# Second-hand Book Trading Mall Based on Micro-service Architecture

# Project Report

Project Title: Second-hand book trading mall based on micro-service architecture

Team Name: Team 21

Name:

Team Numbers:

| **Name** | **Email Address** |
| --- | --- |
| KEXUAN ZHANG (19103255) | [KEXUAN.ZHANG.2020@MUMAIL.IE](mailto:KEXUAN.ZHANG.2020@MUMAIL.IE) |
| HONGZE CHEN (19103191) | [HONGZE.CHEN.2020@MUMAIL.IE](mailto:HONGZE.CHEN.2020@MUMAIL.IE) |
| YICHEN HU (19105053) | [YICHEN.HU.2020@MUMAIL.IE](mailto:YICHEN.HU.2020@MUMAIL.IE) |
| YIHUI SHI (19103981) | [YIHUI.SHI.2020@MUMAIL.IE](mailto:YIHUI.SHI.2020@MUMAIL.IE) |
| ZHENGYANGWANG (19104634) | [ZHENGYANG.WANG.2020@MUMAIL.IE](mailto:ZHENGYANG.WANG.2020@MUMAIL.IE) |
| XU ZHANG (19105215) | [XU.ZHANG.2020@MUMAIL.IE](mailto:XU.ZHANG.2020@MUMAIL.IE) |

**TA: Xiao Zude, He Jia**

## **Chapter 0**

## Introduction to Agile Development (SCRUM)

Agile is one of the most widely used and most recognized software development frameworks in the world.Agile was born on a clear day when 17 people with different development methodology backgrounds, together exploring possible alternative software development solutions that can jointly brainstorm, seeking that may shorten development time and reduce demand for documentation.At the time, software development used to happen so long that when the project was ready to delivery, the business had moved forward and demand had changed.Therefore, even if the project can meet its established goals, it cannot meet its business needs.

So the elites of these different software engineering technologies came together, and the end result of their meeting was what they call the "Agile Declaration."But the agility that was born that day is not what we see in the organization today.The method that experts agree to is called "lightweight" and fast.However, the main outcome of this meeting was the view that faster product delivery and continuous feedback were key to successful software development.

Existing waterfall technology is too cumbersome to provide no feedback until the final product is ready for delivery.This means there is no room for requested corrections and no view of the progress until the entire product is ready.That's what these experts want to avoid.They want a solution that gives constant feedback to avoid the cost of late rework.

Agile is not just a practice of applying settings when developing software.It also brings changes in the team mindset, which prompted them to build better software, work together and ultimately make them a satisfied customer.Agile values and principles enable teams to divert their attention and change the thinking process where they build better software

The existing waterfall technology at the time was too cumbersome to provide no feedback until the final product was ready for delivery.It was called the developed waterfall model because the team first completed one step before moving to the next step.This means there is no room for requested corrections and no view of the progress until the entire product is ready.That's what these experts want to avoid.They want a solution that allows constant feedback to avoid the cost of rework at later stages.That is why agility is also about adaptive and continuous improvement, but also about continuous feedback and delivery speed.

Agile is not a set of rules.Agile is not a set of guidelines.Agile isn't even a methodology.Instead, agility is a set of principles that encourage flexibility, adaptability, communication and work software beyond planning and processes.It is captured very succinctly in the so-called Agile Declaration.

Agile software development enables teams to work more effectively when developing complex projects.It consists of the practice of practicing iteration and incremental techniques that are easily adopted and show good results.

In applying Agile to action, we have a variety of agile-based methods to meet all the needs of the software development industry, from software design and architecture, development and testing to project management and delivery.Not only that, agile methods and methods also open a scope for process improvement as a component of each delivery.

Agile is a practical philosophy of software development, which constructs a self-sufficient and cross-functional team dedicated to continuous delivery through iterations and develops throughout the process by collecting end-user feedback.

Various and diverse industries have various agile methodologies.However, the most popular method of all these methods is: Scrum, Kanban and XP.

“A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.” --Schwaber, K. and Sutherland, J., 2011. The scrum guide.Scrum Alliance,21, p.19.

Scrum means a professional term for rugby for "fight"; naming a development process Scrum, I think you can imagine your development team developing a project, passionate as rugby, and you will be excited to finish it. Scrum is a development process where you see your team work efficiently.

Essentially, Scrum is an agile framework for developing, delivering, and sustaining complex products. Initially focused on software development, it has since been applied to other areas, including research, sales, marketing and other advanced technology areas. A Scrum team suggests a design for a ten-member team that delivers work in an iterative and incremental manner, with each iteration called a Sprint. A Sprint takes no more than a month, usually two weeks.

The Scrum team focuses on a single common goal at each Sprint, and the developers on the Scrum team review progress towards that common goal and adjust the current plan in the Daily. At the end of the Sprint, the team conducts a Sprint review with stakeholders to review current results and adaptation plans, providing an opportunity for information exchange. Finally, the team conducts Sprint reviews to continue improvement.

Scrum is a process framework that includes a range of practices and predefined roles (a process, plan, mode for efficiently developing software).At each sprint (a 15 to 30-day cycle, length determined by the development team), the development team creates an increment of available (whenever available) software.The feature to be achieved in each sprint comes from the product order. The product order (product target) is the summary requirement (target) of the work needed to be done by priority.Which order item (target item) will be added to a sprint, as determined by the sprint planning meeting.During the meeting, the product leader told the development team what orders he needed to complete in the product order.The development team decides how many orders they will be able to commit to completing in the next sprint.During the sprint, no one can change the sprint order, meaning that the demand is frozen in a sprint.

During the sprint, project status meetings are held every day known as the "scrum" or the "daily stand-up meeting".Daily stand-up meetings have several specific guidelines:The meeting begins on time.Late teams often impose penalties (such as fines, push-ups, and rubber chicken toys around their necks) to welcome everyone, but only "pigs" can speak.Sessions were limited to 15 minutes, regardless of team size.All attendees should stand up.(Help to keep meetings short) Sessions shall be held at a fixed place and at the same time of each day.At the meeting, each team member needs to answer three questions:

What work have you done?What do you plan to do in the future?Are there any obstacles to accomplishing your goals?(The Scrum Supervisor needs to note these barriers) After each sprint is completed, a sprint review meeting is held, at which all team members will reflect on the sprint.The sprint review meeting was held to allow for ongoing process improvement.The meeting will be limited to 4 hours. Scrum advocates all team members to sit together, have oral communication, and emphasize the project-related specifications (disciplines) that help to create self-organized teams. A key principle of Scrum is to acknowledge that customers can change their minds and change their needs in the course of the project, and that predictive and planned approaches do not easily address this unforeseen change in needs.

The Product Order (product backlog) is a summary of the entire project.Product orders include a rough description of all the required features.Product orders are about what products are to be created.Product orders are open, and everyone can edit them.Product orders include rough estimates, usually in days.The estimate will help the product leader measure the schedule and priorities.Each Product Backlog item should include the attributes of a description, order (priority), estimate, and value. Each item should include an acceptance criterion for assessing whether it is “Done”. Items in the Product Backlog may be grouped by their attribute as themesor epics. Higher ordered items often have clearer and more detailed descriptions as they are better understood.

Sprint Order (sprint backlog) is a greatly detailed documentation with information on how the team needs for the next sprint.The task is broken down into hours, and no task can exceed 16 hours.If a task exceeds 16 hours, then it should be further decomposed.The tasks on the sprint orders are not assigned, but are signed by team members to claim their favorite tasks. A sprint Backlog should include at least one high priority process improvement identified in the retrospective meeting of the previous Sprint. The plan in the Sprint Backlog should have sufficient details to guide the Daily Scrum. Items in a Sprint Backlog can be changed, but can only be changed by the Development Team

The burn-out diagram (burn down chart) is a publicly displayed chart showing the number of unfinished tasks in the current sprint, or the number of unfinished order items on the sprint order.Do not confuse the burn out map with the earn value map.The burn-out diagram makes the sprint (sprint) to smooth covering most of the iteration cycle and keeps the project still in the planned cycle.

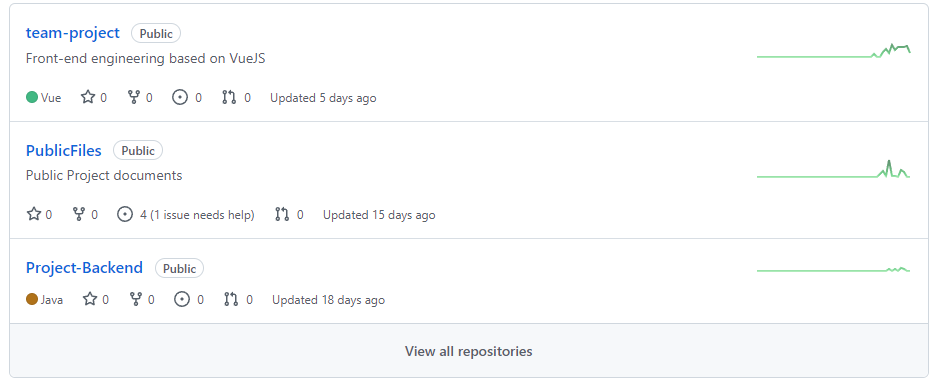
Scrum also has three distinctive characteristics which contains Transparency, Inspection and Adaptation. Because it prescribes process must be visible to all team members and use common language and share a common definition. Scrum artifacts and progress must be inspected frequently without interfering team’s work lead to the Inspection.There are the reasons of the Adaptation. On one hand, if a process or development deviates outside the plan, adjustment must be made in time. On the other hand, use an iterative and incremental approach to control risk and Make decisions based on what is known.

## **Chapter 1//**

## **Chapter 2**

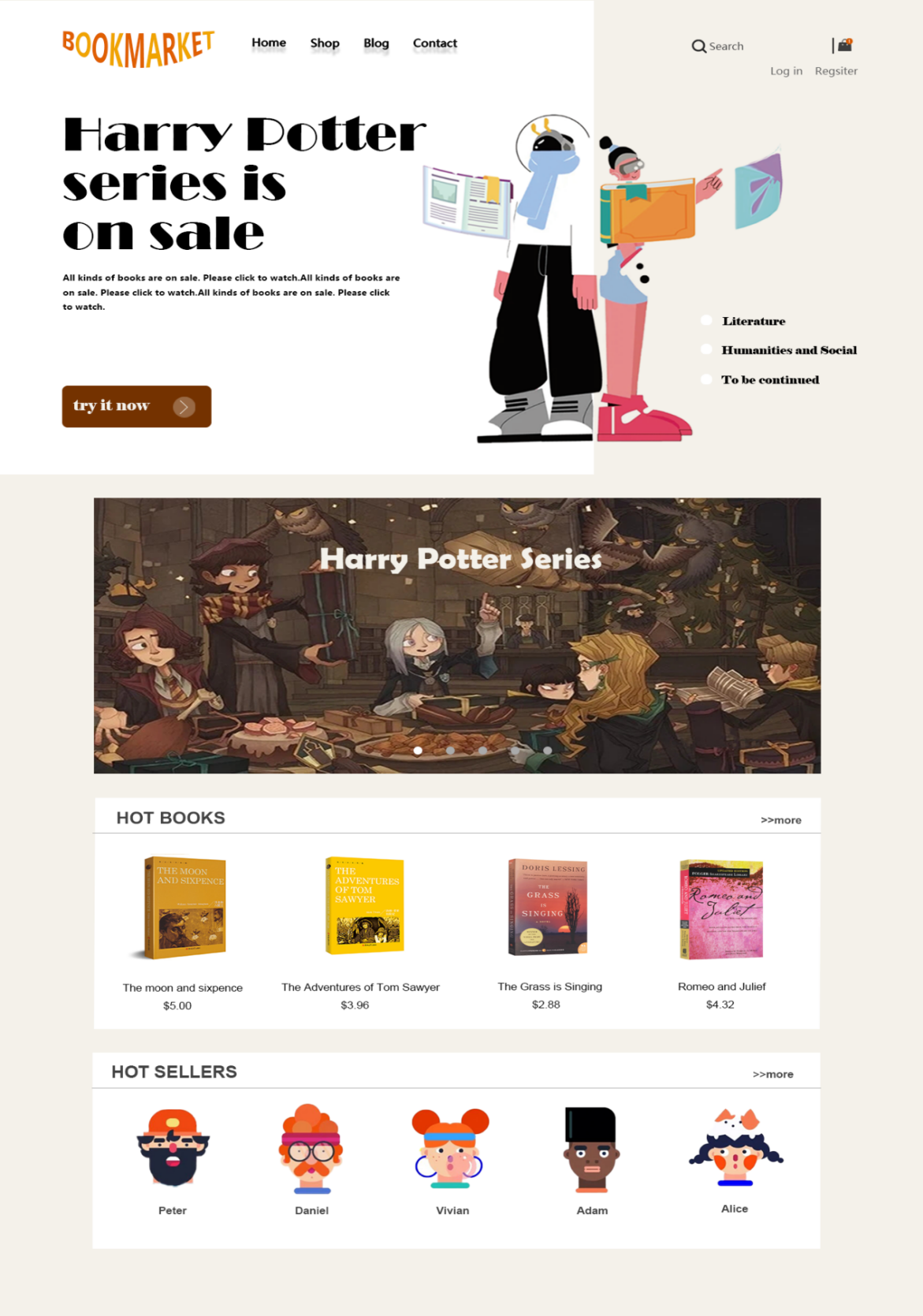
## **Actual Project Introduction**

Online second-hand book mall is a Web application and desktop application based on the architecture of front and back end separation, and can be responsive to different sizes of user devices for adaptation. It is mainly used to serve the needs of second-hand book trading circulation in different scenes of campus and society. At the same time, the website provides exchanges and reviews of books, non-used bookstores to publish books and other functions. The website covers a complete user management, orders, funds, transactions of the whole process of services, while providing a convenient and easy-to-use management background panel for site managers.



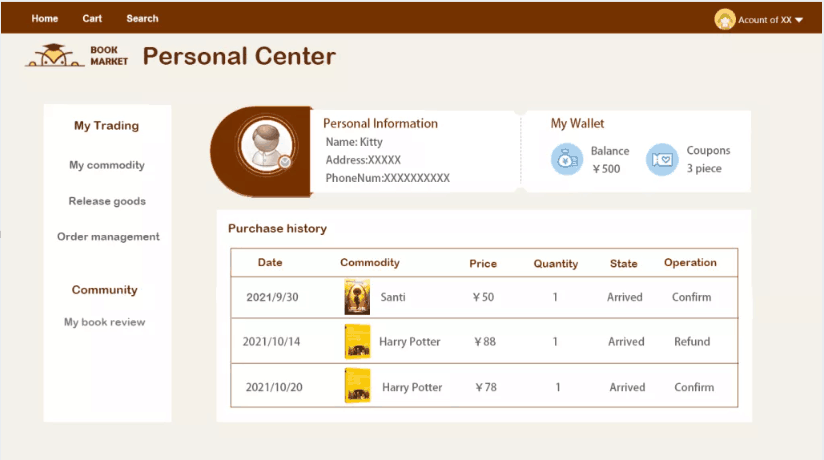
#### **Home Page**

On the top left side of the main page are three obvious buttons, Home, Cart, and Search. Clicking the Home button on the main screen does not jump, but cart and Search will jump to the corresponding screen. On the top right is the personal information section. If the user has not logged in, Login/Register will be displayed. Click to Login or Register. If the user has already logged in, the user's profile picture and user name will be displayed, and the click will take the user to the profile page. The recommended numbers scroll down the middle of the home page, followed by Hot Books and Hot Seller.



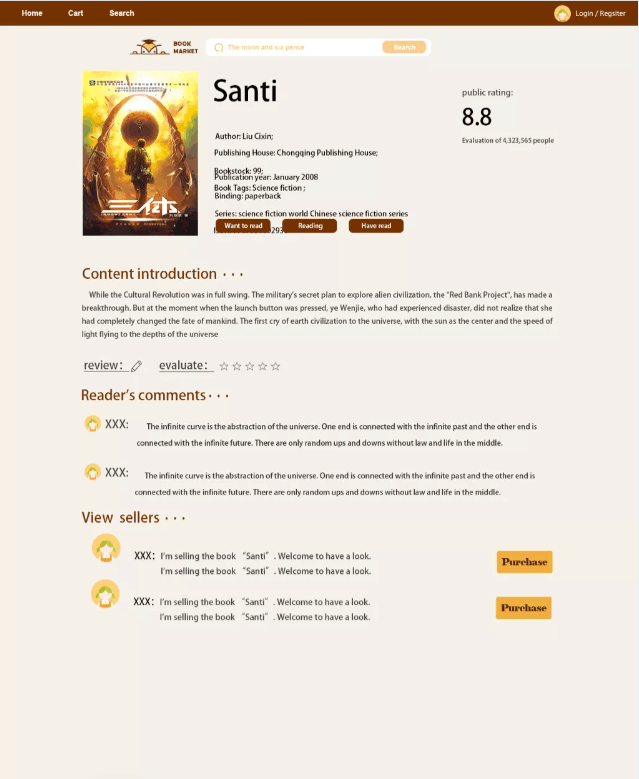
#### **User Center**

In the personal center, it is obvious that we can not only see our name, address, contact information and other personal information, but also see our balance and coupons. On the left side of the page, there are different buttons that can be clicked to enter different pages.



#### **Book Page**

On the product details page, we can clearly see the picture of the book and some specific information, such as author, publisher, book label, and reader rating. The rest of the page displays summaries and reader comments, respectively. At the bottom are sales, where sellers display their books and buyers select.



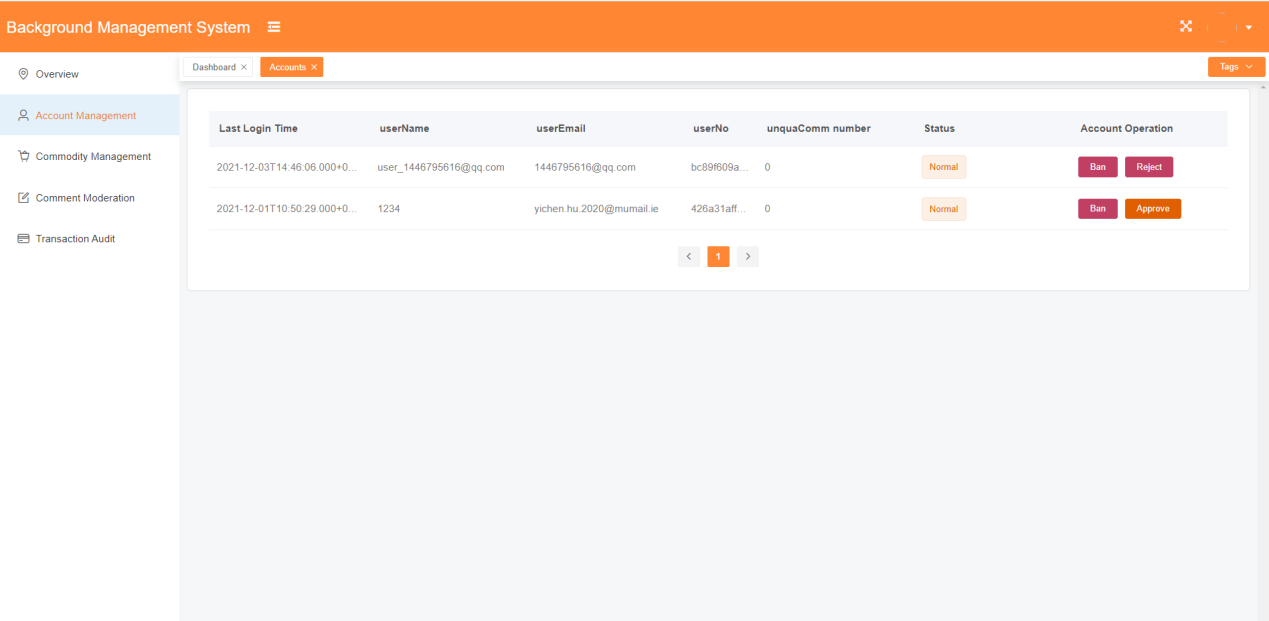
#### **Search Page**

On the product details page, we can clearly see the picture of the book and some specific information, such as author, publisher, book label, and reader rating. The rest of the page displays summaries and reader comments, respectively. At the bottom are sales, where sellers display their books and buyers select.

#### **FJ~I51D[(Q1FJSX8AFQYS7M**

#### **Background Management System**

The background management system is separated from the front-end page. Only authenticated administrator accounts can be accessed through the login page. The list on the left consists of overview, Account Management, Merchandise management, comment management and Transaction Audit. After the administrator clicks the desired function, it will jump to the corresponding page. In the background management system, the administrator can block and unblock user accounts. Bad comments can also be deleted.



### **Technologies**

The front end is deployed through Serverless and is bound with domain names and gateways. Windows applications are also published. Back-end deployment with Linxu server and Nginx reverse proxy.

#### **UI/UX designs**

Our UI and UX design team members use Adobe Photoshop and Sketch for prototype and hi-fi design to provide front-end implementation references and automatically generated element style code. At the same time as an important basis for product and back-end design business. Our team collaborated on UI-graph-based design through Blue Lake (online collaboration platform software).

Project Back-end Details

Spring Boot：it is a new framework in the spring family, which is used to simplify the creation and development of

spring applications. Spring boot is a new framework provided by the pivotal team, which is designed to simplify the new

process. The initial construction and development process of spring application. The service end of the project adopts a

system with more perfect community ecology Java language for development. Due to the light weight, convenience and

rapid development of small program end, in order to maintain the synchronous promotion of both ends, The server not

only inherits the original excellent features of the spring framework, but also has a lighter weight and simplified

configuration Spring boot framework for rapid development, so that the project can be launched as soon as possible, and

also for the scale of later projects. Lay the foundation for building a micro service architecture.

Mybatis Plus: Mybatis plus (opens new window) (MP for short) is an enhancement tool of mybatis (opens new window).

Based on mybatis, it only makes enhancement without change, and is born to simplify development and improve

efficiency.

characteristic

No invasion: it is only enhanced without change, and its introduction will not affect the existing project, which is as

smooth as silk

Low loss: the basic curd will be injected automatically upon startup, with basically no loss of performance and direct

object-oriented operation

Powerful crud operation: built in general mapper and general service, most CRUD operations of a single table can be

realized only through a small number of configurations, and there is a powerful condition constructor to meet various

use requirements

Support lambda formal call: it is convenient to write various query conditions through lambda expression, and there is

no need to worry about wrong fields

Support automatic generation of primary key: support up to 4 primary key strategies (including distributed unique ID

generator - sequence), which can be configured freely to perfectly solve the primary key problem

Support activerecord mode: support activerecord formal calls. Entity classes only need to inherit model classes to

perform powerful CRUD operations

Support custom global general operations: support global general method injection (write once, use anywhere)

Built in code generator: code or Maven plug-in can be used to quickly generate mapper, model, service and controller

layer code, support template engine, and more custom configurations for you to use

Built in paging plug-in: Based on mybatis physical paging, developers do not need to care about specific operations.

After configuring the plug-in, writing paging is equivalent to ordinary list query

The paging plug-in supports multiple databases: MySQL, MariaDB, Oracle, DB2, H2, hsql, SQLite, postgre, sqlserver

and other databases

Built in performance analysis plug-in: it can output SQL statements and their execution time. It is recommended to

enable this function during development and testing to quickly find out slow queries

Built in global interception plug-in: it provides intelligent analysis and blocking of full table delete and update

operations, and can also customize interception rules to prevent misoperation

Active MQ: it is a kind of message middleware, which is the medium through which applications transmit messages in

distributed systems. Apache ActiveMQ is an open source message middleware developed by the Apache Software

Foundation. Using ActiveMQ to build message queues can not only further reduce the coupling between services through

the publish subscribe working mode, but also reduce the system response time and peak traffic through asynchronous

processing, so as to improve the system performance and high concurrency processing ability。

Maven: is a project management tool that can build and manage Java projects. It is an automatic build tool.

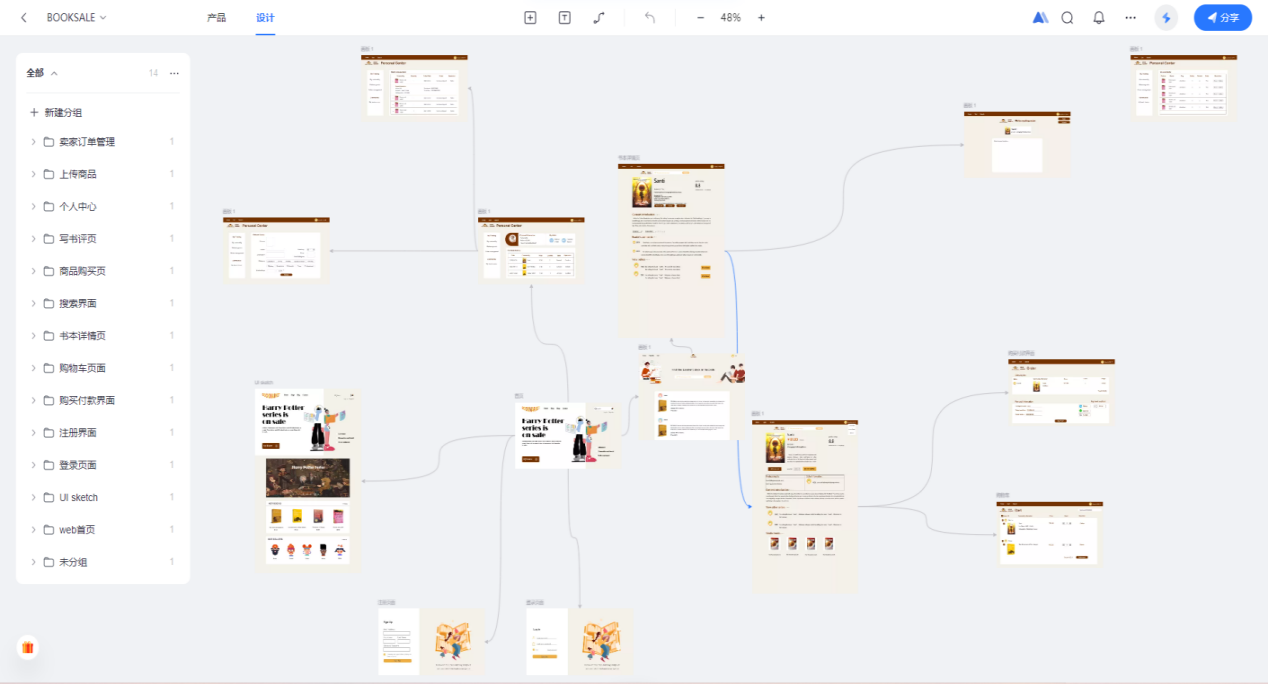
MySQL: it is a safe, cross platform and efficient database system, which is closely combined with PHP, Java and other

mainstream programming languages.Object storage service (OSS) : is a mass, secure, low-cost and highly reliable cloud storage service, which is suitable for

storing any type of files.

Tortoisegit: Is an open source git version control system

XShell: Powerful security terminal simulation software



#### **Front end**

Vue.js: Use the mainstream MVVM framework vue. js to build SPA applications. Vuex manages front-end status and VUE-Router controls routes

ElementUI: UI component library from Ele. me

Animate. CSS: Provides CSS animations

Store: local browser cache management tool(Based on LocalStorage)

Axios: Powerful AJAX tool for pre-control of data responses and requests

Webpack: Project build packaging tool.

Eslint: Project code style specification

Less: precompiled CSS extension language

Electron.js: Used to build and publish desktop applications based on Web code

### **Back end**

Spring Boot：it is a new framework in the spring family, which is used to simplify the creation and development of spring applications.

Mybatis Plus: It is an enhancement tool of mybatis. On the basis of mybatis, it is only enhanced without change. It is born to simplify development and improve efficiency.

Active MQ: it is a kind of message middleware, which is the medium through which applications transmit messages in distributed systems.

Maven: is a project management tool that can build and manage Java projects. It is an automatic build tool.

MySQL: it is a safe, cross platform and efficient database system, which is closely combined with PHP, Java and other mainstream programming languages.

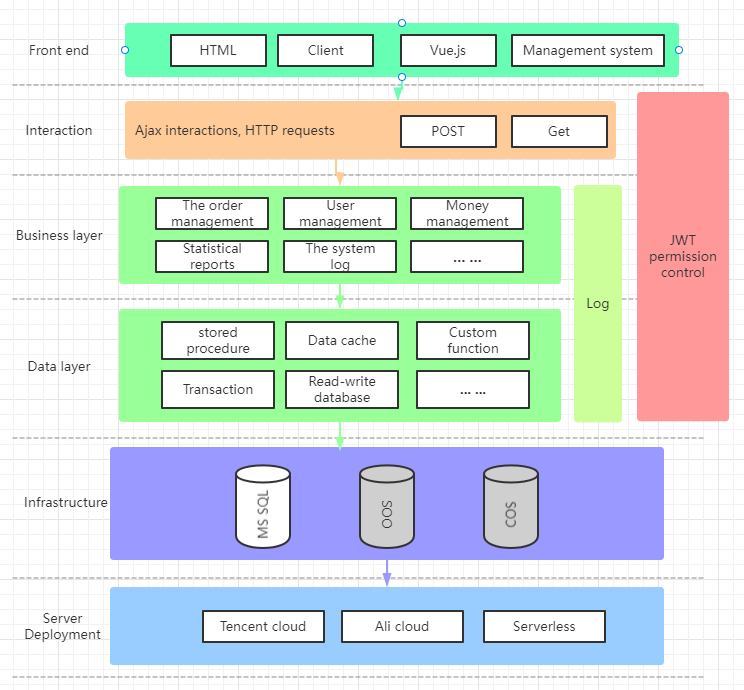
Object storage service (OSS) : is a mass, secure, low-cost and highly reliable cloud storage service, which is suitable for storing any type of files.

Tortoisegit: Is an open source git version control system

XShell: Powerful security terminal simulation software

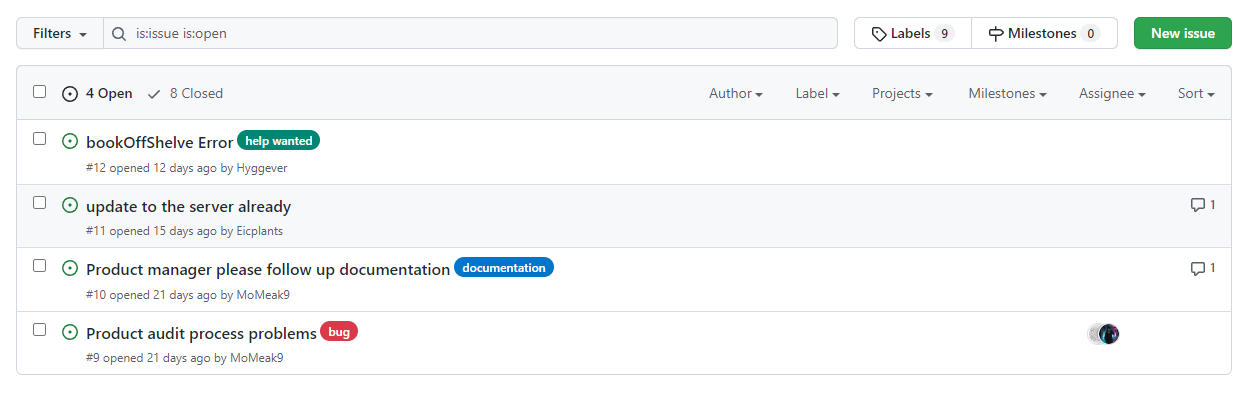
### **Architecture**

The project is based on the front-end and back-end separation architecture. JSON data is exchanged between the client and server through HTTP, and data is exchanged and displayed through AJAX. See the architecture diagram for the specific architecture:



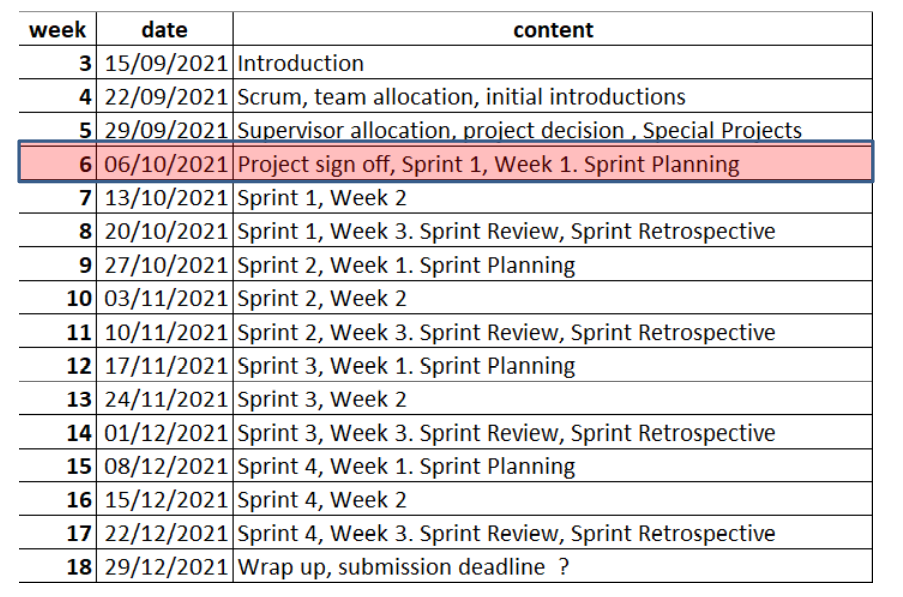
### **Testing**

The backend API uses Postman for interface testing, and issues are published and tracked through Github. Front-end In addition to the self-test of the pre-release version by the front-end engineer, the students in charge of the test will directly test the published and deployed applications and write test reports.



**Chapter 3**

## The SCRUM Process



Sprint Preparation

**Work allocation:**

Team leader : ShiYihui

Product manager:WangZhengyang

UI/UX : ZhangKexuan,HuYichen

Front-end : ShiYihui,HuYichen

Back-end : ChenHongzhe,ShiYihui

Testing : ZhangXu

Report editor : WangZhengyang,ZhangKexuan

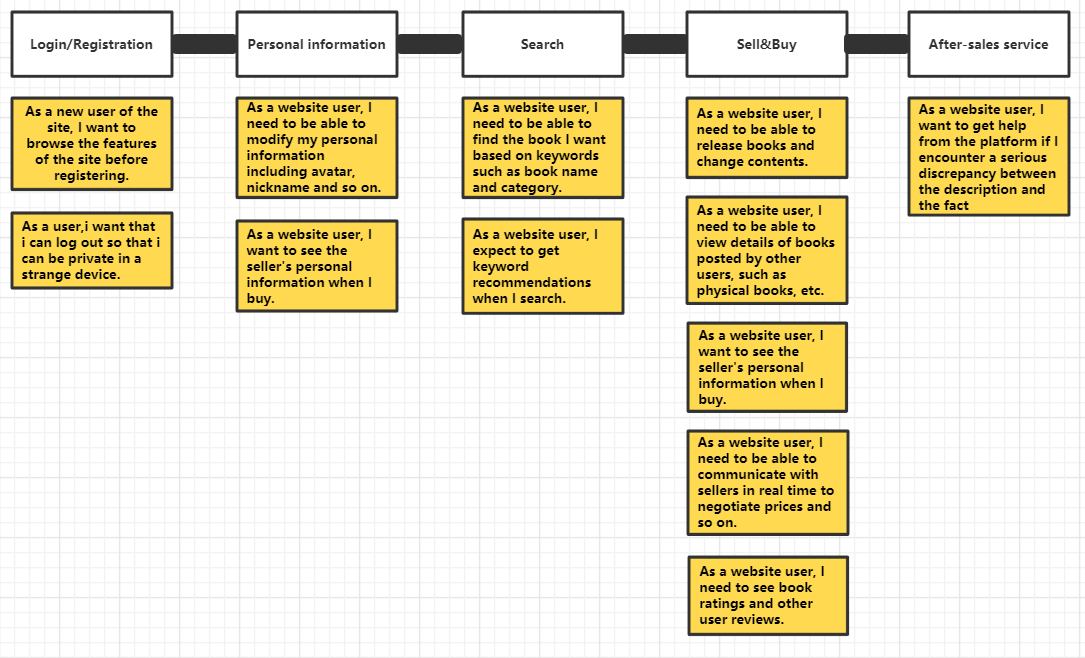
|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Preparation |  |  |  |
| Week3 | Introduction of the course | Get to know the course content |  |
| Week4 | Learn scrum | Initial Introduction | Team allocation |
| Week5 | TA allocation | Project decision | No special project |

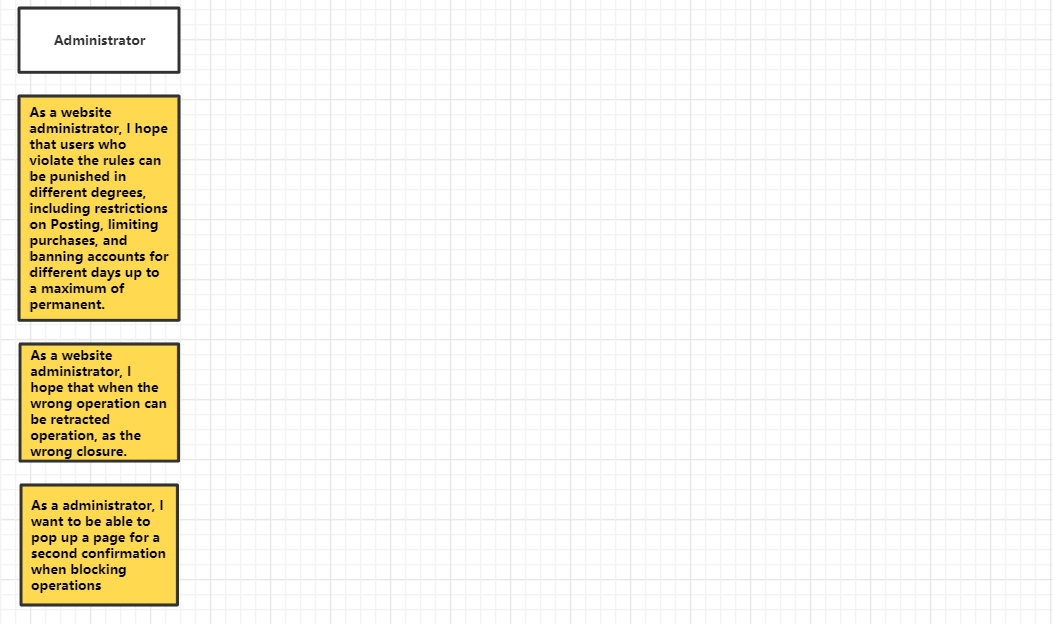
During the first sprint, we focused on project preparation. We learned about Agile and Scrum and thought we could do a good job on this project. After the initial planning of the project, we created our Github repository, which links can be accessed.

<https://github.com/Team21st/PublicFiles>

In week 8, we decided on the main color of the page and decided to start our project with the design of the homepage. Together, we wrote 20 user stories including the identities of buyers, sellers and administrators. The user story map will be displayed as follows

**User story map**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint 1 |  |  |  | Start to draw burndown charts |
| Week 06 | Sprint planing |  |  |  |
| Week 07 | Create the github repository |  |  |  |
| Week 08 | UI/UX | Design homepage | User Story | Discuss the possible function |

In week 9, we tried to implement login and registration functions, but we had some difficulties at first. As a newcomer to industrial development, we were at a loss to start, so we searched a lot of videos from the Internet and got some suggestions from the teaching assistant, so as to complete this function on time. In addition, we also realized the static page of the home page in front and designed the search page.

In week 10, we completed the search function and started designing the shopping cart page and the book details page. This week's task was not difficult and we finished it quickly.Of course, in week 11, we successfully implemented the shopping cart page and the personal page.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint 2 |  |  |  |
| Week 09 | Login/Register | Implement Homepage | Design search page |
| Week 10 | Implement search function | Design cart page | Design book details page |
| Week 11 | Implement cart page | Implement Personal page |  |

Sprint3 was a tough three weeks. In the twelfth week, we designed the page of the administrator system, and discussed the functions that the administrator system should have, including reviewing products, reviewing comments, blocking users and so on. In Dashboard we selectively display line charts and bar charts that represent site activity. In week 13, we implemented the shopping cart page and discussed the item details page. This week we also prepared for the test section. In week 14, we implemented the product details page discussed last week and continued to work on the admin system. But we decided to change the way administrators log in. At first we wanted to use different urls to distinguish between regular users and administrators, but later we decided to identify users by identifying their ID.

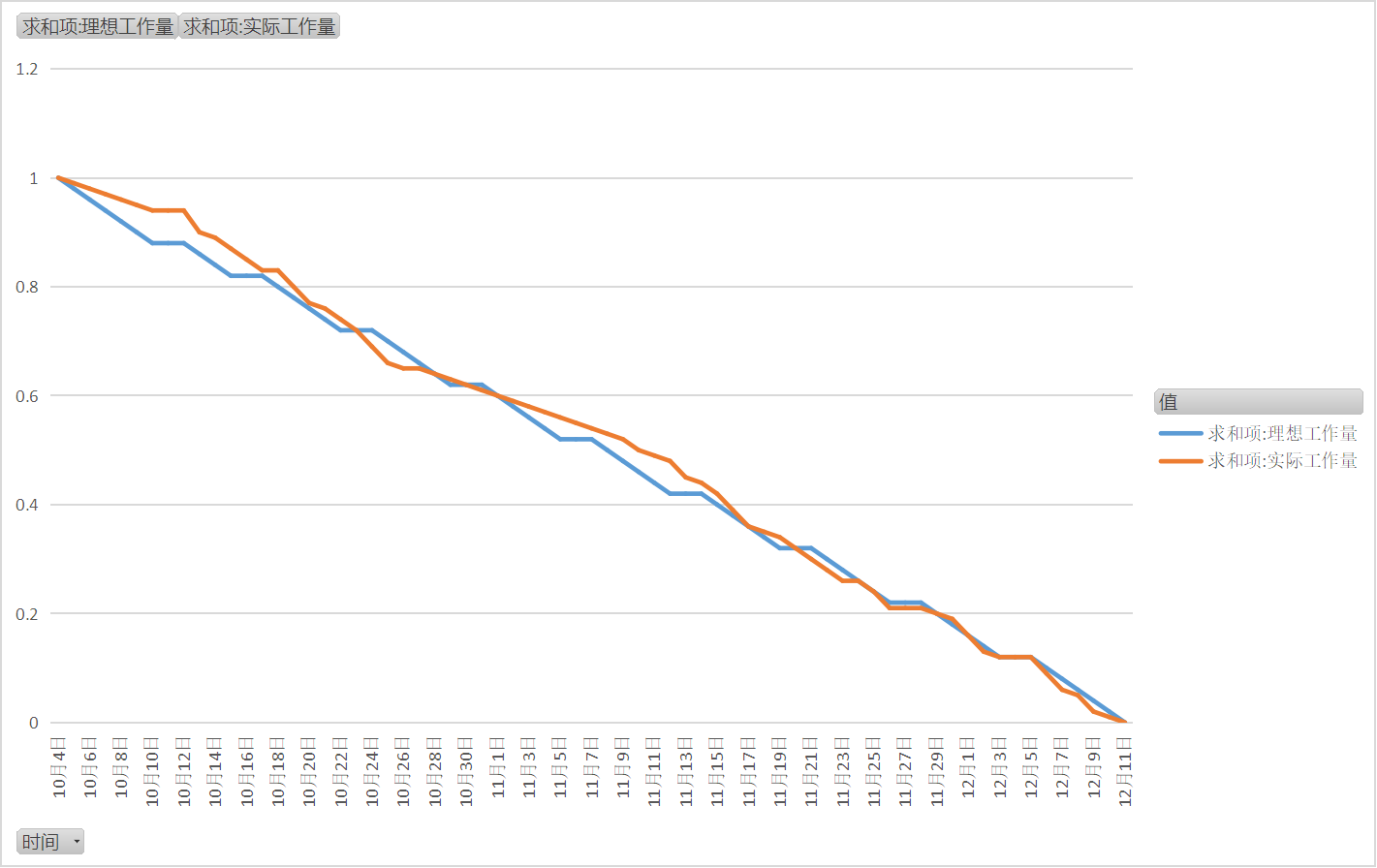
|  |  |  |  |
| --- | --- | --- | --- |
| Sprint 3 |  |  |  |
| Week 12 | Design administrator system | Discuss the function | Discuss the data need to be show |
| Week 13 | Implement the cart page | Discuss details page | Discuss the function need to be test |
| Week 14 | Implement details page | Continue on administrator system | Change the way of landing |

Test Driven Development, Version control, Documentation

This was the last sprint, in which we implemented the last system, the administrator system, and began centralized testing. We found some minor bugs and some connectivity issues, but after the update we fixed all the issues. After all this was done, we started shooting the presentation video and editing the final report.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint 4 |  |  |  |
| Week 15 | Implement administrator system | Start testing |  |
| Week 16 | Test some function | Solve some problems |  |
| Week 17 | Finish the main body of project | Shoot video | Edit report |

Burndown chart



**Some user Story**

As a website user, I need to be able to find the book I want based on keywords such as book name and category.

As a website user, I need to be able to modify my personal information including avatar, nickname and so on.

As a website user, I need to be able to publish books and publish content changes.

As a website user, I need to be able to view details of books posted by other users, such as physical books, etc.

As a website user, I need to see book ratings and other user reviews.

As a website user, I need to see all of a seller's sales.

As a website user, I need to be able to communicate with sellers in real time to negotiate prices and so on.

As a website user, I like to be able to label my items when I sell them, including the type of book and how old they are.

As a website user, I want to see the seller's personal information when I buy.

As a website user, I would like to report suspected violations.

As a website user, I expect to get keyword recommendations when I search.

As a new user of the site, I want to browse the features of the site before registering.

As a second-hand book seller, I hope to enter the website to sell.

As a second-hand bookseller, I hope to be certified and recommended by the website.

As a webmaster, I want to be able to review the content posted by users.

As a website administrator, I hope that users who violate the rules can be punished in different degrees, including restrictions on Posting, limiting purchases, and banning accounts for different days up to a maximum of permanent.

As a website administrator, I hope that when the wrong operation can be retracted operation, as the wrong closure.

As a webmaster, I want to be able to pop up a page for a second confirmation when blocking operations.

As a website operator, I hope that when users release products, website manager review can cross two audit, that is, a manager review can be released, and occasionally the published content is pushed to other managers, if the audit results are different, then push to a higher level of administrator to determine.

As a website operator, I hope to make personalized recommendation to users' home pages through some algorithms.

**Testing**

Over the course of time, we've created several interfaces. From the beginning of user related interface, such as user registration, login, and so on, now gradually derived from the transaction related interface, administrator related interface, the whole system is more perfect, gradually developed mature.

At the beginning of the project, we encountered many difficulties. At the beginning, the connection between the server and the port was unstable, which caused us a lot of difficulties during that time (for example, the verification email could not be sent to the user's mailbox). After continuous coordination with the back end, this problem was improved after a period of time, and gradually completed the user related interface, allowing faster development of subsequent ports in other aspects.

With the increasing number of transact-related interfaces, it is sometimes easy to generate logical conflicts, which requires constant communication and coordination with the backend. After time and time of modification, we finally completed a relatively complete purchase process, from ordering to delivery to confirm receipt of goods, as well as shopping cart function and order inquiry and so on. Every interface added increases the probability of an error, and we need constant testing to ensure that each new interface added does not cause an error in the original interface.

//Code versioning (Git etc...)

**Team communication**

The product manager has a close relationship with every part of the team. Make requests to all parts of the project from the user's point of view. After understanding the requirements of the product manager, the UI completes the design sketch based on the project content. Make improvements after discussion with other team members. Discussions include whether the front end can be implemented perfectly, how it can be implemented better, and whether the back end can work with the front end to make the functionality complete.

UI according to the requirements to make a design, and marked on all the data, spacing, picture materials, etc., to the front end. UI interface is the main task of the prototype design (skeleton), interactive logic comb (from which interface to another interface), interactive experience design, how to reduce the user, unnecessary repetition operation), visual interface design (interface beautification, color, icon design, etc.), design code and components finishing (reusable components, And reduce communication costs with front-end development).

The front end gets the design document and starts to build a static page according to the design drawing. The main tasks of the front end are interface restoration (use code to turn the interface in the drawing into an operable software interface), component development (development and arrangement of common components such as buttons), and docking with the back end (use the interface of the back end for docking), etc.

The back end begins to do preparatory work when the front end is set up. The main tasks include data processing (database storage, query, sorting, optimization), data structure, algorithm optimization, etc.

Our test section did not differentiate between front and back ends. The first test of both the front end and the back end is done. At the end of the project, we also conducted intensive tests on the whole project and each function. The result of the test is quite satisfactory. There were a few hiccups, but we worked them out pretty quickly.

**Impact of initial remote working**

Because of the epidemic, the initial remote work did have some impact on our project schedule. Not being able to meet face-to-face for discussions was a hindrance to the exchange of information between our team members. We set up our communication group (Tecent QQ Group 792051411) to ensure a smooth flow of information and to exchange some of our files. The Github repository is used to save our documents (https://github.com/Team21st/PublicFiles)

Normally, we must have a lot of meetings at the beginning of the project, including content discussion and technical stack preparation. Make a good arrangement of work and prepare all kinds of things for starting work. But the new crown epidemic disrupted the normal rhythm and people could not move and gather, so we had to transfer all the work to be done online. But the online work was not fully focused for some of the personnel, which made our initial progress of the project less than perfect.

The biggest disadvantage of working remotely is the efficiency of communication and self-management. Communication in remote work is not as efficient as face-to-face communication. Communication through the Internet and screens is more or less lost and unfamiliar to us. The richness and concentration of face-to-face information cannot be replaced by text, voice and video. As for the self-management aspect, it is even more important. If we can't focus completely on our work, we can't be as productive as we could be. Of course, while we were working on CS353, we had other projects to complete, which made time allocation more challenging.

**Chapter 4//**

****Your contribution**** – including background research and code that may not have featured in the final application. [up to 1500 words]

**Chapter5**

Summary of front-end project experience

 It is difficult to coordinate and unify the front and back end in the development project of front and back end separation, which requires the full communication of the front and back end engineers.  In this project, we use Ajax to communicate with the server and client through HTTP.  The biggest difficulty is to coordinate the content and usage of the interface.  Our front-end and back-end engineers both wrote pages and APIS in accordance with UI design drawings at the beginning, and then back-end engineers showed and explained API usage methods for front-end engineers by providing Swagger interface documents.  But we also encountered many difficulties.  First of all, engineers at the front and back end have different understandings of UI and user interaction methods, resulting in different data contents to be presented.  Secondly, this also involves the modification of the database tables, and we need to update the database fields on the old basis after negotiation.  Finally, of course, we need to spend more time adjusting and communicating. For us, effective communication is the only way to advance the project, although the front and back end engineers are very capable.  Of course, fortunately, our front and back end engineers were very patient and communicated well, often meeting the current Spring workload ahead of schedule.

All in all, our front-end work is also quite busy, and we need to communicate with the UI and back end at the same time, but this does not prevent our work plan and content to proceed smoothly and complete. Of course, this also includes communication between front-end engineers and testers, because many front-end page bugs can also be caused by the back-end interface problems, but will be directly reflected in the front-end page. This requires front-end, test, and back-end engineers to debug, find and troubleshoot problems.

**Back-end**

Obviously, there are a lot of problems on the back end as well. Everything from technology selection to technology implementation is not easy. We've summarized some of the issues and solutions here. We hope we can face these problems with ease next time

**Problem**:The connection between the server and the database is unstable.It is not easy

modify the user information. Sometimes the modification fails, and sometimes the

 information is not updated after modification. At the same time, the profile picture

information and files in the user information are also difficult to upload to the server.

**solution：**

our back-end solve this problem. just simply change a server which is much better then the old one. for the upload of the picture, back-end user the OSS service from Ali Cloud, it increase the speed of saving the picture and save the storage space of the server.

**Problem**: Sometimes the sellers can not find the information of the goods they have uploaded. The updated information is not displayed on the queryMyCommodity interface after modifying the item already on the shelf. The connection between the server and the database is not very stable.

**Whose problem**: The back-end

**solution：**

This is a simple code error. we have the wrong query condition

**Problem**: In our logical scenario, we query for the item to get the bookNo, which can then be used to add the item to our shopping cart. But addShoppingCart interface is faulty. Cannot add a book to the shopping cart by bookNo.

**solution：**

This is a simple logic error. This is a good lesson

**Problem:** There is a problem with the format agreement between the front end and the back end. Because the front end and the back end did not communicate in time before writing the code, the transmission format did not meet the expectations of the front end

**solution：**

The back end rewrites the return format for simple calls from the front end

**Problem:** There are many omissions in the returned data of the back end, the team did not communicate effectively in time

solution： Conduct timely and effective communication, and each team member speaks out his understanding of a scene

**Problem:** The project exception caused by the plug-in accidentally returns an error

**solution**：The back-end finds the problem plug-in and deletes it in time

**Problem:** Ambiguity caused by requirement change

**solution**：It is forbidden for team members to quarrel over requirements, and the product manager shall decide the scenario method in time

**Testing**

Since this was my first experience with interface testing, I had to start from scratch, The testing software used in our project is Postman, which is used for testing with Swagger.

Nowadays, many developers use the model of front and back end separation, and our project is no exception. Advantages of the separation of the front and back ends: developed separately, they are relatively independent and loosely coupled. The front and back ends interact with each other through APIS. The back end provides an interface to the front end, and the front end calls the interface. The front end just calls the interface and renders, and the only connection between the front end and the back end becomes the API interface. As a result, API documentation is becoming increasingly important. Swagger is a framework for better API documentation, and Swagger can simulate HTTP request calls.

Postman is an interface debugging and testing tool that supports THE HTTP protocol. Its main features are powerful, easy to use, and easy to use. Postman is one of our preferred tools for both developers debugging interfaces and testers testing them.

Therefore, we use Swagger and Postman to work together to obtain interface correlation from Swagger, then use Postman to test the interface. Since this is the first time to use, from the beginning of the installation in the reference tutorial. This is followed by learning how to use basic functions, such as interface requests. There are four types of interfaces in Postman: query parameter interface, form interface, JSON interface, and upload file interface. The following describes the four types of interfaces and how to request in Postman. There are also form-type interface requests. In addition, there are interface response data analysis, interface management, global variables, environment variables and other functions. In the whole process of the test, I learned a lot of new knowledge and enriched my ability.

**References**

Appendix 1

No.01\_20211003\_Syh

Get to know team members and allocate work initially

Team leader : ShiYihui

Product manager:WangZhengyang

UI/UX : ZhangKexuan,HuYichen

Front-end : ShiYihui,HuYichen

Back-end : ChenHongzhe,ShiYihui

Testing : ZhangXu

Report editor : WangZhengyang,ZhangKexuan

No.02\_20211006\_Syh

Propose project ideas and discuss them in preparation for Sprint 1

The project "Second-hand book Mall" was preliminarily approved, and the alternative project was "Library Management System".

No.03\_20211013\_Chz

"Second-hand Book City Project" is confirmed, github repository is established, and the permission to edit the warehouse is obtained

No.04\_20211020\_Hyc

UI and login and registration functions, designing personal pages, discussing various features of the project, writing 20 user stories

Discuss the distinction between administrator and user accounts and consider the possibility of using different addresses

No.05\_20211027\_Zkx

The registration and login function is preliminarily completed, the home page design is completed, and the search page is designed  
Front-end and UI discussions, fine-tuning the script

No.06\_20211027\_Zx

Realize search function, design shopping cart page and product details  
Discuss the display contents and functions of the product details page

No.07\_20211103\_Wzy

Implement shopping cart page, personal page  
Finalize the details of personal page presentation

No.08\_20211110\_Syh

Discusses the design of background management system, the corresponding functions and the need to show the data.

No.09\_20211117\_Chz

Complete the shopping cart page, discuss the product details page again, and discuss the test module.

No.10\_20211201\_Hyc

Finished product details page, background management system is still in implementation, decided to unified login interface, by differentiating user ID to distinguish between administrators and general users.

No.11\_20211208\_Zkx

Basic implementation of background management system, began to test the function

No.12\_20211215\_Zx

Test the function and fix some small connection problems

No.13\_20211222\_Wzy

Test function complete, project main body complete, start video shooting and document editing section.

****Appendix 2 - Additional individual information including personal reflection on the process and contributions to the project  [at least 1000 words]****

****Appendix 3 Peer Reviews****