. Based on the flow control table, we conclude the dependent paths:

In the **login** function, we have 6 paths:

Path1: 0-1-2-4-5-6-7-10-11-12-14-15-16

Path2: 0-1-2-3-2-4-5-6-7-10-11-12-14-15-16

Path3: 0-1-2-4-8-16

Path4: 0-1-2-4-5-6-7-10-11-12-13-14-15-16

Path5: 0-1-2-4-5-6-7-11-12-13-14-15-16

Path6: 0-1-2-4-5-6-9-16

In the **making comments** function, we have 4 paths:

Path1: 0-1-2-4-3-10

Path2: 0-1-2-5

Path3: 0-1-2-4-6-8-9-10

Path4: 0-1-2-4-6-7-10

4.2.2.3 Condition Coverage Method

The decision and conditional coverage meet the need for both decision coverage and conditional coverage, but it cannot guarantee that the combination of conditions within all decisions of the tested code is covered so we need to use conditional combination coverage to cover the combination of the conditions of the decisions in the tested code.

In the function of **login**, we have 5 conditions:

Password input is compliant, true is T1, false is -T1

Number of logins in one minute less than 5, true is T2, false is -T2

Password matches username, true is T3, false is -T3

User IP address is changed, true is T4, false is -T4

The token exists, true is T5, false is -T5

In the function of **making comments**, we have 3 conditions:

Teacher name exist, true is T1, false is -T1

Student token is legal, true is T2, false is -T2

Comment is legal, true is T3, false is -T3

4.2.2.4 Testing Cases of Login System

For the analysis of the above conditional coverage and path coverage, we design the following test cases.

Login Part

We assume the database has stored the following information:

Token	0094de1e48e461040e68ec24c9f1f391
username	tiffany
password	123456
Previous IP address	192.168.1.8

Case number	Case	Execution path	Covered condition
1	Username: tiffany Password:123456 Login-Count=2 IP address:60.246.49.9 Token: 056bdb20b4ed7516941407284ffed9a1	0-1-2-4-5-6-7-10-11-12-14-15-16	T1 T2 T3 T4 -T5
2	First time Username: tiffany Password:1234566666666666 Second time: Username: tiffany Password:123456 Login-Count=2 IP address:60.246.49.9 Token: 056bdb20b4ed7516941407284ffed9a1	0-1-2-3-2-4-5-6-7-10-11-12-14-15-16	T1 T2 - T2 T3 T4 -T5
3	Username: tiffany Password:123456 Login-Count=5 IP address:60.246.49.9 Token: 056bdb20b4ed7516941407284ffed9a1	0-1-2-4-8-16	T1 -T2
4	Username: tiffany Password:123456 Login-Count=2 IP address:60.246.49.9 Token: 0094de1e48e461040e68ec24c9f1f391	0-1-2-4-5-6-7-10-11-12-13-14-15-16	T1 T2 T3 T4 T5
5	Username: tiffany Password:123456 Login-Count=2 IP address: 192.168.1.8 Token: 056bdb20b4ed7516941407284ffed9a1	0-1-2-4-5-6-7-11-12-13-14-15-16	T1 T2 T3 -T4 -T5
6	Username: tiffany Password:654321 Login-Count=2 IP address:60.246.49.9 Token: 056bdb20b4ed7516941407284ffed9a1	0-1-2-4-5-6-9-16	T1 T2 - T3

Figure 51: Test Case of Login

Comment Part

We assume the database has stored the following information:

Teacher name	Tony		
Legal token	0094de1e48e461040e68ec24c9f1f391		
Case number	Case	Execution path	Covered condition
1	Teacher name: Tony Token: 77a4fd0ec3aca4430b1375cad55e519 Comment: "Teacher is nice"	0-1-2-4-3-10	T1 -T2
2	Teacher name: Lucy Token: 0094de1e48e461040e68ec24c9f1f391 Comment: "Teacher is nice"	0-1-2-5	-T1
3	Teacher name: Tony Token: 0094de1e48e461040e68ec24c9f1f391 Comment: "Teacher is nice"	0-1-2-4-6-8-9-10	T1 T2 T3
4	Teacher name: Tony Token: 0094de1e48e461040e68ec24c9f1f391 Comment: "Teacher is so foolish"	0-1-2-4-6-7-10	T1 T2 -T3

Figure 52: Test Case of Comment

4.3 Automatic Testing

Automated testing refers to the process of converting human-driven test behavior into a machine program to drive testing, from generating some test data through scripts to generating test scripts and driving test behaviors based on test cases.

4.3.1 Visual Bug Testing

At the front-end part, we use Selenium to execute automatic testing to find the visual bugs.

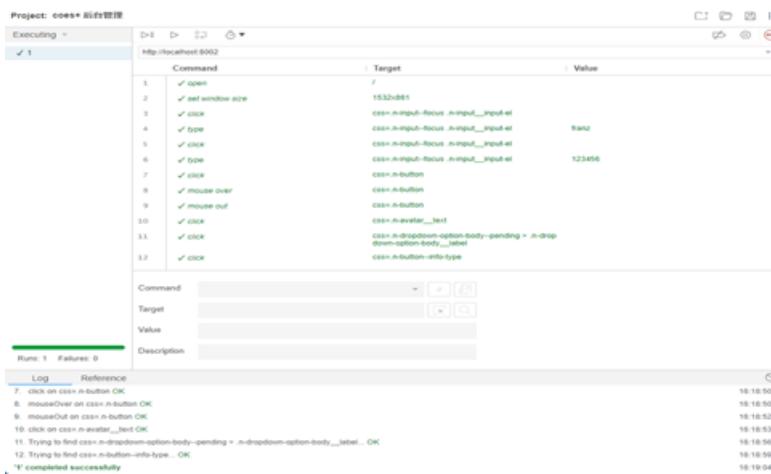


Figure 53: Visual Bug Testing by Selenium

4.3.2 SQL Injection Testing

SQL injection is through a web client, through user input data entry, inserting or 'injecting' partial or complete SQL statements. A successful SQL injection attack can

read sensitive data in the database, modify the database (insert, delete, update), perform administrative operations on the database, etc. We want to keep the database secure, so we want to execute SQL injection tests.

```

root@YemaoluoServer:~/jupyter# sqlmap -u "http://yemaoluo.top:7000/api/admin/teach
geSize=10&name=&teacherId=&faculty=&joinYear=&telephone=&isDeleted="
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual c
It is the end user's responsibility to obey all applicable local, state and federa
assume no liability and are not responsible for any misuse or damage caused by thi
[*] starting @ 16:31:58 /2022-12-12/
[16:31:58] [WARNING] provided value for parameter 'name' is empty. Please, always
meter values so sqlmap could be able to run properly
[16:31:58] [WARNING] provided value for parameter 'teacherId' is empty. Please, al
parameter values so sqlmap could be able to run properly
[16:31:58] [WARNING] provided value for parameter 'faculty' is empty. Please, alwa
parameter values so sqlmap could be able to run properly
[16:31:58] [WARNING] provided value for parameter 'joinYear' is empty. Please, alw
parameter values so sqlmap could be able to run properly
[16:31:58] [WARNING] provided value for parameter 'telephone' is empty. Please, al
parameter values so sqlmap could be able to run properly
[16:31:58] [WARNING] provided value for parameter 'isDeleted' is empty. Please, al
parameter values so sqlmap could be able to run properly
[16:31:59] [INFO] testing connection to the target URL
[16:31:59] [INFO] testing if the target URL content is stable
[16:31:59] [INFO] target URL content is stable
[16:31:59] [INFO] testing if GET parameter 'currentPage' is dynamic
[16:31:59] [WARNING] GET parameter 'currentPage' does not appear to be dynamic
[16:31:59] [WARNING] heuristic (basic) test shows that GET parameter 'currentPage'
table
[16:31:59] [INFO] testing for SQL injection on GET parameter 'currentPage'
[16:31:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:31:59] [INFO] testing 'Boolean-based blind - Parameter replace (original value
[16:31:59] [INFO] testing 'MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY
(FLOOR)'
[16:31:59] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[16:31:59] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or
Oracle AND error-based - WHERE or HAVING clause (XMLEtyp
[16:31:59] [INFO] testing 'MySQL > 5.0 error-based - Parameter replace (FLOOR)'
[16:31:59] [INFO] testing 'Generic inline queries'
[16:31:59] [INFO] testing 'PostgreSQL > 9.1 stacked queries (comment)'
[16:31:59] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[16:31:59] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - com
[16:31:59] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[16:31:59] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[16:31:59] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[16:31:59] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one o
chnique found. Do you want to reduce the number of requests? [Y/n] y
[16:32:02] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:32:02] [WARNING] GET parameter 'currentPage' does not seem to be injectable
[16:32:02] [INFO] testing if GET parameter 'pageSize' is dynamic
[16:32:05] [INFO] testing 'MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY
or GROUP BY clause (FLOOR)'
[16:32:05] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[16:32:05] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or
HAVING clause (IN)'
[16:32:05] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLEtyp
e)'
[16:32:05] [INFO] testing 'MySQL > 5.0 error-based - Parameter replace (FLOOR)'
[16:32:05] [INFO] testing 'Generic inline queries'
[16:32:05] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[16:32:05] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[16:32:05] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - com
ment)'
[16:32:05] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[16:32:05] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[16:32:05] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[16:32:05] [INFO] testing 'Oracle AND time-based blind'
[16:32:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:32:05] [INFO] testing if GET parameter 'telephone' does not seem to be injectable
[16:32:05] [INFO] testing if GET parameter 'isDeleted' is dynamic
[16:32:05] [WARNING] GET parameter 'telephone' does not appear to be dynamic
[16:32:05] [WARNING] GET parameter 'isDeleted' does not appear to be dynamic
[16:32:05] [WARNING] heuristic (basic) test shows that GET parameter 'isDeleted' m
ight not be injectable
[16:32:05] [INFO] testing for SQL injection on GET parameter 'isDeleted'
[16:32:05] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:32:05] [INFO] testing 'Boolean-based blind - Parameter replace (original value
)'
[16:32:06] [INFO] testing 'MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY
or GROUP BY clause (FLOOR)'
[16:32:06] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[16:32:06] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or
HAVING clause (IN)'
[16:32:06] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLEtyp
e)'
[16:32:06] [INFO] testing 'MySQL > 5.0 error-based - Parameter replace (FLOOR)'
[16:32:06] [INFO] testing 'Generic inline queries'
[16:32:06] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[16:32:06] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[16:32:06] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - com
ment)'
[16:32:06] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[16:32:06] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[16:32:06] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[16:32:06] [INFO] testing 'Oracle AND time-based blind'
[16:32:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:32:06] [INFO] testing if GET parameter 'isDeleted' does not seem to be injectable
[16:32:06] [WARNING] GET parameter 'isDeleted' does not appear to be injectable
Try to
increase values for '--level'/'--risk' options if you wish to perform more tests.
If you suspect that there is some kind of protection mechanism involved (e.g. WAF
) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and
/or switch '--random-agent'
```

Figure 54: SQL Injection Testing

4.3.3 Back-end Concurrent Testing

In order to check whether the system has concurrency problems, such as memory leaks, thread-locks, resource contention, etc., we used Jmeter to test the system for 1000 concurrency, 5000 concurrencies, and 10000 concurrencies.

汇总报告										
名称: 1000并发汇总报告										
注释:										
所有数据写入一个文件										
文件名	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec	平均字节数
HTTP请求	1000	30	2	523	76.64	0.00%	991.1/sec	346.72	254.55	358.2
总计	1000	30	2	523	76.64	0.00%	991.1/sec	346.72	254.55	358.2

汇总报告										
名称: 5000并发汇总报告										
注释:										
所有数据写入一个文件										
文件名	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec	平均字节数
HTTP请求	5000	10156	113	23258	\$348.33	0.00%	203.5/sec	58.89	52.27	346.6
总计	5000	10156	113	23258	\$348.33	0.00%	203.5/sec	58.89	52.27	346.6

汇总报告										
名称: 10000并发汇总报告										
注释:										
所有数据写入一个文件										
文件名	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec	平均字节数
HTTP请求	10000	25386	18	59718	9168.28	38.41%	167.1/sec	202.80	25.94	1242.5
总计	10000	25386	18	59718	9168.28	38.41%	167.1/sec	202.80	25.94	1242.5

Figure 55: Concurrent Testing

Following is the diagram of HTTP response time.

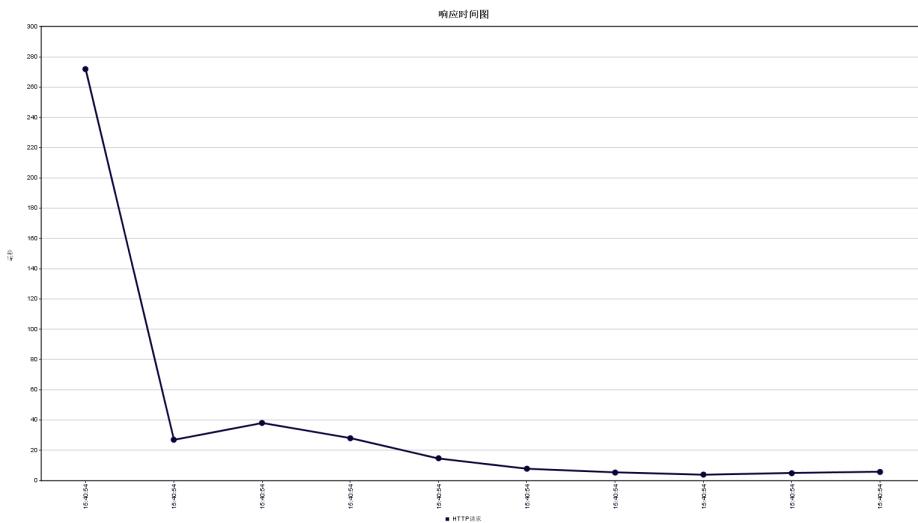


Figure 56: Time of HTTP Response

4.3.4 Back-end Performance Testing

Performance testing is the process of simulating normal and peak load access to the system under test through automated testing tools or code means to observe whether the performance indicators of the system are qualified.

We use MySQL and Redis benchmarks to conduct the performance testing.

```
Benchmark
    Running for engine innodb
    Average number of seconds to run all queries: 11.270 seconds
    Minimum number of seconds to run all queries: 11.270 seconds
    Maximum number of seconds to run all queries: 11.270 seconds
    Number of clients running queries: 1000
    Average number of queries per client: 10

PING_INLINE: 106044.54 requests per second, p50=4.671 msec
PING_MBULK: 104602.52 requests per second, p50=4.671 msec
SET: 108695.65 requests per second, p50=4.599 msec
GET: 109649.12 requests per second, p50=4.615 msec
INCR: 109890.11 requests per second, p50=4.687 msec
LPUSH: 108577.63 requests per second, p50=4.687 msec
RPUSH: 105152.48 requests per second, p50=4.663 msec
LPOP: 108932.46 requests per second, p50=4.575 msec
RPOP: 106723.59 requests per second, p50=4.711 msec
SADD: 107758.62 requests per second, p50=4.623 msec
HSET: 109890.11 requests per second, p50=4.607 msec
SPOP: 108932.46 requests per second, p50=4.607 msec
ZADD: 105596.62 requests per second, p50=4.671 msec
ZPOPMIN: 108108.11 requests per second, p50=4.687 msec
LPUSH (needed to benchmark LRANGE): 109170.30 requests per second, p50=4.719 msec
LRANGE_100 (first 100 elements): 63011.97 requests per second, p50=7.967 msec
LRANGE_300 (first 300 elements): 24248.30 requests per second, p50=21.327 msec
LRANGE_500 (first 500 elements): 16157.70 requests per second, p50=33.151 msec
LRANGE_600 (first 600 elements): 13715.54 requests per second, p50=39.135 msec
MSET (10 keys): 110011.00 requests per second, p50=5.167 msec
```

Figure 57: Performance Testing

4.3.5 Back-end Unit Testing

Unit testing belongs to the most stringent software testing means and is the verification method closest to the underlying implementation of the code, which can ensure the quality of local code at the lowest cost in the early stage of software development. Unit tests are executed in an automated manner, so they can bring high benefits in the scenario of a large number of regression tests. The implementation of unit tests can also help development engineers improve the design and implementation of code, and can provide examples of the use of functions in unit test code, because unit tests are specifically expressed in calling functions with a variety of different combinations of input parameters, and these calling methods constitute the instructions for using functions. We did a unit test by Junit for every function of the back end.

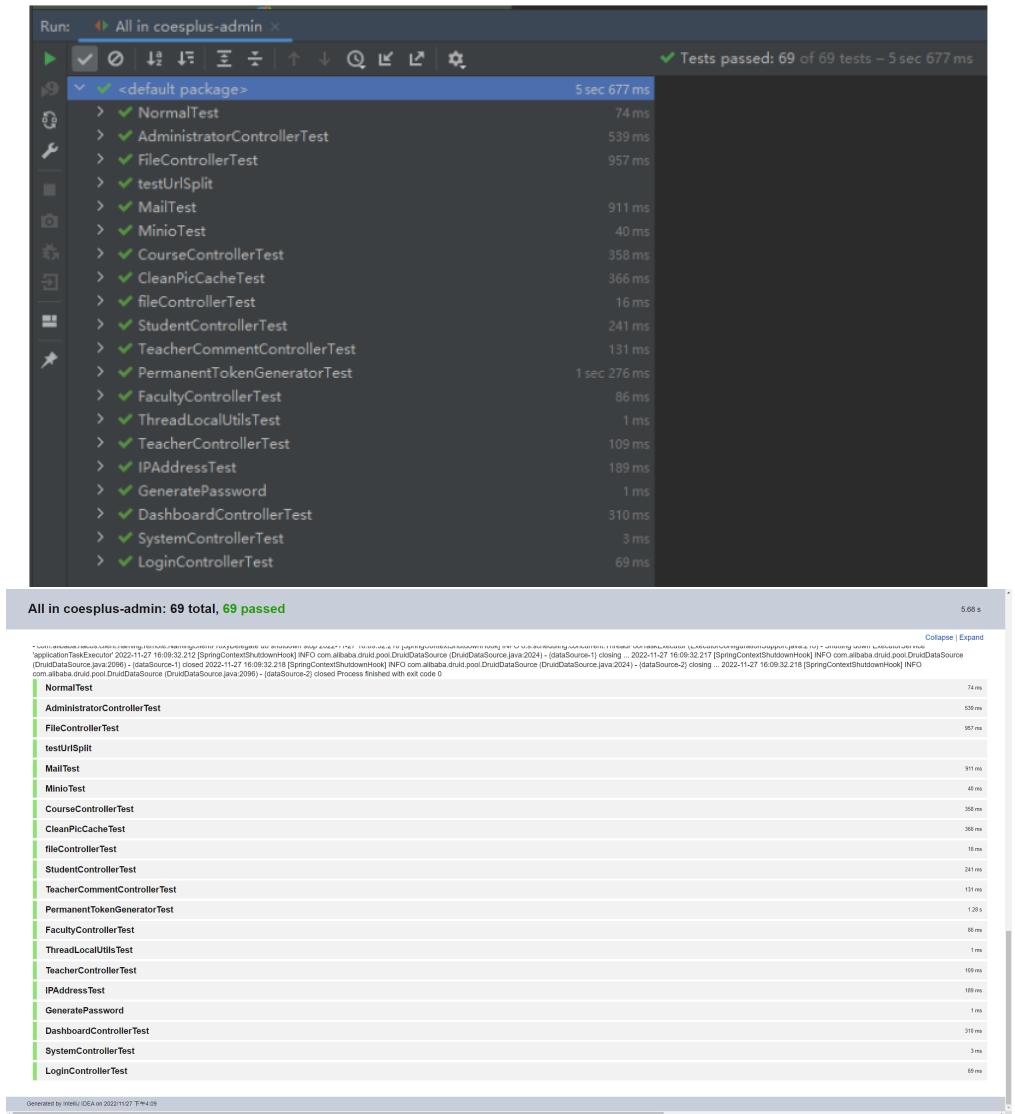


Figure 58: Unit Testing

4.3.6 Back-end Automatic testing of interface

An interface test is a test that tests the interface between components of a system. Interface testing is primarily used to detect the interaction points between external systems and between systems and internal subsystems. The focus of the test is to examine the exchange, transmission, and control management processes of data, as well as the inter logical dependencies between systems. We conducted the testing for the interface with the tool Apipost.



Figure 59: API Interface Testing

5 Management and Maintenance

Software management involves all aspects of software development, and its direct targets include people, money, and things. Simply put, people refer to software developers, money refers to project funds, and things refer to software projects. As a software manager, you should stand on a high ground and overlook the entire project. With the premise of the awareness of overlooking the overall situation and the use of appropriate management techniques, the project will be easier to carry out.

5.1 Software Management

Our project management decision proceeds from four aspects, which are Scope, Time, Cost and Quality. Scope, time, and cost together make the Quality triangle, popularly known as the project management triangle.



Figure 60: The Quality Triangle

- 1) Scope, that is, the scope of work refers to all the work that needs to be done in order to achieve the final goal. By scope we can indicate "what needs to be done to achieve the project goals", or "what work can be done and can be closed". This is very important if without a defined scope of work, projects may be difficult or never complete. Therefore, we must strictly control changes in the scope of work. Once there is a change, it is likely to affect the expected goals and final delivery results of the project.
- 2) Time, which is the time required to complete the task goal, can be used to help us quantify the amount of tasks and the expected completion time of the task. Time is also the most important indicator in our task plan. Time specifies the start and end time of a specific task.
- 3) Cost refers to all money consumed during the software development process, including a series of expenses such as personnel salaries and equipment expenses. Of course, these expenses are not unlimited. We need an overall budget. When the project is completed, the overall expenses should be within the budget within.

- 4) Quality refers to the degree to which the output meets expectations. Generally speaking, the degree to which we judge whether a project meets expectations is the quality of the project. We can also put forward requirements on the quality of the project according to our expectations, so that the completion of the project is more in line with expectations.

The goal of our team is to try to find a balance among time, cost, and scope, so as to achieve the most perfect quality expectation under a reasonable balance of the three. So as to achieve our expectation of adopting agile development "sooner and better"!

5.2 Management Model

5.2.1 Egoless Team Organization

Egoless programming is a state of mind in which programmer are supposed to separate themselves from their product. In this team organization goals are set and decisions are made by group consensus. Here group, 'leadership' rotates based on tasks to be performed and differing abilities of members.

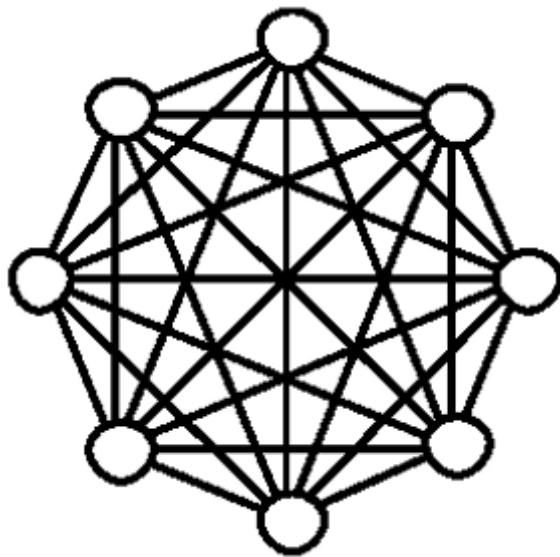


Figure 61: Egoless Programming

5.2.2 CMMI

The full name of CMMI is Capability Maturity Model Integration. CMMI is the latest version of the CMM model. Early CMMI (CMMI-SE/SW/IPPD), SEI began to promote and try in some countries and regions. With the promotion of applications and the development of the model itself, deduction has become a comprehensive model that is widely used. The full name of CMMI is Capability Maturity Model Integration. There are 5 levels of CMMI certification, CMMI level 1, completion level; CMMI level 2, management level; CMMI level 3, definition level; CMMI level 4, quantitative management level; CMMI level 5, optimization level. We define our team as CMMI level 5

for the following reasons. In project management, we have fully utilized JIRA for digital management and progress monitoring to achieve management accuracy, so we have already met CMMI level 4. Secondly, we have also applied many new technologies. Continuously improve and optimize the process to make a series of processes such as software design, testing, and development faster and better.

5.2.3 PERT Chart

And we also using PERT chart and Gantt Chart for scheduling. This is our RERT chart: Each node in the figure lists the start and end times when we did the task. For clarity, the time required is also listed. The units of time in the figure are weeks, days and hours.

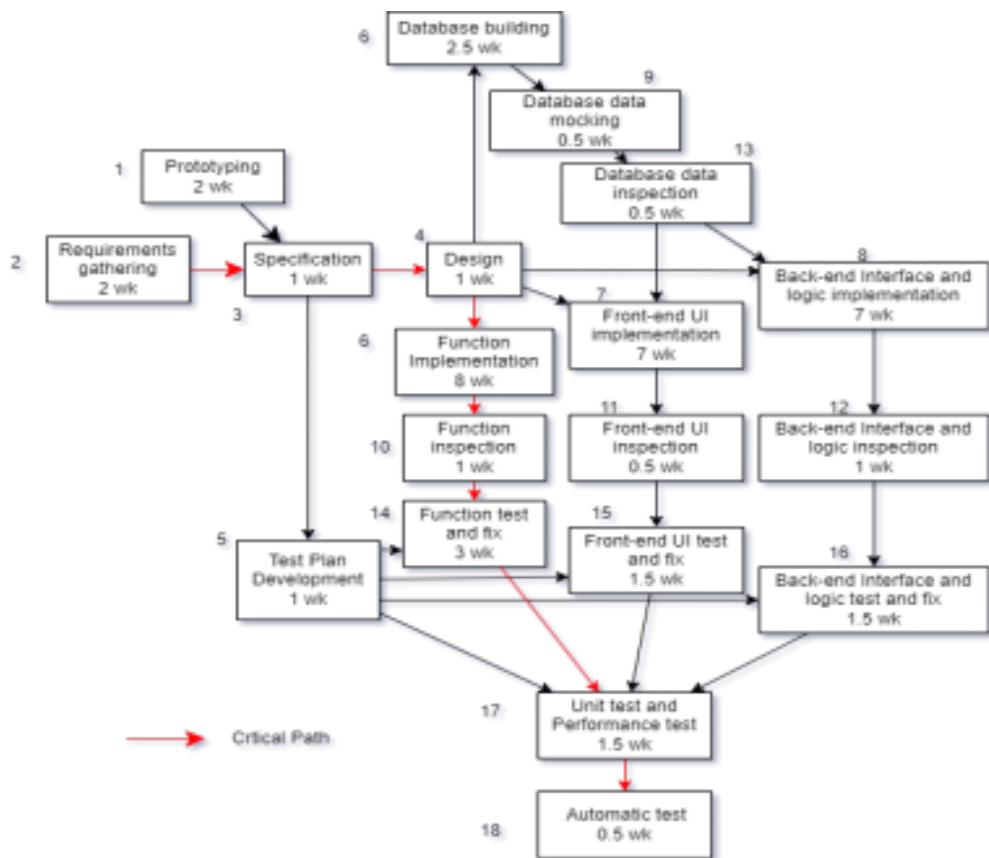


Figure 62: Pert Chart

5.2.4 Gantt Chart

The project lasted 10 weeks in total, divided into 8 sprints. We took 2 weeks for the initial requirement collection, then we took one week for each sprint. All of these sprints played a very important role in our development process. Agile testing started from the fourth week throughout, and the final maintenance, review, and presentation preparation took one week each.

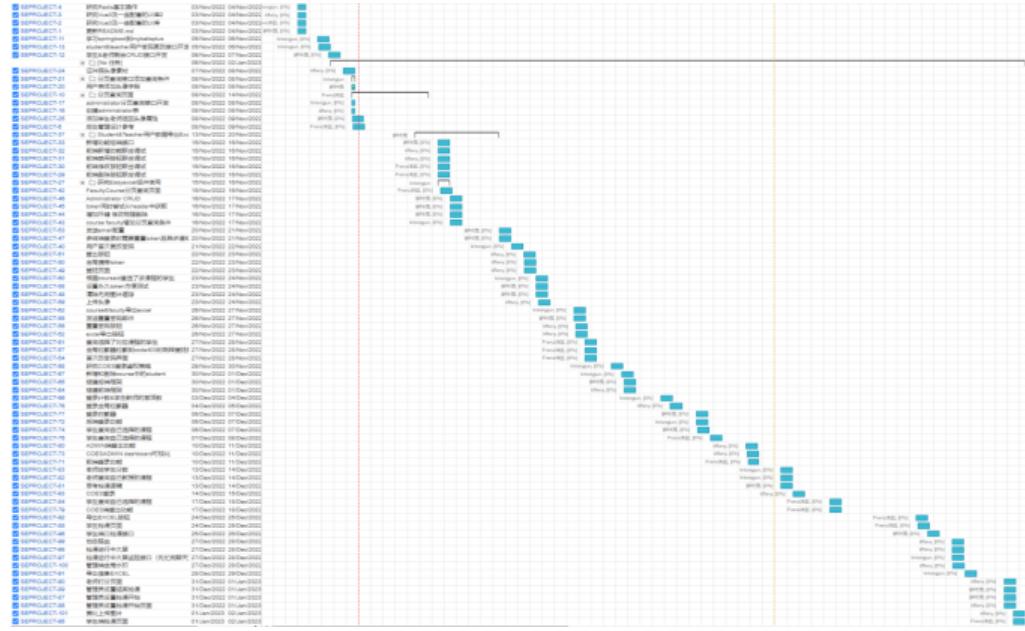


Figure 63: Gantt Chart

5.2.5 Earned Value Chart

Earned value analysis is a method to calculate the actual workload, estimated total cost and completion time of a project. It involves three key values of each work: the planned value (PV) is the approved cost estimate to be spent on the work within the specified time; Actual Cost (AC) The actual cost of completing the work within the specified time (the sum of direct and indirect costs); Earned Value (EV) The value of work actually done.

The earned value is compared with the completed work through the plan at the beginning of the project, and gives an estimate of when the project will be completed. The project manager can estimate how much resources will be spent when the project is completed by extrapolating from the completed part of the project.

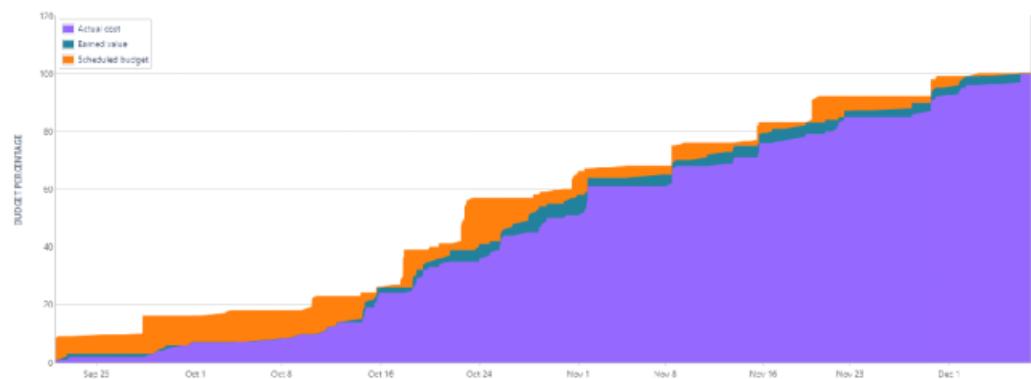


Figure 64: Earned Value Chart

5.3 Software Maintenance

There are roughly four types of software maintenance activities: corrective maintenance, adaptive maintenance, perfection maintenance and preventive maintenance.

Corrective maintenance is to correct some potential program errors or design defects exposed under specific use conditions.

Adaptive maintenance is to modify software to adapt to changes in data environment or processing environment during software use.

Perfection maintenance is to modify the software to incorporate these requirements into the software after users and data processors propose requirements for improving existing functions, adding new functions, and improving the overall performance when using the software.

Preventive maintenance is to improve the maintainability and reliability of software.

Advanced software engineering methods are used in advance to design, prepare and test the software to be maintained or a part (re) of the software, so as to lay a good foundation for further software improvement.

5.3.1 Corrective Maintenance

We realized that although we have adopted an active testing strategy, it is inevitable that errors may occur in the future. Therefore, when new problems are reported, we will also actively respond, and timely arrange developers to locate and repair problems, so as to make the system more perfect.

5.3.2 Adaptive Maintenance

We have done the following work. First, the main language of the project is written in JAVA, CSS, HTML, and JAVASCRIPT, which means that our front end can run on all browsers in the world (IE browsers cannot be guaranteed). At the same time, JAVA language has cross platform characteristics because it is re run in JVM. Our back-end programs can also be run on servers with JDK1.8 installed in all types of systems. This means that the environmental impact on our software is very weak. It has good adaptive ability. Secondly, we have adopted the NACOS registry mechanism to make hot updates to the changed configurations, which also improves the adaptability of our software.

5.3.3 Perfection Maintenance

We considered the problem of code decoupling at the early stage of system design. We split complex logic into small logic. The low coupling of code also means that when we receive new requirements, we can develop new requirements more quickly without affecting the original logic composition, which greatly reduces the cost of our completeness maintenance.

5.3.4 Preventive Maintenance

We also considered in the early stage of software architecture design. First, we analyzed that the biggest weakness of the current system is the performance problem. Therefore, we felt that the micro-service architecture was adopted in the early stage of design to facilitate the system's performance expansion, so that we can better cope with the sudden increase in access. At the same time, we will continue to optimize the logic of the interface, So as to avoid problems that may be encountered in the later stage.