

Software Requirements Specification

System Analysis and Design
Spring 2021



@Tongji

A Comprehensive Campus
Community Platform
for Students

Team Members:
1853572 Qiao Liang
1854062 Zhibo Xu
1854117 Zhenyu Li
1953803 Yixi Zheng
1954106 Wenchao Chen



同濟大學
TONGJI UNIVERSITY

Table

1. Introduction.....	3
1.1 Project Background.....	3
1.2 Project Purpose.....	3
1.3 Project Ascendancy.....	3
1.4 Core Users.....	4
1.5 System implementation.....	4
2. Agile requirements gathering.....	5
2.1 Empathy Map Canvas.....	5
2.2 User Journey Map.....	6
2.2.1 Second-hand trading system.....	6
2.2.2 Public Course Evaluation System:.....	7
2.2.3 Community Information System.....	8
2.3 User Story.....	9
3. Use case modelling.....	10
3.1 Global View on Use Cases.....	10
3.2 Subsystems.....	11
3.2.1 Personal Information.....	11
3.2.2 Authority management and information audit system.....	18
3.2.3 Second-hand trading homepage sysytem.....	21
3.2.4 Payment Transaction System.....	26
3.2.5 Chatting and After-sale Service System.....	29
3.2.6 Course Evaluation System.....	30
3.2.7 Association Management System.....	35
4 Glossary of terms.....	39
5 Supplementary specification.....	40
5.1 Performance.....	40
5.2 Reliability.....	40
5.3 Security.....	40
5.4 Availability.....	40
5.5 Maintainability.....	41
5.6 Understandability.....	41
6. Initial snapshots of the system's user interface.....	42
7. List of references.....	46
8. Contribution of team members.....	48

1. Introduction

1.1 Project Background

With the advent of the information age, the demand for college students to **obtain campus information and facilitate their study and life through the Internet** is increasing day by day. At present, students mainly use WeChat, QQ and other social software to obtain major learning and other notifications. However, due to the lack of pertinence of social software, the above information is very complicated, and the efficiency of students only obtaining useful information through such software has become very low. To this end, this project integrates a series of campus information that is **highly demanded** by students, such as **community information and management, and public elective course evaluation platforms**.

In addition, we learned through the needs analysis of the student groups that students urgently need a platform for **second-hand goods transactions on campus**. Although the current second-hand trading software in the software market is very rich, there has never been a software that only uses students as a user group. Therefore, the campus community software also provides a second-hand trading platform only for the student groups on the campus of Tongji University. These three platforms are **combined** to form the campus community software-@tongji.

1.2 Project Purpose

This project aims to provide **a campus information exchange platform** with the three core functions of on-campus second-hand trading, community information management, and evaluation of public elective courses for the majority of students of Tongji University. Among them, campus second-hand trading is the core function of this software platform.

1.3 Project Ascendancy

Compared with similar software Tongxin Cloud, this software platform has the following advantages:

- Centralized demand:** Tongxin Cloud software has many functions, but only a few of them really poke the "pain points" of middle school students. @tongji avoids functional redundancy and gathers three main needs of students, making the software concise and easy to use.
- Portable and easy to use:** The platform centrally presents the three functions of second-hand trading, community information management, and public elective course evaluation. The software is lightweight and easy for users to use.

·Complete information: This software concentrates on displaying the information on the community and the evaluation of public elective courses, and the information is complete.

1.4 Core Users

The core users are:

- Student users who wish to purchase and sell second-hand items within the campus and have high requirements for the safety of second-hand transactions.
- Student users who need to quickly and centrally obtain information released by clubs and student organizations.
- Student users who need to know student evaluation information about public elective courses.

Ordinary users are:

All students and community organizations.

1.5 System implementation

In order to be portable and visually display information, realize user use case functions, and facilitate subsequent system analysis, we divide the system into the following 7 subsystems:

User Information System: Users can uniquely and effectively register an account through the system to realize "One person one ID", and can maintain all aspects of their personal information in real time, including browsing records, sold items, selected elective courses, and so on.

Authority management system: The authority administrator has his own personal account and can manage all kinds of authority of various users in real time, including the authority to publish items, the authority to publish public elective course evaluation, and the authority to publish community information.

Second-hand transaction homepage system: All student users can search, browse and filter second-hand items on this homepage. The homepage can also push the preferences of each user according to the internal

Data Generation system of the system, and users can also add their favorite items to their favorites.

Item trading system: Buyer users and seller users can implement commodity transactions safely and quickly, and provide relatively complete after-sales service.

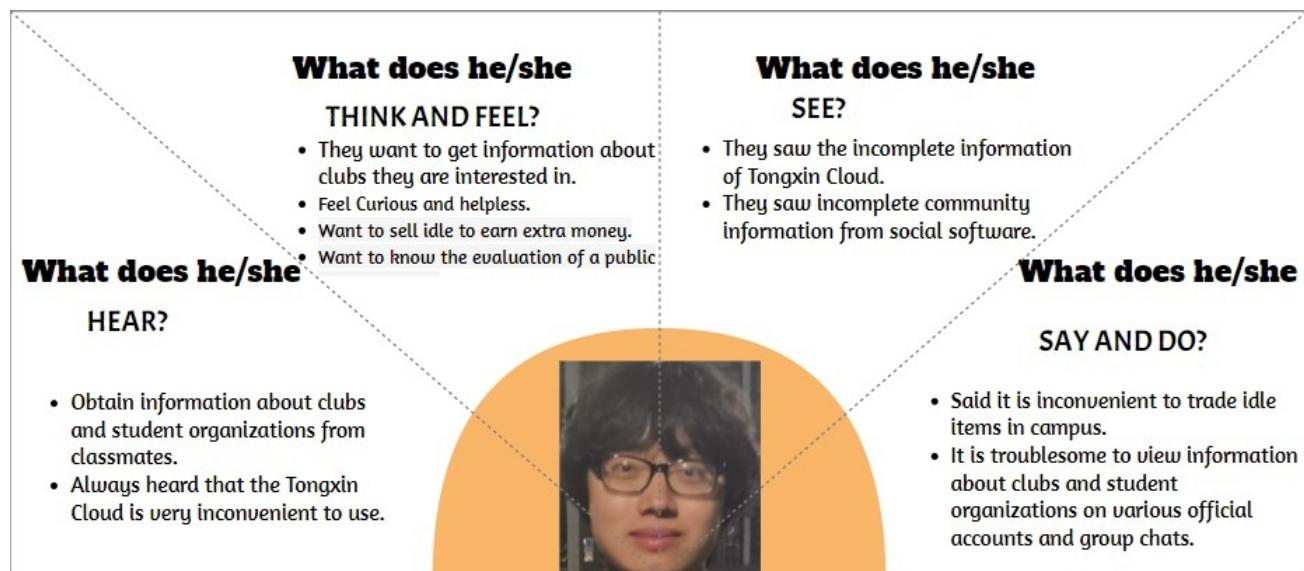
Real-time chat system: All users can chat in real time after entering the second-hand trading platform, which is convenient for discussing the time and location of item transaction, and can also chat with the customer service administrator in real time.

Course Evaluation System: As one of the secondary functions of the platform, the system provides a platform for students to view the evaluation of public elective courses and publish the evaluation of public elective courses. In order to provide a harmonious evaluation environment, the evaluation system does not support the posting of text comments, and the function posts scoring comments on the courses that they have taken.

Community Information System: As one of the secondary functions of the platform, the system provides a platform for community users to edit and publish various information related to the community, and also allows student users to view information about clubs and student organizations that they are interested in.

2. Agile requirements gathering

2.1 Empathy Map Canvas



PAIN	GAIN
<ul style="list-style-type: none"> Unable to get different kinds of information about clubs and student organizations that interest him in time . Can't find a reliable platform that is convenient for selling idle items. Unable to find a way to know the pros and cons of public elective courses in advance . 	<p>Through our software platform, he/she will gain:</p> <ul style="list-style-type: none"> An official platform to facilitate the sale of idle and purchased items in school. Know the information of the clubs he/or is interested in and the student organizations he/she want to join in time Rapidly get others' evaluations of public elective courses that interest you

Through this analysis, we know the core needs of the target users:

First of all, in terms of second-hand transactions, students hope to have a unified platform that only trades idle items within the university, and the safety of students' transactions can be guaranteed to the greatest extent.

On the other hand, as for college students who have just entered school, they want to know the recruitment information of various clubs and student organizations through a centralized platform, rather than not knowing where to obtain the information; for the old students who have already enrolled, they also

want to know the recruitment information at any time. Quickly understand the new information released by various organizations.

In addition, students need a platform that can collect all students' evaluations of various public elective courses, so that they can know the basic information of the course in advance before deciding on the course.

Thus we divided the software platform into three modules: **campus second-hand transaction system**, **community information system** and **public course evaluation system**. Since all students in the school have the needs of second-hand transactions on campus, the target users of the social information system and the public elective system are only some students, and maybe some students only want to obtain such information

through communication with classmates or group chats. Therefore, for the **priority of the three systems of the software**, we will focus on analyzing the **campus second-hand transaction system**.

2.2 User Journey Map

2.2.1 Second-hand trading system



When using @Tongji, a user must first register an account, and then log in and enter the online system.

Here, we need to design two use cases of registration and login, and users can modify their personal information after logging in. Therefore, we designed the use case of editing personal information. The user also needs a search interface when searching for items in the second-hand trading system, and the system needs to be able to give feedback when the search fails, so the Filter Products use case is required. After

discovering the products they like, users can contact the sellers through the instant messaging system. Users can inquire about the specific information of the products and communicate with the sellers about the delivery of the goods. After users choose to buy goods, they need to pay through the trading system, and after the payment, they can also know the status of the transaction through the trading status interface. After finally receiving the goods, users can rate the sellers according to the actual situation of the goods and their satisfaction.

2.2.2 Public Course Evaluation System:



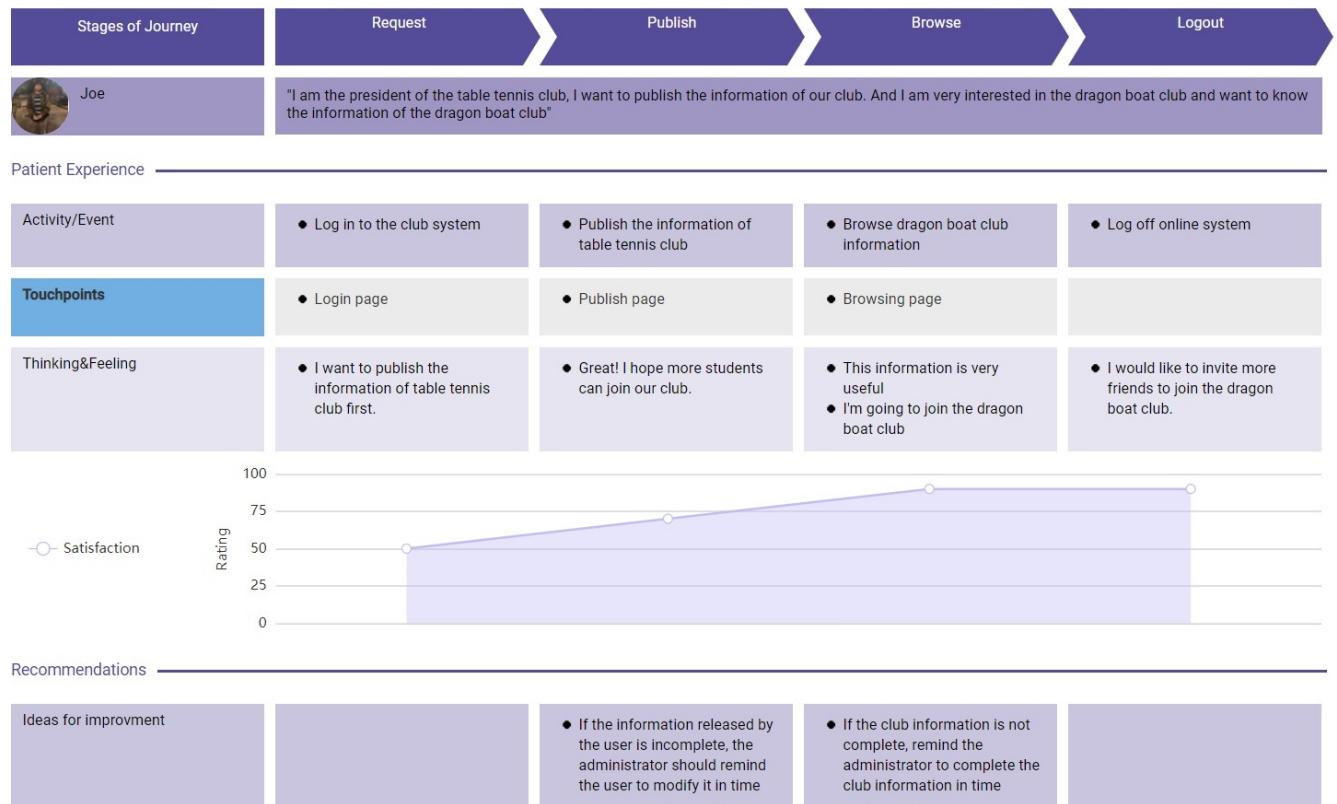
Users need to know the course information through @Tongji.

First of all, users can search for the course name in the search interface, which requires designing a use case to obtain the course information.

Then, users can search for the course evaluation after browsing the course information to know the students' evaluation of the course. When users finish learning a course, they can also submit their own evaluation of the course in the system, which requires an updated course evaluation use case.

For non-standard comments, the administrator needs to prompt the user to modify or withdraw them, which is related to the Review and Comment use case. When the course information is incomplete, the user can also remind the administrator to update the course information in time.

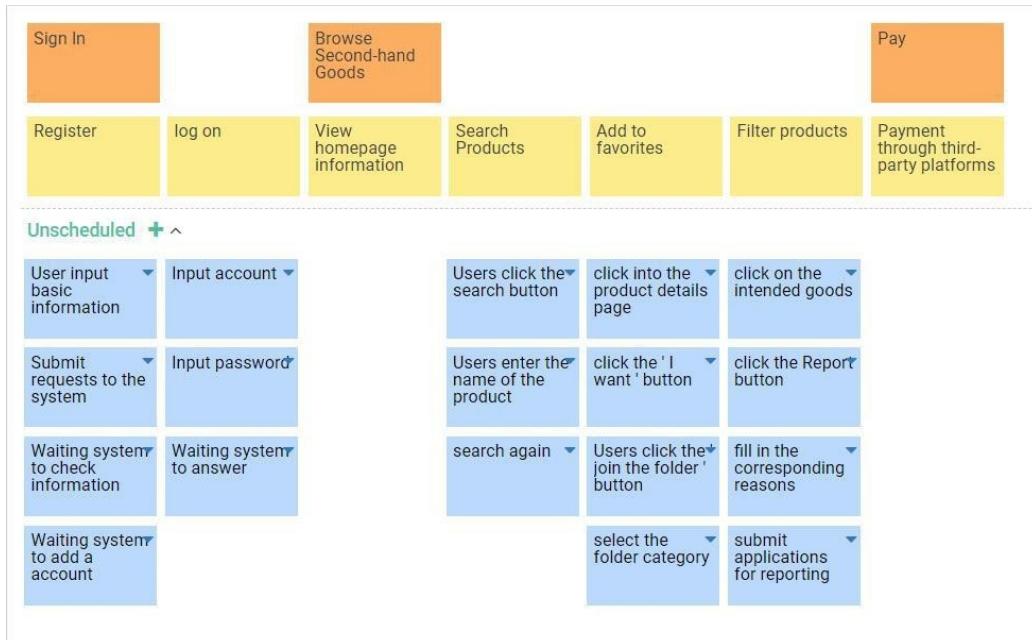
2.2.3 Community Information System



If a user wants to find out about a group, he can go to @Tongji to find out about that group.

Firstly, he or she can search for the group in the search interface and browse information related to the group, which requires a browse information use case. And group users can also submit group information through the system. Users can also submit their own comments under the group information.

2.3 User Story



Users can achieve the purpose of second-hand trading through our project. Users should first login, then browse the goods to confirm the goods they like, and finally pay. His behavior can be divided into three activities .'sign in' 'browse second-hand goods', and 'pay'.

After our analysis, users need to register before login to complete ' sing in '. These processes can be described as several epics. They are 'Register' and 'log on', respectively. To complete Register, users need to complete four user stories, namely 'User input basic information required for registration', 'Submit request to the system', 'Waiting system to check information, and Waiting system to add a account'.

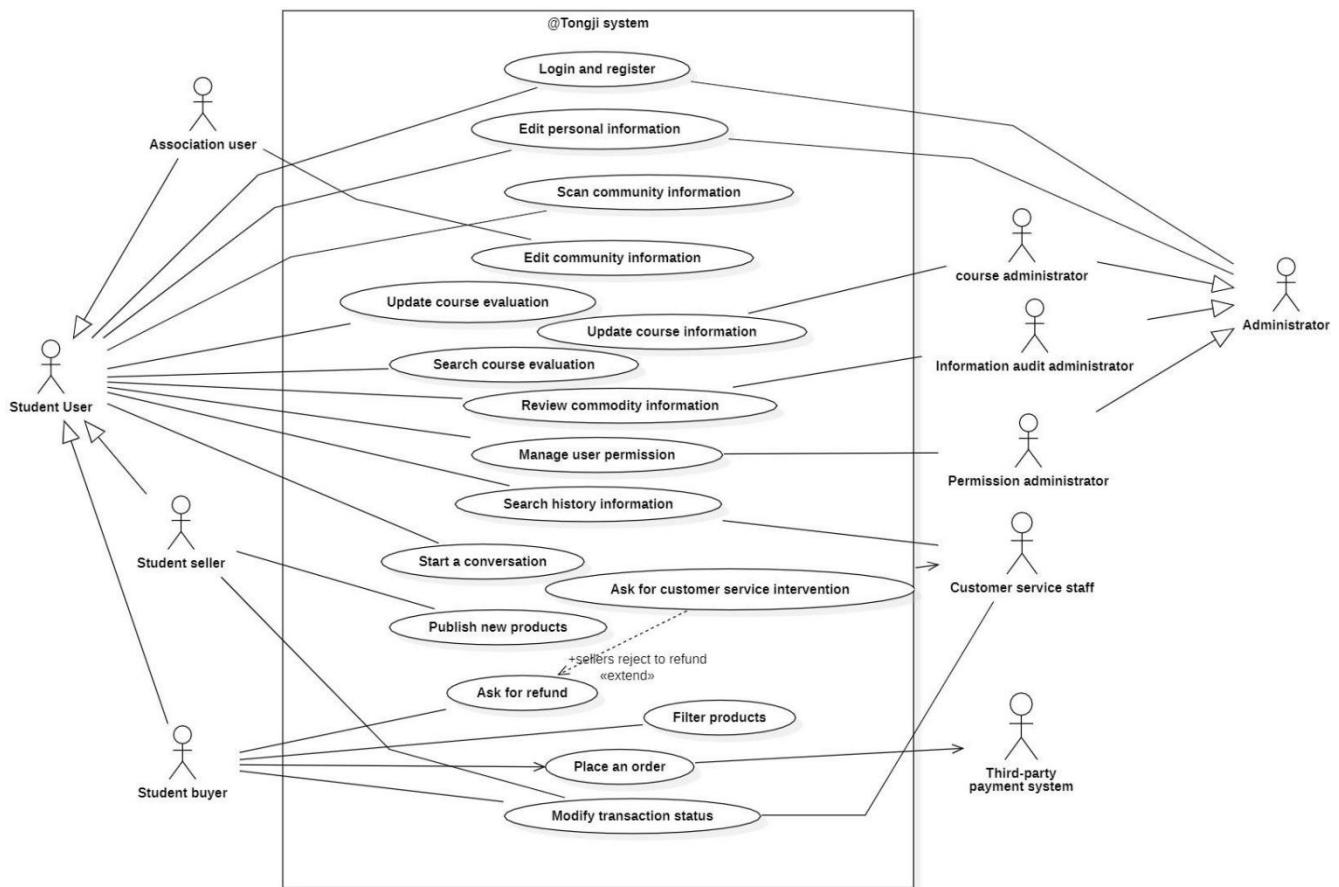
To complete ' sing in ', the user needs to enter the account, enter the password and wait for the login result. The above process can be described as three user stories, namely, 'Input account', ' Input password ', ' Waiting system to answer ', ' search again '. Through the above analysis, we design several use cases such as ' register ' and ' log on ' to help users.

After our analysis, the user completes ' Browse Second-hand Goods ' by browsing the goods, adding the collection, and finally completing the payment. During this period can also report some goods. The above process can be summarized as several epics. They are ' View Homepage Information ', ' Search Products ', ' Add to likes ', ' Filter Products ' When users complete Search Products, they need to click the search box and enter the name of the item to be searched. If they cannot find the goods they need, they need to search again. The above procedure can be described as three user stories: ' Users click the search button ', ' Users enter the name of the product ', and ' Search again '.

Through the above analysis, we design several use cases such as ' view page information ', ' search products ', ' add to favorites ', ' filter products ' to help users.

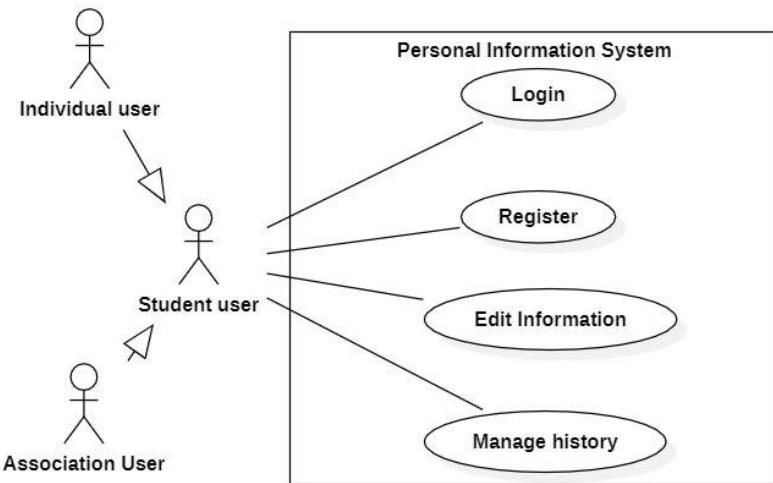
3. Use case modelling

3.1 Global View on Use Cases



3.2 Subsystems

3.2.1 Personal Information



Use case specification:

Use case: Register	
ID:UC01	
Actors:	
User	
Brief Description:	
This use case describes how a user registers an account.	
Pre-Conditions:	
The system displays the registration page.	
Basic Flows:	
1.User submit campus card photos to the system.	
2.User enters the basic information required for registration.	
3.User submits a registration request to the system.	
4.System checks whether the registration information entered by the user is legitimate.	
5.System add an account for user.	
Alternative Flows:	
Exception:	
1a.User identification failed. System rejects the user's registration request.	
2a.User incorrectly enters information. The system alerts the user to input errors, and the user can choose to reenter or quit the registration.	
Post-Conditions:	

User successfully register an account and the account is added to system.

Special Requirements

None

Use case:Login

ID:UC02

Actors:

User

Brief Description:

This use case describes how a user logs into an online system.

Pre-Conditions:

The system displays the login page.

Basic Flows:

- 1.User enter username and password.
- 2.User submits a login request to the system.
- 3.System checks that the username exists and that the username and password match.
- 4.System allows the user to log in.

Alterative Flows:

Exception 3a:

Username entered by the user does not exist, or the username and password do not match

The system prompts the user name/password input error, the user can choose to re-enter or log out,

if log out, the use case ends.

Variant 1a:

User forget password

The user enters the mobile phone number/mailbox used for registration, the system sends the verification code to the user's mobile phone number/mailbox, the user enters the verification code, and then reset the password.

Post-Conditions:

User logs into the online system.

Special Requirements

None

Use case>Edit Information

ID:UC03

Actors:

User

Brief Description:

This use case describes how the user edits personal information in a personal page.

Pre-Conditions:

User has successfully logged into the system and entered the personal page.

Basic Flows:

1. User edits personal information and saves it.
2. User submits a request to update your personal information.
3. System checks whether the personal information edited is legal.
4. System updates the user's personal information.

Alternative Flows:

Exception

3a. Personal information edited by the user is illegal.

System gives an input error prompt, and the user can choose to edit the information again or quit editing.

If quit editing, the use case will end.

Post-Conditions:

User successfully modifies personal information.

Special Requirements

None

Use case:Manage

History

ID:UC04

Actors:

User

Brief Description:

This use case describes how a user manages his or her history in a personal page.

Pre-Conditions:

The user has successfully logged in and entered the personal page.

Basic Flows:

1. User browses personal history.
2. include(add to collection)
3. include(delete from history)
4. User submits a management request for history.
5. System updates the history status for the user.

Alternative Flows:

None

Post-Conditions:

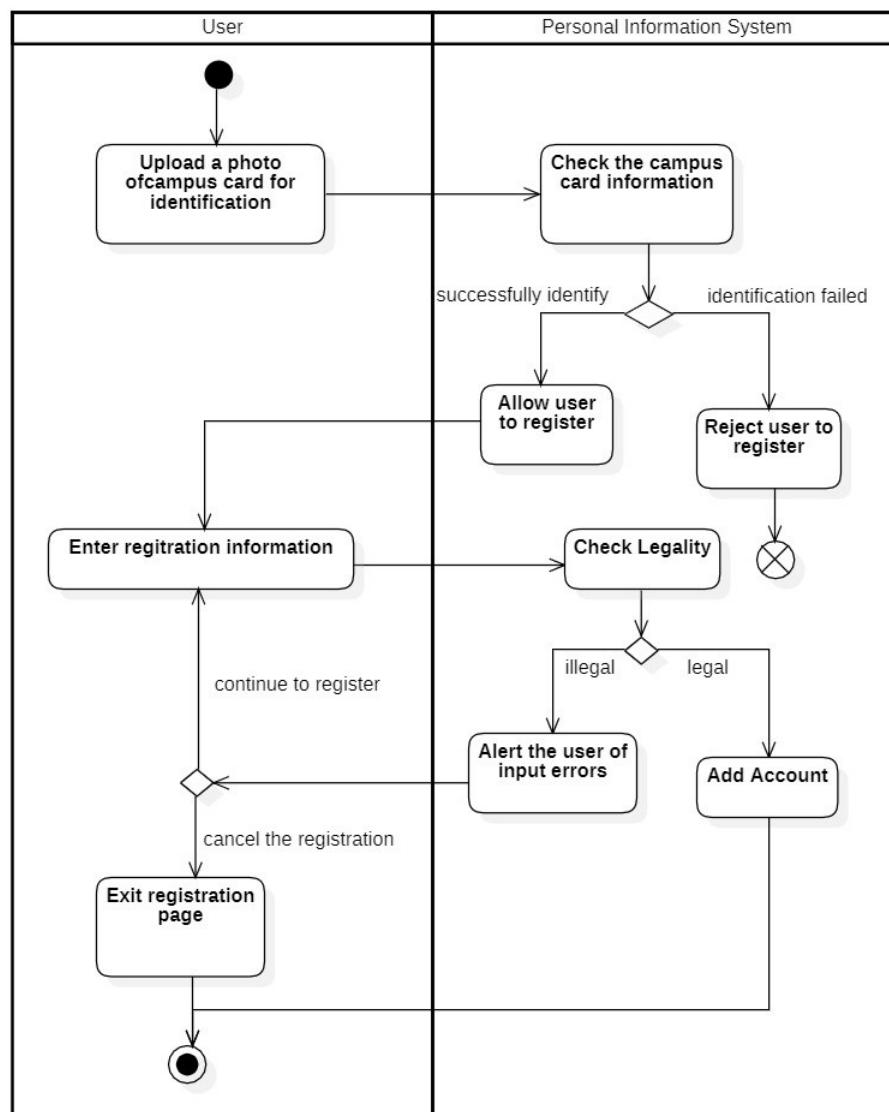
The user successfully browses and updates the status of personal history.

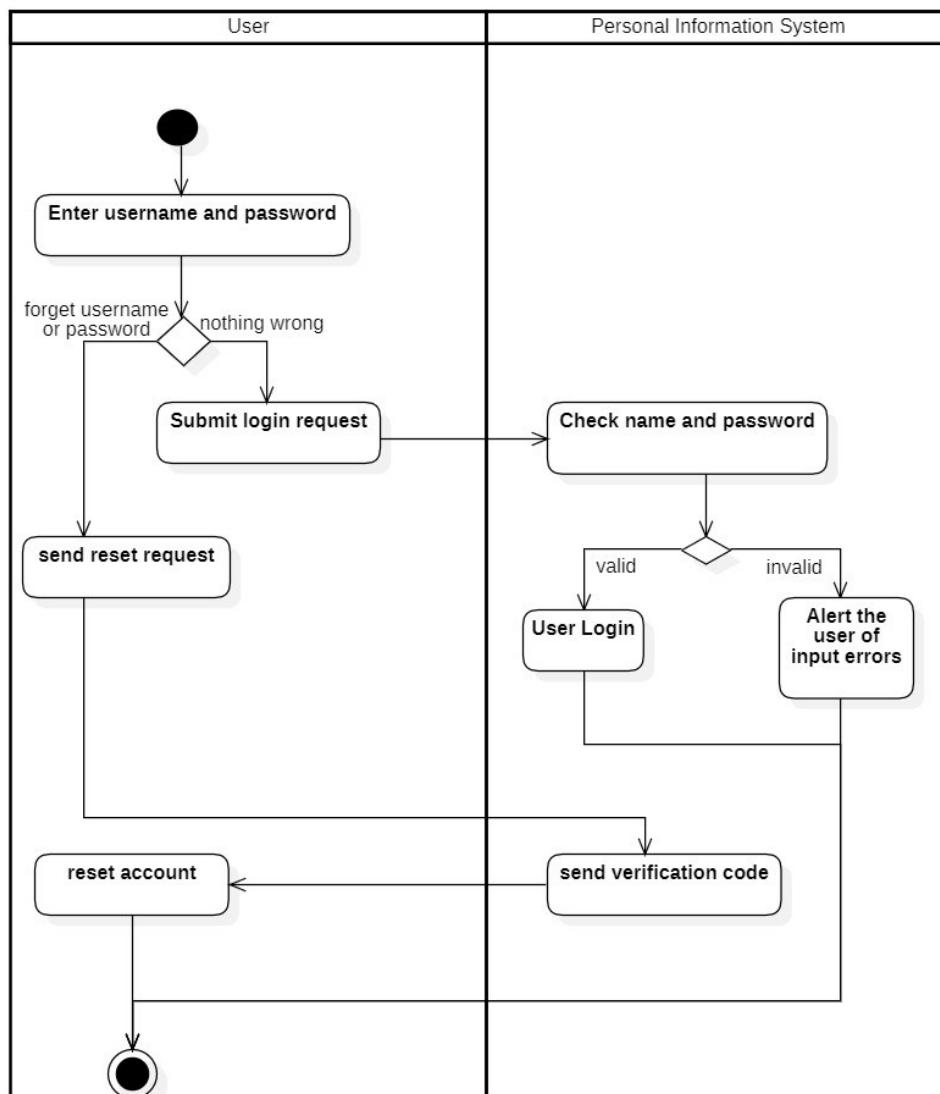
Special Requirements

None

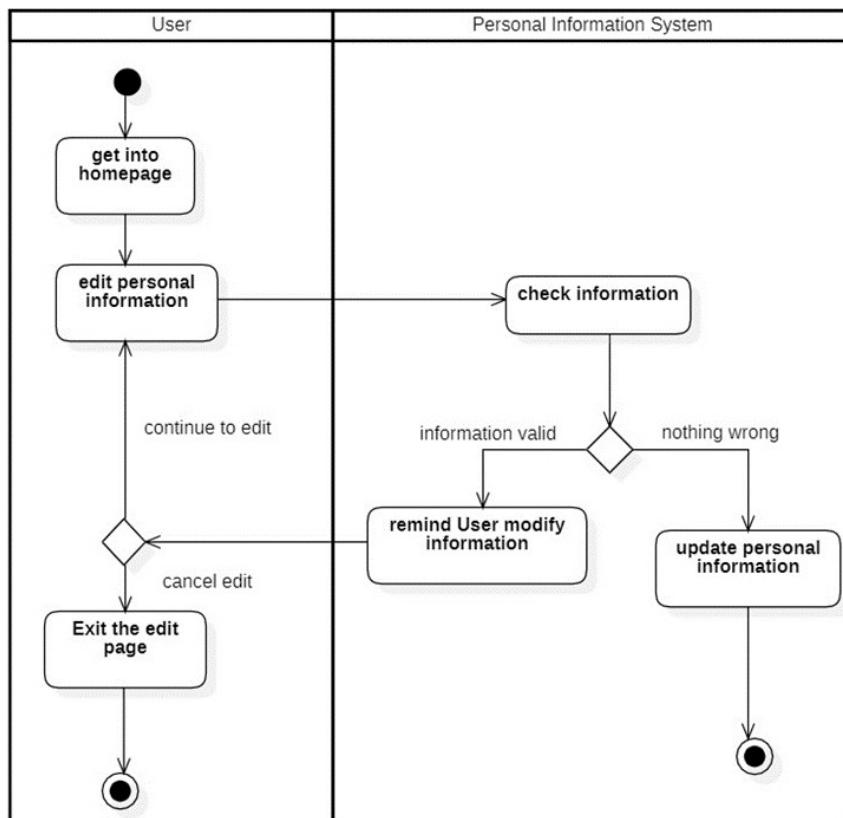
Activity Diagram

Register:

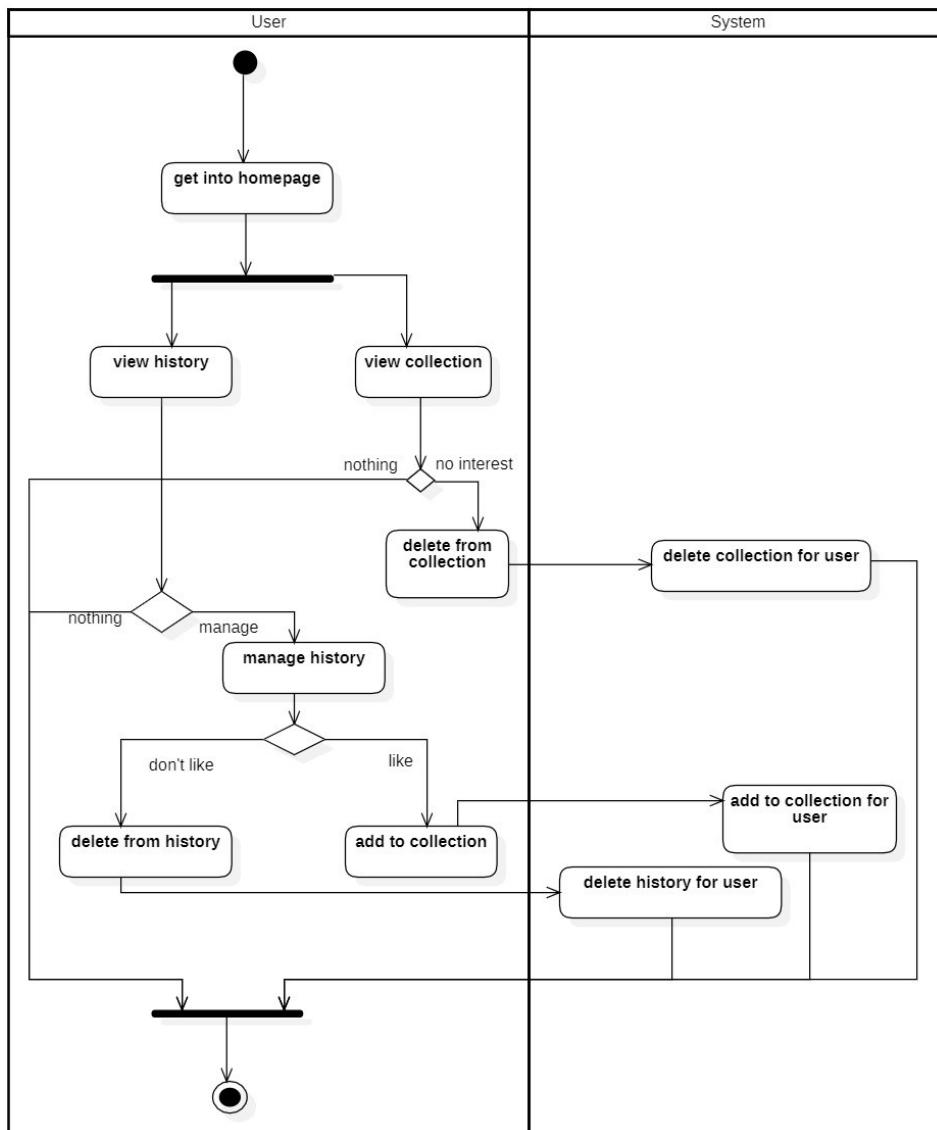


Login:

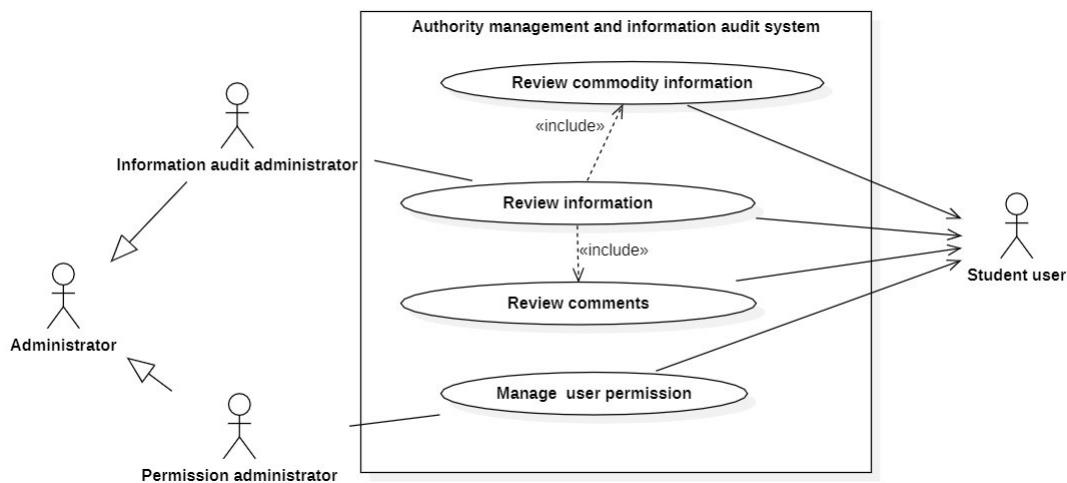
Edit Information:



Manage History



3.2.2 Authority management and information audit system



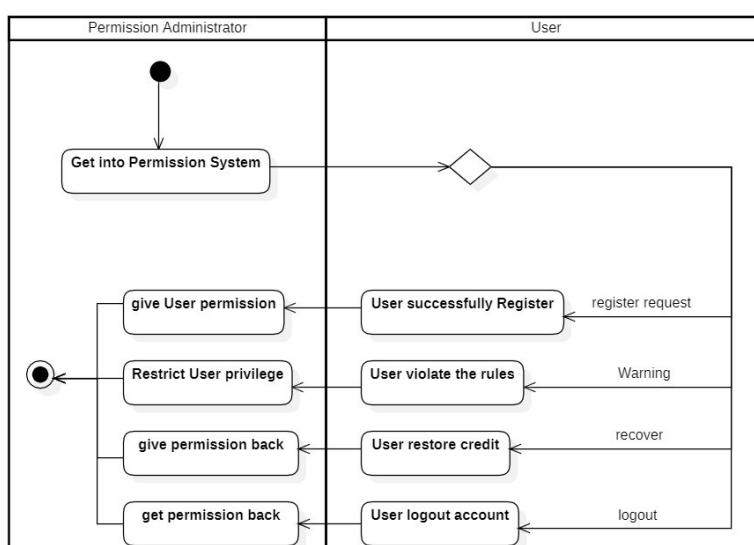
Use Case Specification:

Use case: Manage User Permission	
ID:UC05	
Actors:	
Permission Administrator	
Brief Description:	This use case describes the administration of user permissions by a permission administrator.
Pre-Conditions:	Permission administrator logs into the permission management system.
Basic Flows:	<ol style="list-style-type: none"> 1. Permission administrator logs into the permission management system. 2. Permission administrator accepts the request sent by the user and reviews the user's activities. <ol style="list-style-type: none"> 2.1 User registers an account and the permission administrator grants the authority. 2.2 User behavior is not standard, and the authority administrator restricts the user's authority. 2.3 User logs out the account, and the permission administrator takes back the authority. 3. Permission administrator manages the authority of the user. 4. Permission administrator ends the management of the user's authority.
Alterative Flows:	
Variant	<p>2a. User permission limits expire When the user permission is not permanently blocked, the permission administrator will re-grant the user permission when the release period is reached.</p>
Post-Conditions:	Permission administrator successfully manages user permissions.
Special Requirements	None

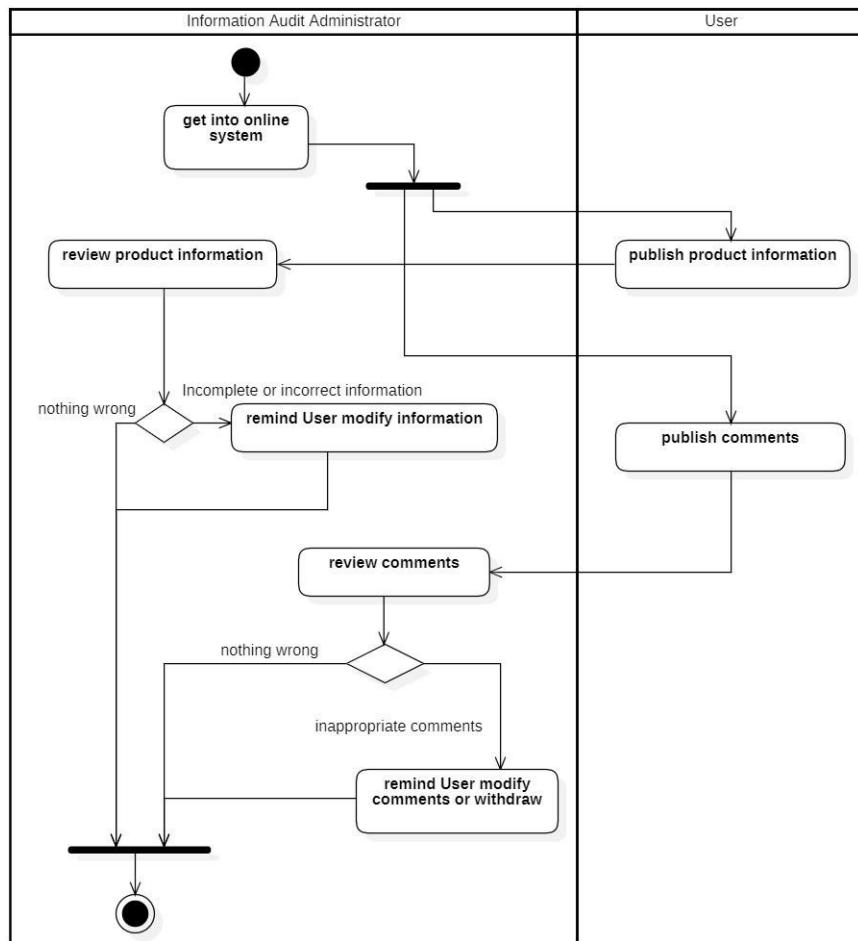
Use case:Review Information	
ID:UC06	
Actors:	Information Audit Administrator
Brief Description:	This use case describes how the information audit administrator audits information.
Pre-Conditions:	Information audit administrator logs into the management system.
Basic Flows:	<p>1.Information audit administrator logs in the audit management system.</p> <p>2.Information audit administrator reviews the information posted by users.</p> <p>3.Information audit administrator ends the review of messages and comments posted by users.</p>
Alternative Flows:	
Exception:	<p>2a.Information and comments posted by users are not standard Information audit administrator remind users to modify or withdraw information and comments posted by users.</p>
Post-Conditions:	Information audit administrator successfully reviewes the information posted by the user.
Special Requirements	None

Activity Diagram:

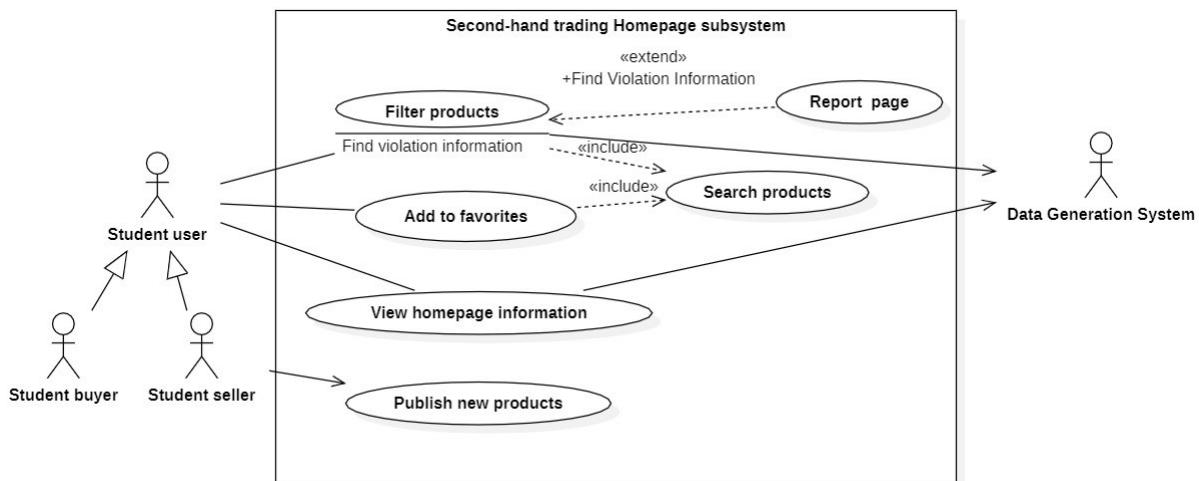
Manage User Privilege:



Review Information:



3.2.3 Second-hand trading homepage system



Use case specification:

Use case: Search Products	
ID: UC07	
Actors:	Student Buyers, Student Sellers, Data Generation System
Pre-Conditions:	The user registers and logs in to the account.
Description:	Users can accurately search for the items they want through the search bar, or fuzzy search for related items.
Basic Flows:	<ol style="list-style-type: none"> 1. The user enters the name of the product he wants to search. 2. The user clicks the "Search" button. 3. The system returns the search results and displays them according to the keywords. 4. If the system finds related products: <ol style="list-style-type: none"> 4.1 The system displays related products based on the results obtained by the search engine. 5. If no related products are found: <ol style="list-style-type: none"> 5.1. Output prompt message: "Related product not found". 5.2. Prompt the user to enter the product search again.
Alternative Flows:	<p>Alternative 2a: The user clicks the "Fuzzy Search" button:</p> <ol style="list-style-type: none"> 1. The system uses the Get intelligent search data use case of "Data Generation System" to generate intelligent search results. 2. The system displays the fuzzy searched product information in a comprehensive order.
Post-Conditions:	The user successfully searched for the target product.
Special Requirements	None

Use case: Filter Products	
ID: UC08	
Actors:	Student Buyers, Student Sellers
Pre-Conditions:	<ol style="list-style-type: none"> 1. The user has registered and logged in to the account. 2. The user has completed the product search and the search is successful.
Description:	Users can filter and sort items that meet the price range, campus, brand newness, brand and other requirements according to their own needs.
Basic Flows:	<ol style="list-style-type: none"> 1. include(Search Products) 2. The user selects the filter criteria. 3. The user enters the product interface. <Find Violation Information> 4. The user clicks the "Report" button. 5. The user fills in the corresponding reason for the report. 6. The user checks the reason for the election. 7. The user submits a report application.
Alternative Flows:	None
Post-Conditions:	The user successfully screened the target product.
Special Requirements	None

Use case: Add To favorites	
ID: UC09	
Actors:	Student Buyers, Student Sellers
Pre-Conditions:	<ol style="list-style-type: none"> 1. The user has registered and logged in to the account. 2. The user has completed the product search and the search is successful.
Description:	Users can add favorite items to their own favorites of different categories; the system supports adding new favorites.
Basic Flows:	<ol style="list-style-type: none"> 1. include(Search Products) 2. The user enters the product details page.

3. The user enters the real-time chat interface.
4. The user clicks the "Add to Favorites" button.
5. The user selects the favorite category.
6. If the favorites are not full:
 - 6.1 The system adds commodities to favorites.
7. If the favorites are full:
 - 7.1 System prompt message: "Your favorites are full, please delete some products and try again".
 - 7.2. The user enters the personal homepage.
 - 7.3. The user enters the favorite page.
 - 7.4. The user deletes some products.
 - 7.5 The user repeats the above steps.

Alternative Flows:

Alternative5a: The user creates a new favorite:

1. The user clicks the "Add to Favorites" button.
2. The user clicks the "New Favorite" button.
3. The user enters the name of the new favorite.
4. The user adds the product to a new favorite.

Exception 6a: The goods in the user's favorites are sold:

1. The system deletes the product from the favorites.
2. The system sends system messages to users.

Post-Conditions:

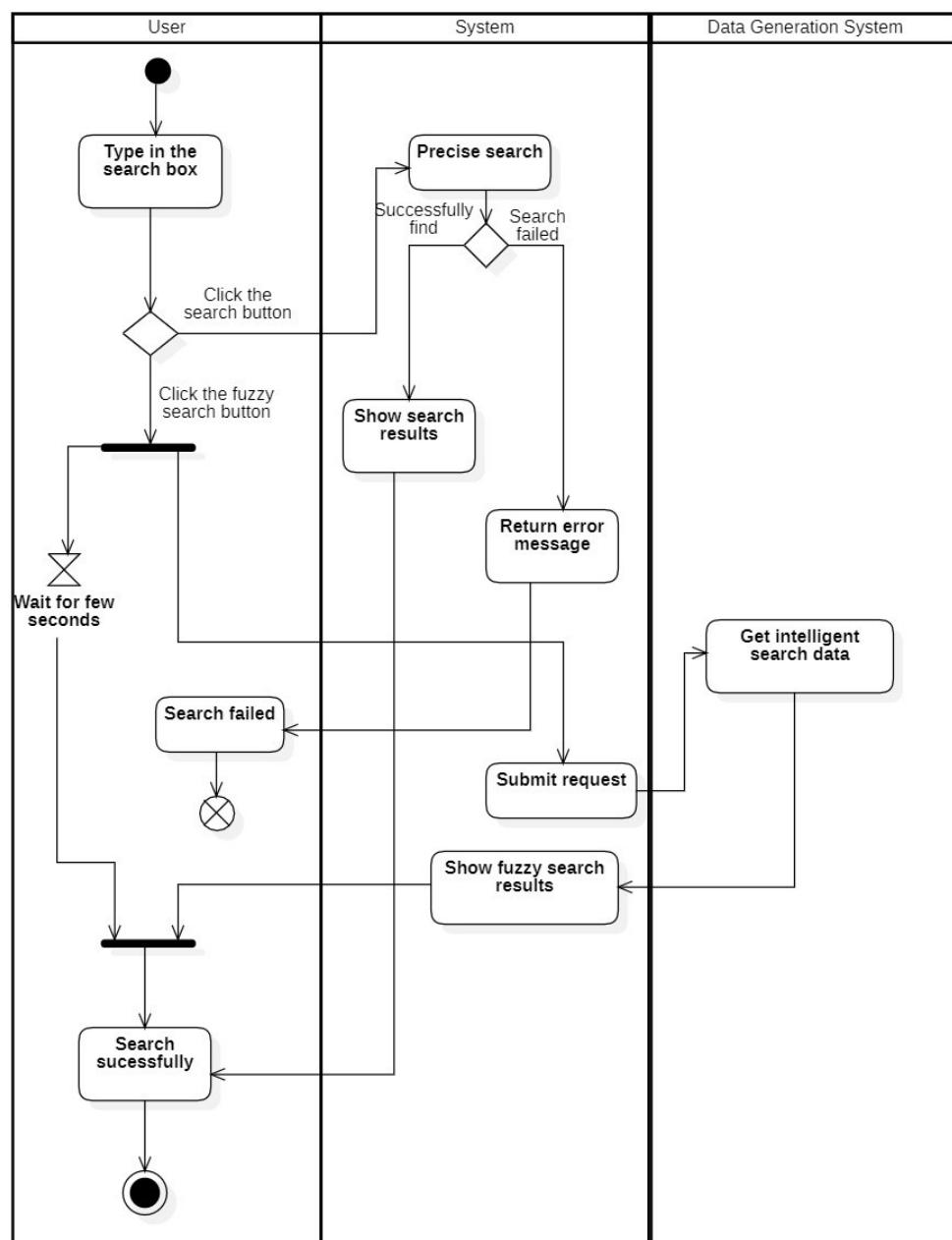
The user successfully added the product to his favorites.

Special Requirements

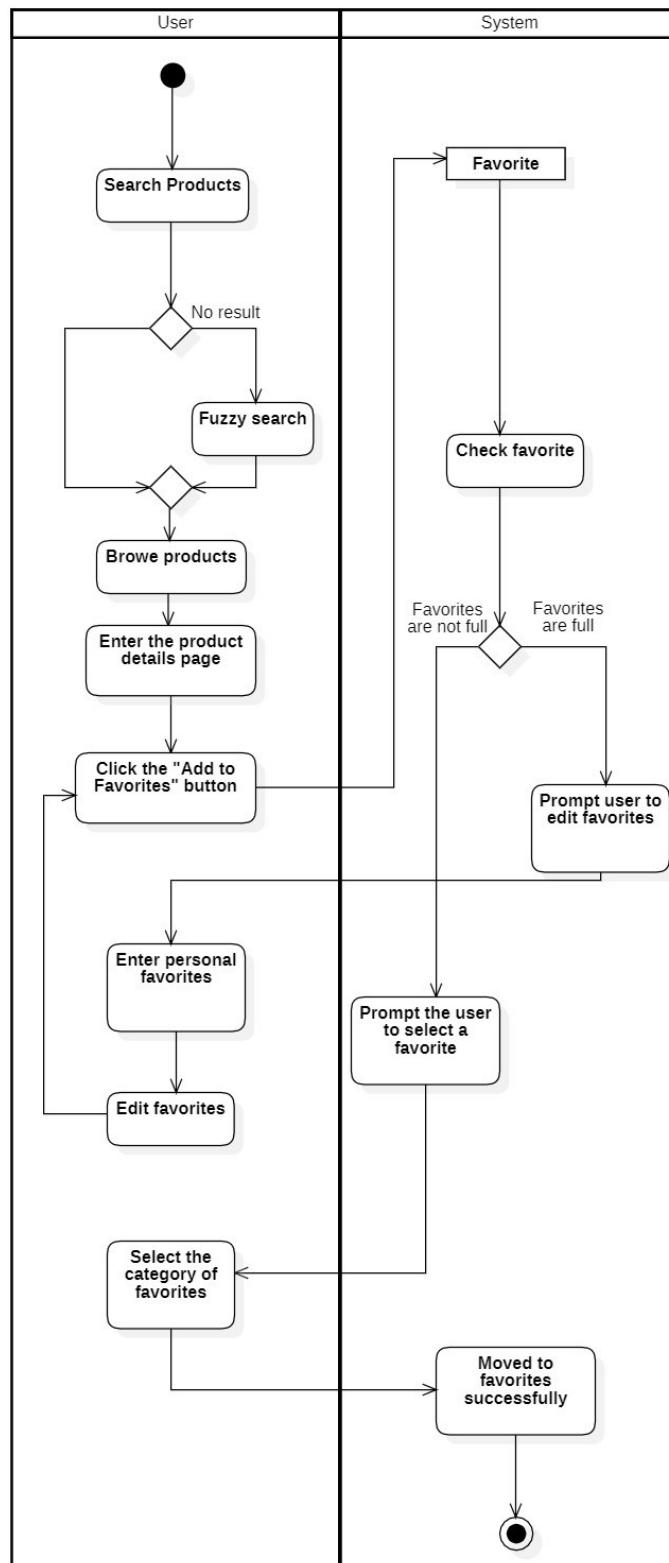
None

Activity Diagram:

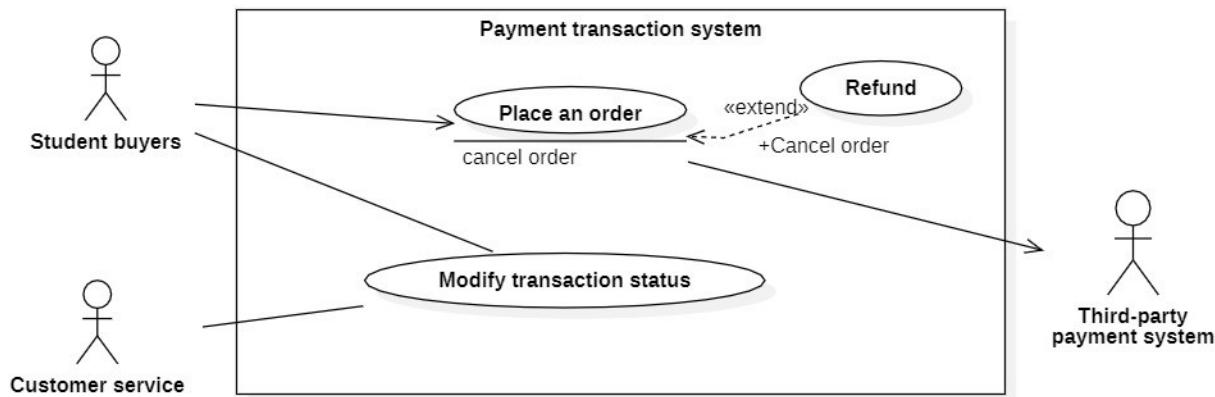
Use case: Search Products:



Use case: Add to favorites:



3.2.4 Payment Transaction System



Use case specification:

Use case: Place an order	
ID:UC10	
Actors:	Student buyers, Third-party payment system
Trigger Event:	The student buyer clicks the "Buy" button.
Pre-Conditions:	<ol style="list-style-type: none"> 1. The user has registered and logged in to the account. 2. The buyer and seller have negotiated.
Description:	Users can place an order for goods, and the money is temporarily stored in the third-party payment guarantee platform until the order is completed.
Basic Flows:	<ol style="list-style-type: none"> 1. The user clicks the "Buy" button. 2. The user enters the order details page. 3. The user fills in the necessary order information. 4. The user selects payment methods such as "WeChat Pay" and "Alipay Pay". 5. If the user clicks the "Pay" button: <ol style="list-style-type: none"> 5.1 The user uses a third-party payment platform to pay in the details page. 5.2 The third-party payment platform receives the money. 5.3 The system temporarily stores the money in the third-party guaranteed payment platform. 6. If the user does not click the "Pay" button, click the "Cancel button": <ol style="list-style-type: none"> 6.1 The page carries out corresponding order temporary storage and payment prompts. 6.2 The system gives 15 minutes of order retention time.
Alternative Flows:	<p>Exception5a: The user wants to cancel the order within half an hour after payment:</p> <ol style="list-style-type: none"> 1. The user clicks the "Cancel Order" button.

- | |
|--|
| 2. The user fills in the reason for canceling the order and submits the application. |
| 3. Cancel the order after the system review, and the money will be refunded. |

Post-Conditions:

The user successfully placed an order.

Special Requirements

The user needs to bind a third-party payment platform account.

Use case: Publish New Product

ID:UC11

Actors:

Student Sellers

Pre-Conditions:

- | |
|--|
| 1. The user has registered and logged in to the account. |
|--|

Description:

Users can add favorite items to their own favorites of different categories; the system supports adding new favorites.

Basic Flows:

- | |
|--|
| 1. Student sellers click to enter the product release page. |
| 2. The seller user edits the detailed information of the product, including pictures, description, price, release campus, etc. |
| 3. Student sellers click the "Post" button. |

Alternative Flows:

None

Post-Conditions:

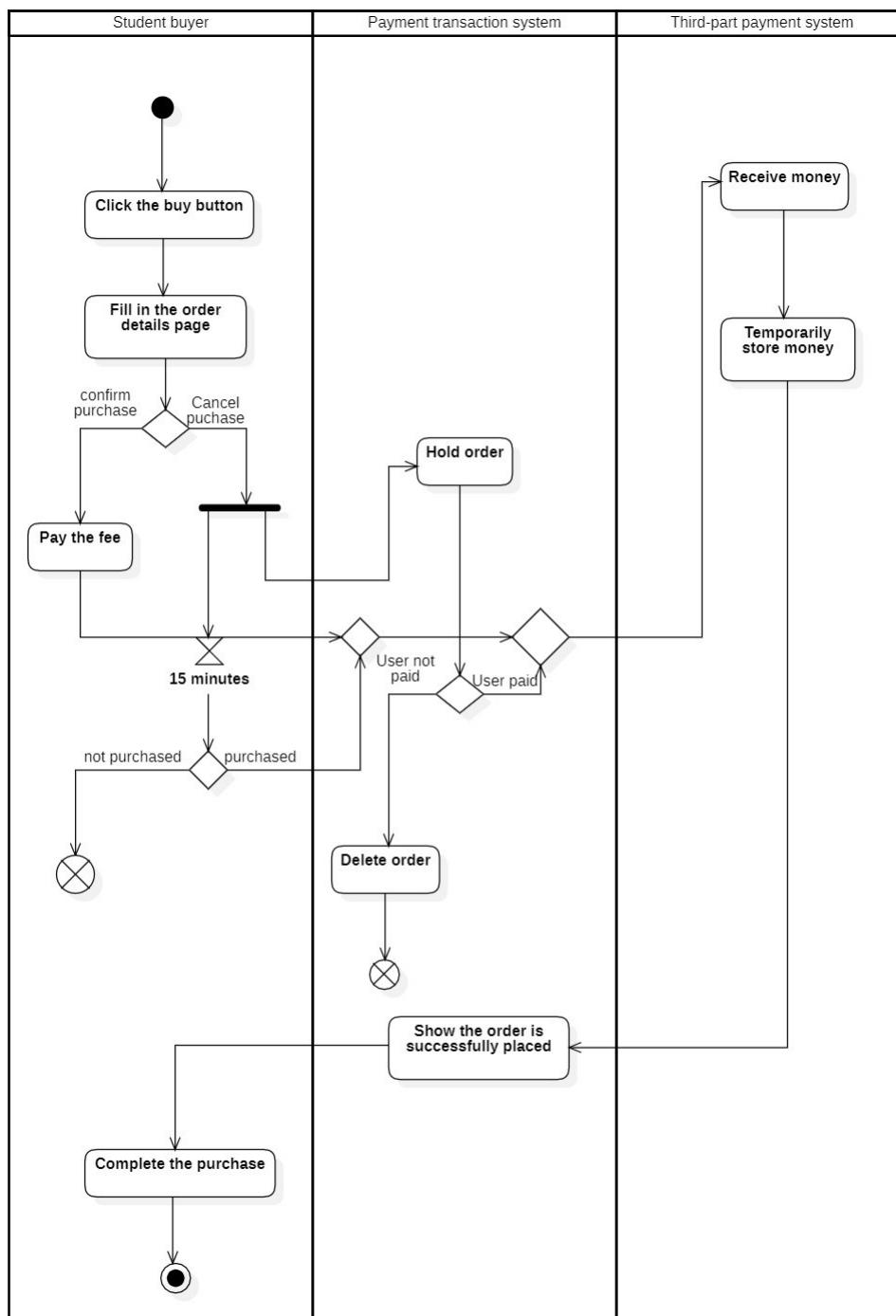
Student sellers successfully published their products.

Special Requirements

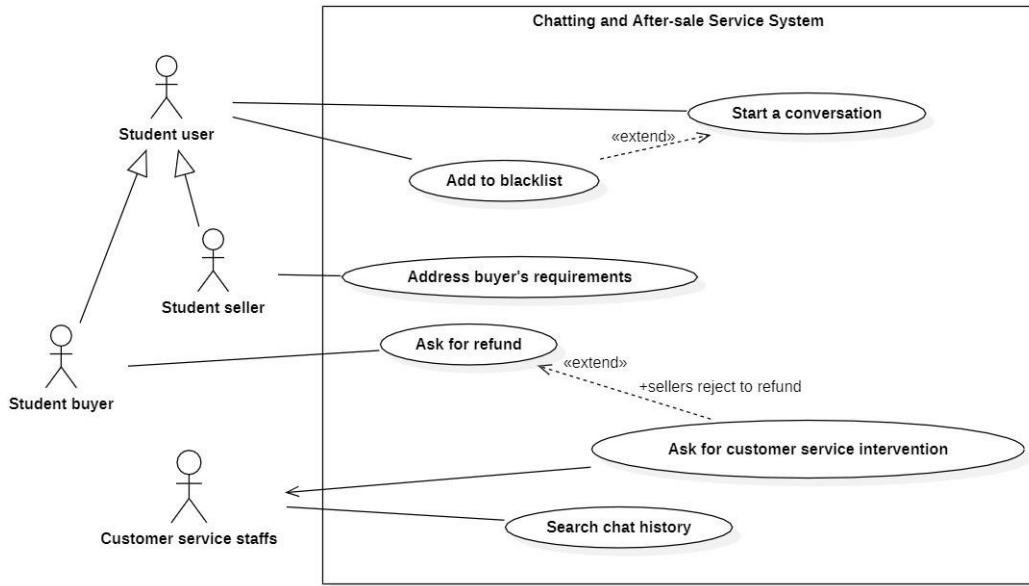
The user's credit rating meets the lowest requirements for publishing products.

Activity diagram:

Place An Order:



3.2.5 Chatting and After-sale Service System

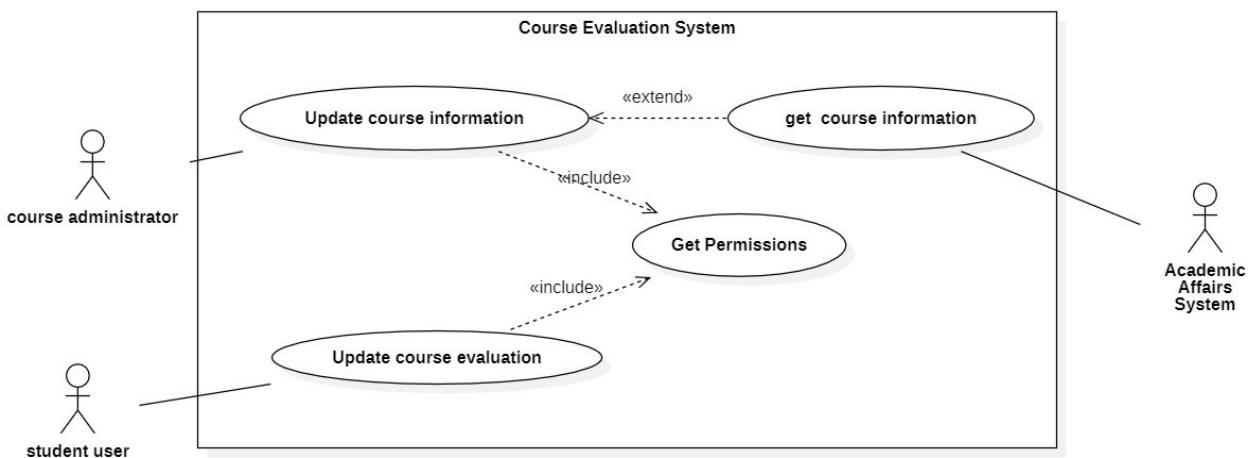


Use case specification:

Use case: Start a conversation	
ID: UC12	
Actors:	student buyer, student seller, customer service staff.
Trigger Event:	Buyer starts conversation with seller.
Pre-Conditions:	Buyer intends to purchase the listed items
Basic Flows:	<ol style="list-style-type: none"> 1. Buyer starts conversation through in the product details page. 2. Buyer ask for more detailed information from seller. 3. Buyer and seller reach an agreement.
Alternative Flows:	<ol style="list-style-type: none"> 1. Buyer and seller can't reach an agreement. 2. Buyer decides not to buy the good. 3. Buyer ask for customer service intervention.
Post-Conditions:	Buyers decide whether to buy the item or not.
Special Requirements	None.

Use case: Ask for refund	
ID: UC13	
Actors:	student buyer, student seller, customer service staff.
Trigger Event:	Buyer starts conversation with seller.
Pre-Conditions:	Buyer needs to exchange or return the goods.
Basic Flows:	<ol style="list-style-type: none"> 1. Buyer ask for return and refund. 2. Communication between buyers and sellers. 3. Buyer and seller reach an agreement to refund or not refund.
Alternative Flows:	<ol style="list-style-type: none"> 1. Buyer and seller can't reach an agreement. 2. Buyer ask for customer service intervention. 3. Customer service staff judge the responsible subject. 4. Refund to the buyer or send money to the seller.
Post-Conditions:	Complete the transaction.
Special Requirements	None.

3.2.6 Course Evaluation System



Use Case Specification:

Use case:Update course information	
ID:UC14	
Actors:	
administrator	
Pre-Conditions:	
Academic Affairs System Course Updated	
Description:	
When there are changes in the school curriculum, the administrator needs to update the course information available for evaluation immediately.	
Basic Flows:	
1.The administrator enters the administrative account and logs in 2.include(Get Permissions) 3.The system verifies the administrator's identity and grants permissions 4.Administrator applies to access the course interface of the Academic Affairs System 5.extend(get course information) 6.Administrators enter the course management interface of this system 7.The administrator add or delete the corresponding courses 8.The administrator clicks the "Save" button 9.The page indicates that the update was successful 10.Administrator issues course update notifications to users	
Alternative Flows:	
Exception 4a.If the application fails, contact the faculty administrator 1.If the administrator does not update 1.1 The system regularly reminds administrators 1.2 Student users send reminder notifications to administrators	
Post-Conditions:	
Course information updated, student users receive notification of updates	
Special Requirements	
none	

Use case:Update course evaluation	
ID:UC15	
Actors:	
student user	
Pre-Conditions:	
Students have logged in	
Description:	
Student who has taken the course has been given permission to update the course evaluation.	
Basic Flows:	

- 1.Student users click to enter the corresponding course evaluation interface
 - 1.1 Student users enter the evaluation screen by searching for courses
 - 1.2 Student users are invited to the course evaluation screen through the system
- 2.Systematic review of student evaluation course permissions
 - 2.1 If a student receives an invitation to evaluate a course, no review privileges are required
- 3.Students evaluate the course
- 4.The page indicates whether the evaluation is successful or not

Alternative Flows:

Alternative 1a:The system invites student users to evaluate the corresponding courses

- 1.Students search for courses and click to invite evaluations
- 2.The system sends invitations to people who have taken courses by accessing student course selection information
- 3.Invitation notifications appear on the user's home page

Alternative 3a:Students change posted ratings

- 1.1 Students click on the "Publish" button
- 1.2 The system prompts "You have already posted this course evaluation, re-evaluation will overwrite the original evaluation, do you want to continue"
 - 1.2.1 The user selects "Yes" to make changes to the evaluation
 - 1.2.2 User selects "No" to exit the interface

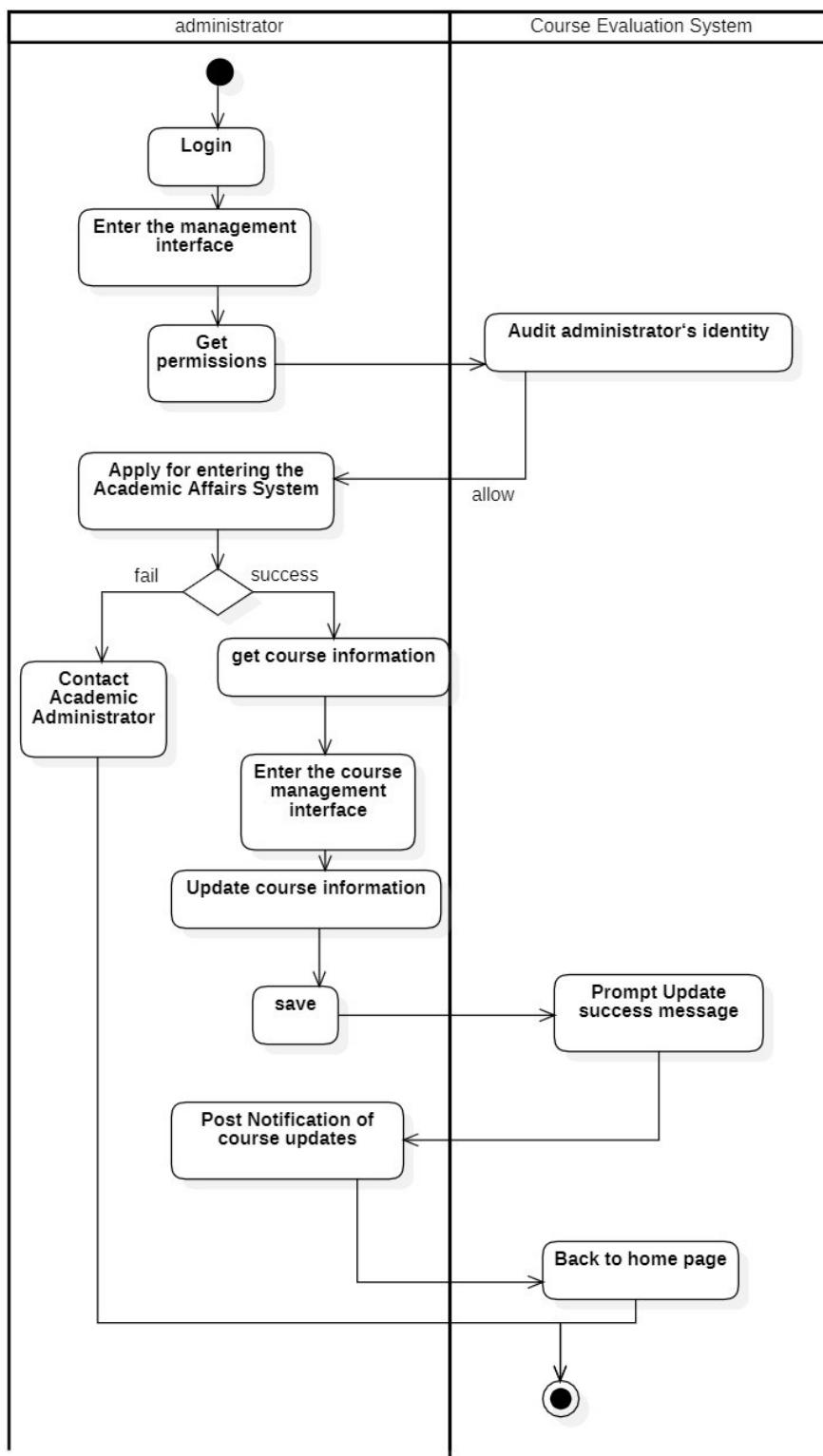
Post-Conditions:

Course Evaluation Has Updated

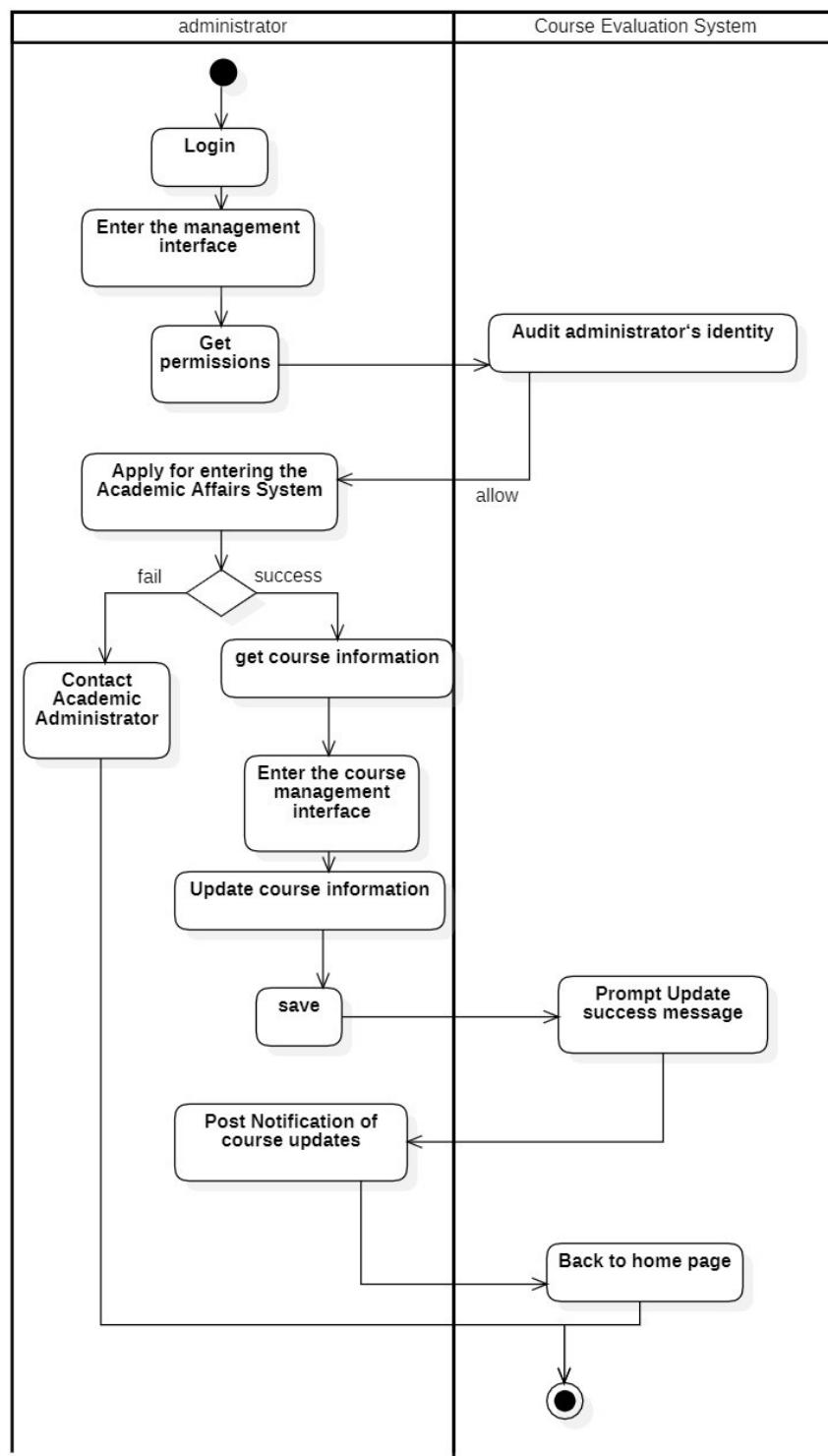
Special Requirements

none

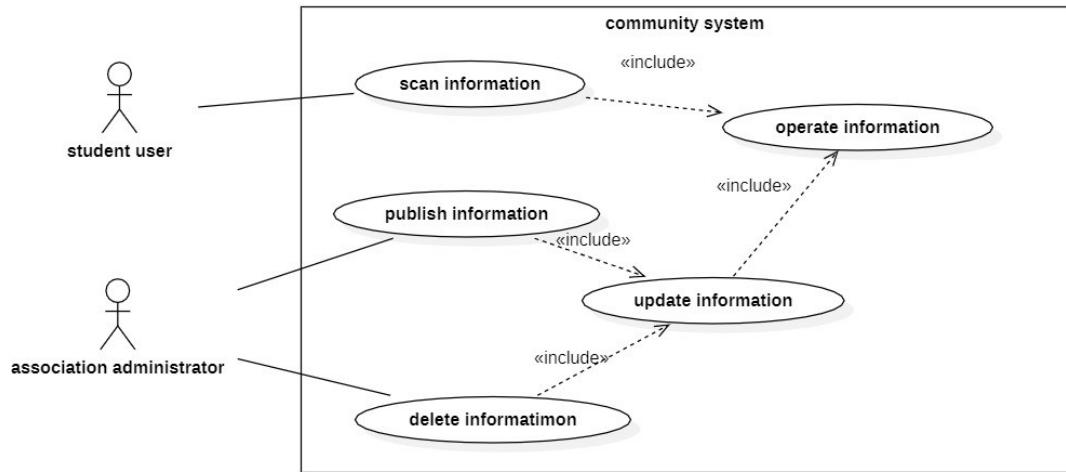
Activity Diagram:

Update course evaluation:

Update course information:



3.2.7 Association Management System



Use case specification:

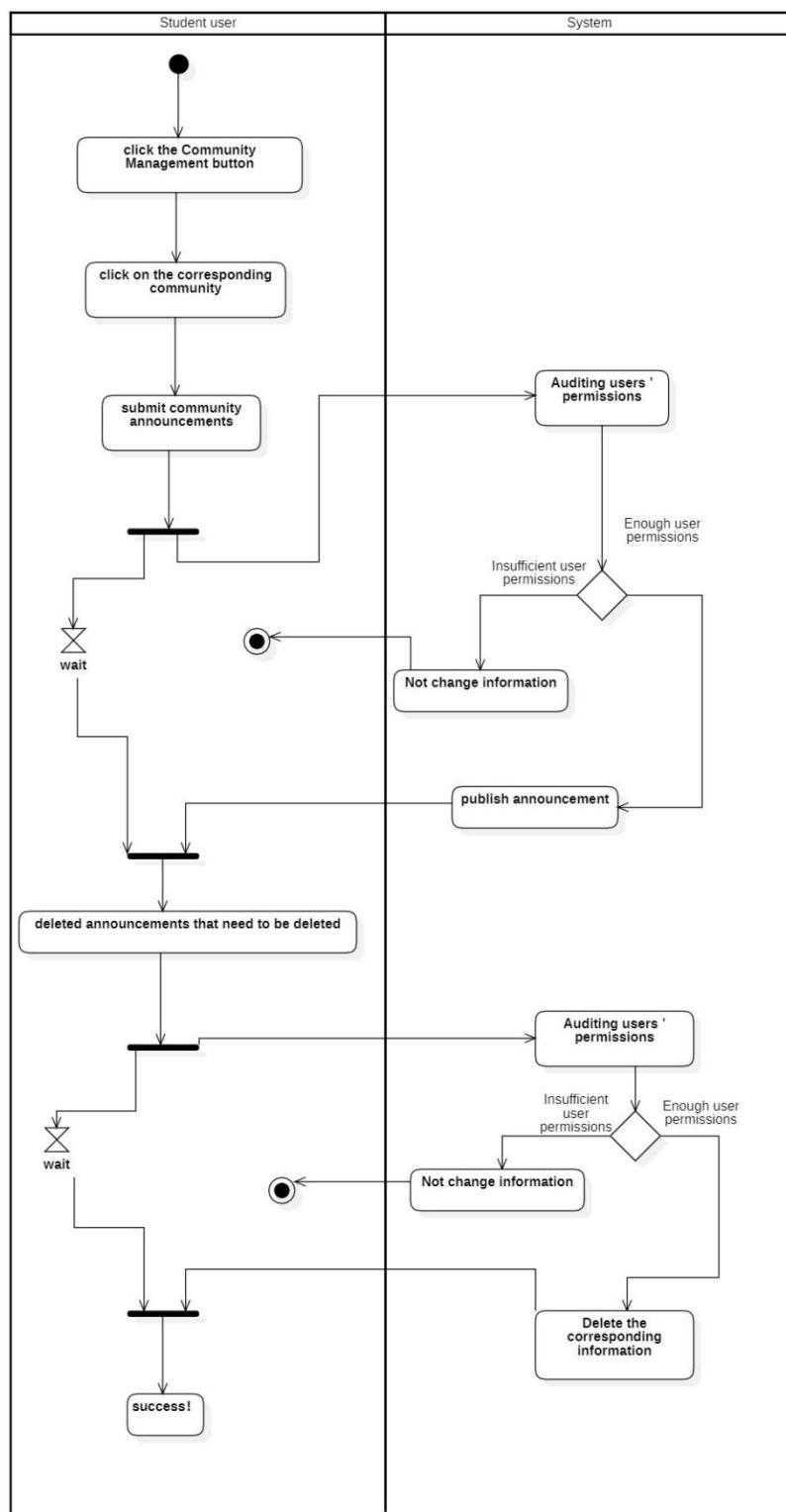
Use case: Scan information	
ID: UC16	
Actors:	Student, Association administrator
Pre-Conditions:	<ol style="list-style-type: none"> 1. Users have successfully registered 2. User login status
Basic Flows:	<p>include(operate information)</p> <ol style="list-style-type: none"> 1. Users click the Community Management button 2. Users click on the corresponding community 3. System returns information about selected communities 4. Selection of corresponding communities
Alternative Flows:	<ol style="list-style-type: none"> 1. The community administrator updates the information when the user gets it <ol style="list-style-type: none"> 1.1 System prompts users to update information
Post-Conditions:	<ol style="list-style-type: none"> 1. Student users have successfully browsed community information
Special Requirements	None

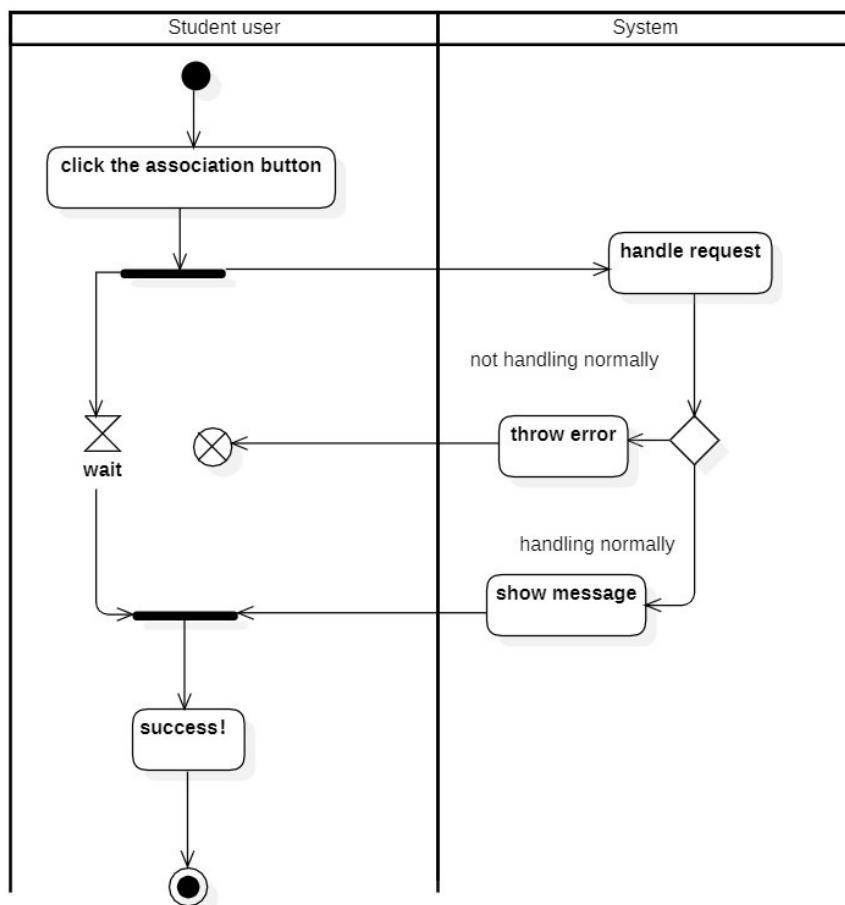
Use case:publish information	
ID:UC17	
Actors:	
association administrator	
Pre-Conditions:	
1. Users are registered and logged in 2. User identity as community administrator	
Basic Flows:	
include(update information) 1.The community administrator enters the community page 2. Community administrator and click the announcement button 3. Announcements required for community administrators to submit community announcements 4. Whether the page prompt update is successful or not	
Alternative Flows:	
3. Information uploaded by the administrator does not meet the requirements of the database 3.1 No updates are made and exceptions are thrown	
Post-Conditions:	
The community administrator issued a new community letter announcement	
Special Requirements	
None	

Use case:Delete information	
ID:UC18	
Actors:	
Association administrator	
Pre-Conditions:	
1. Users are registered and logged in 2. User identity as community administrator	
Basic Flows:	
Include(update information) 1.The community administrator enters the community page 2.Community administrator and click the delete announcement button 3. Community administrators select announcements that need to be deleted 4. Whether the page prompt update is successful or not	
Alternative Flows:	
1. Insufficient competence of community administrators to delete 1.1, insufficient system prompt authority 1.2 No change to the announcement	
Post-Conditions:	
Community Administrator Successfully Updates Community Information	

Activity Diagram:

Update Information:



Scan Information:

4 Glossary of terms

Terms	Definition
Personal information	User's personal details, including personal profile picture, courses taken, gender, date of birth, college and major, and grade
Manage history	User's management of personal browsing history. Users can add the content they are interested in to their favorites and delete the content they are not interested in directly
Favorites	Users can add the items they are interested in to favorites
Course information	Specific information of the course, including the time and place of the course, the teacher, the course materials, the assessment method, the course content, etc.
Course evaluation	Users' evaluation of elective courses
Permission	Users need permissions to use the system normally, and users with different identities have different permissions. For example, only the club administrator can release the club information. Users are also not allowed to sell illegal items, or their rights will be limited. Only users who have taken an elective course can rate that course.
Data generation system	A system that generates recommended content for users based on a person's browsing history.
Publish products	Users publish information in the second-hand trading system.
Order	User's order in a second-hand trading system.
Transaction status	The process of the transaction of the order status in second-hand trading system. There are mainly paid, to receive goods, received goods, to refund, refund these several states.
Third-party payment system	Third party payment platform, such as Alipay, WeChat Pay.

5 Supplementary specification

5.1 Performance

Performance is an important factor of our system, because student buyers want to look for commodities and view commodities as quickly as possible.

- At any given time, the system can support more than 2000 users to use the central database at the same time, and at any time can support up to 500 users to use the local server at the same time.
- Single user session response time does not exceed 5 seconds.
- The system can predict sales within 30 minutes and provide timely decision-making information.
- The system can generate reports on bestsellers and most profitable student buyers within an hour.

5.2 Reliability

- The system must be able to operate uninterruptedly for 7*24 hours.
- The system can automatically backup data regularly, and can use backup data to restore when a fault occurs.
- The starting time of the back end of the system should not exceed 30 minutes.
- If the system fails, the recovery time should not exceed 2 hours.
- The session time must be maintained for 30 minutes each time when the user is inactive.
- When an error occurs in a single session, it does not affect other sessions.

5.3 Security

- When customer shopping on our platform, most of them will choose to pay for the order online. The commodities store support Ali-Pay. Encryption technology is used to ensure that any information entered on the payment page can be safely transmitted to their center. They will also use huge amounts of data and cloud computing to determine whether there is a risk in users' payment behavior in time. The system can automatically backup data regularly, and can use backup data to restore when a fault occurs.
- student buyers' information in the system is confidential, so only with the password can student buyers log into the virtual commodities store.

5.4 Availability

- student buyers can have access to our platform for 24 hours every day and 365 days every

year when student buyers' devices are connected to the internet.

5.5 Maintainability

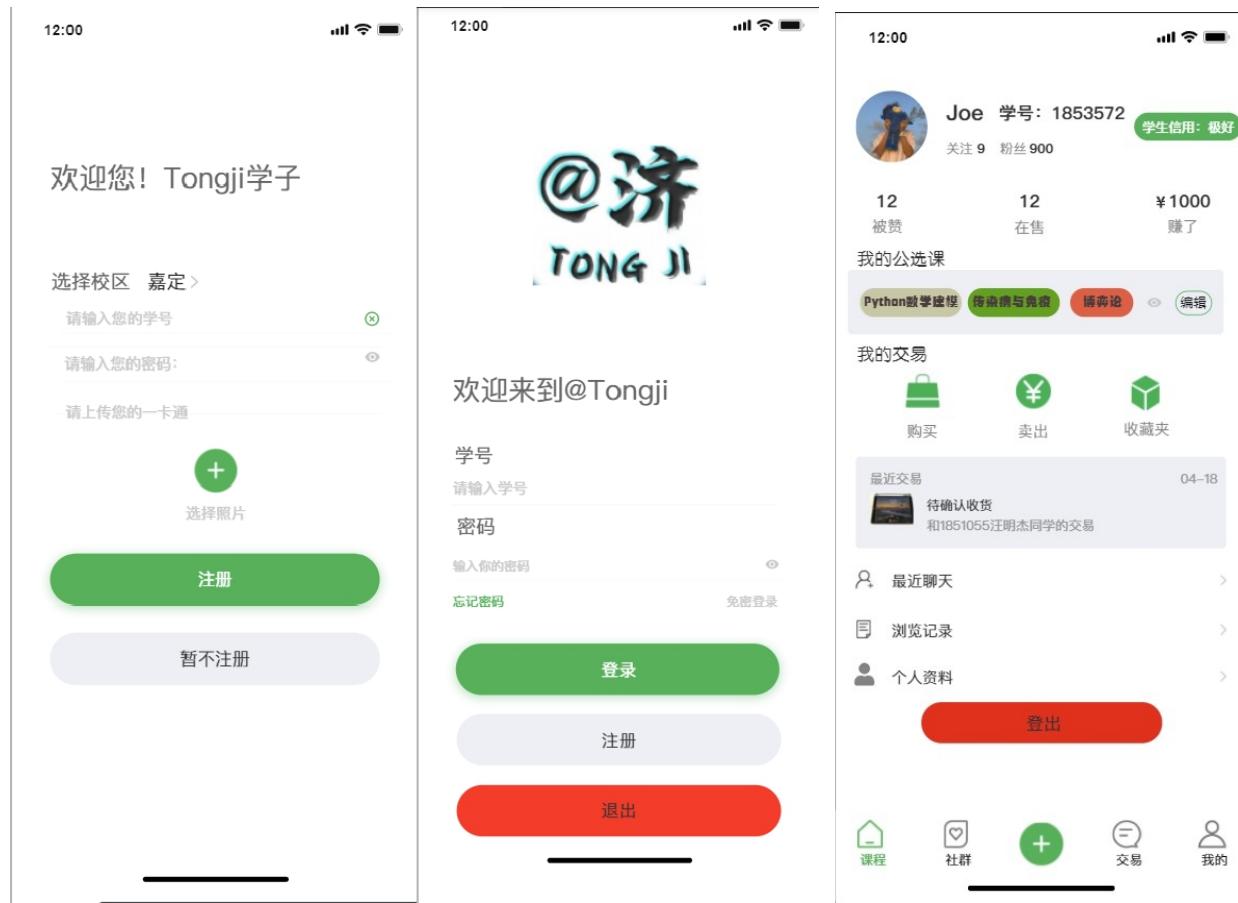
- When users use this system, if there are abnormalities or errors, a prompt interface will pop up, allowing users to submit the error information to the background, for background engineers to optimize and process.

5.6 Understandability

- Our platform directly faces users, so we need to design an online shopping mall with the characteristics of simple and easy to use, concise and intuitive, so as to attract student buyers in the first step. If the interface of the platform is too complicated, it will make student buyers feel confused, so that they will not register, so that the revenue of enterprises will be reduced.

6. Initial snapshots of the system's user interface

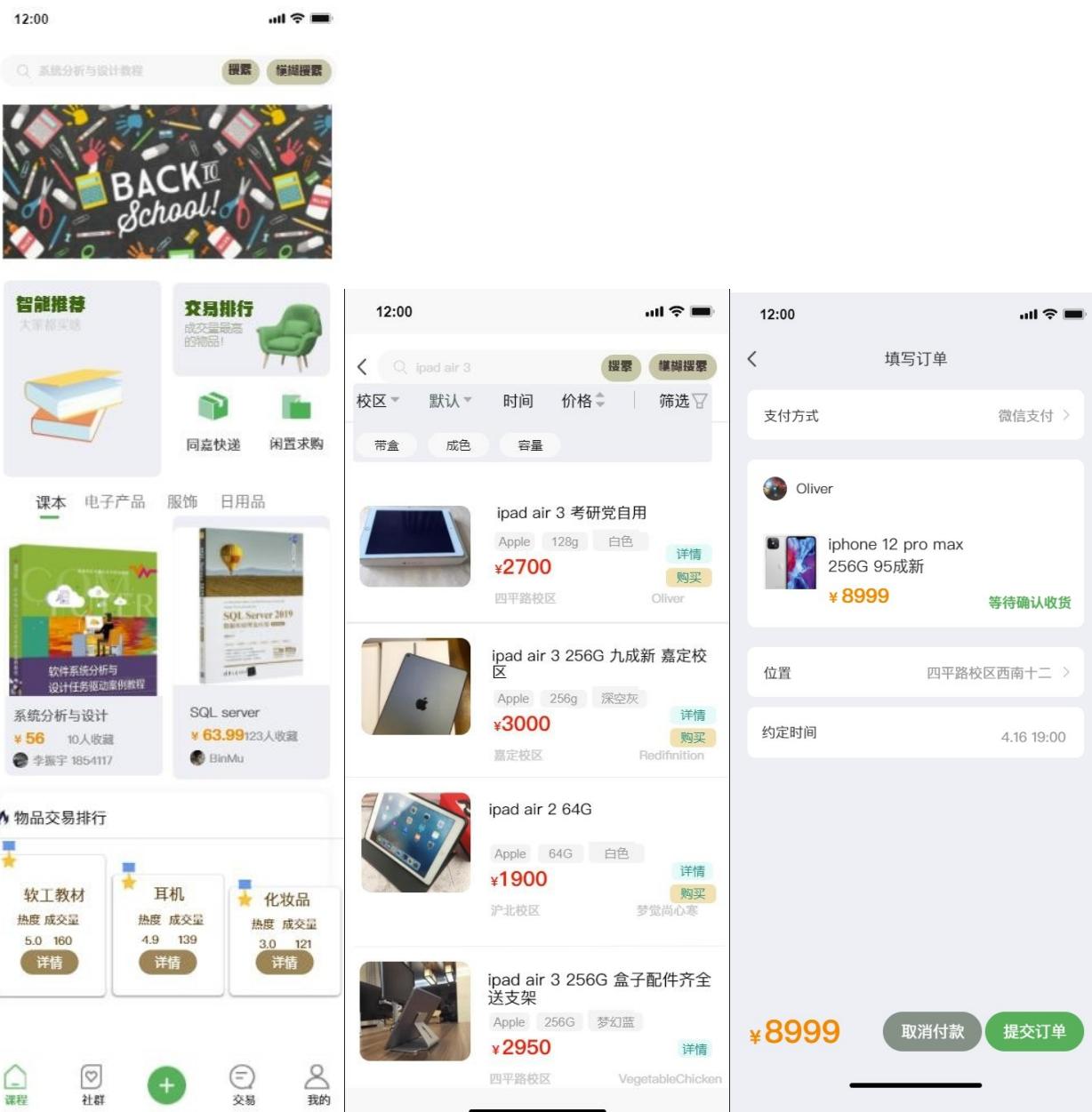
The following UI prototypes were completed using the online UI prototyping software "Modao".



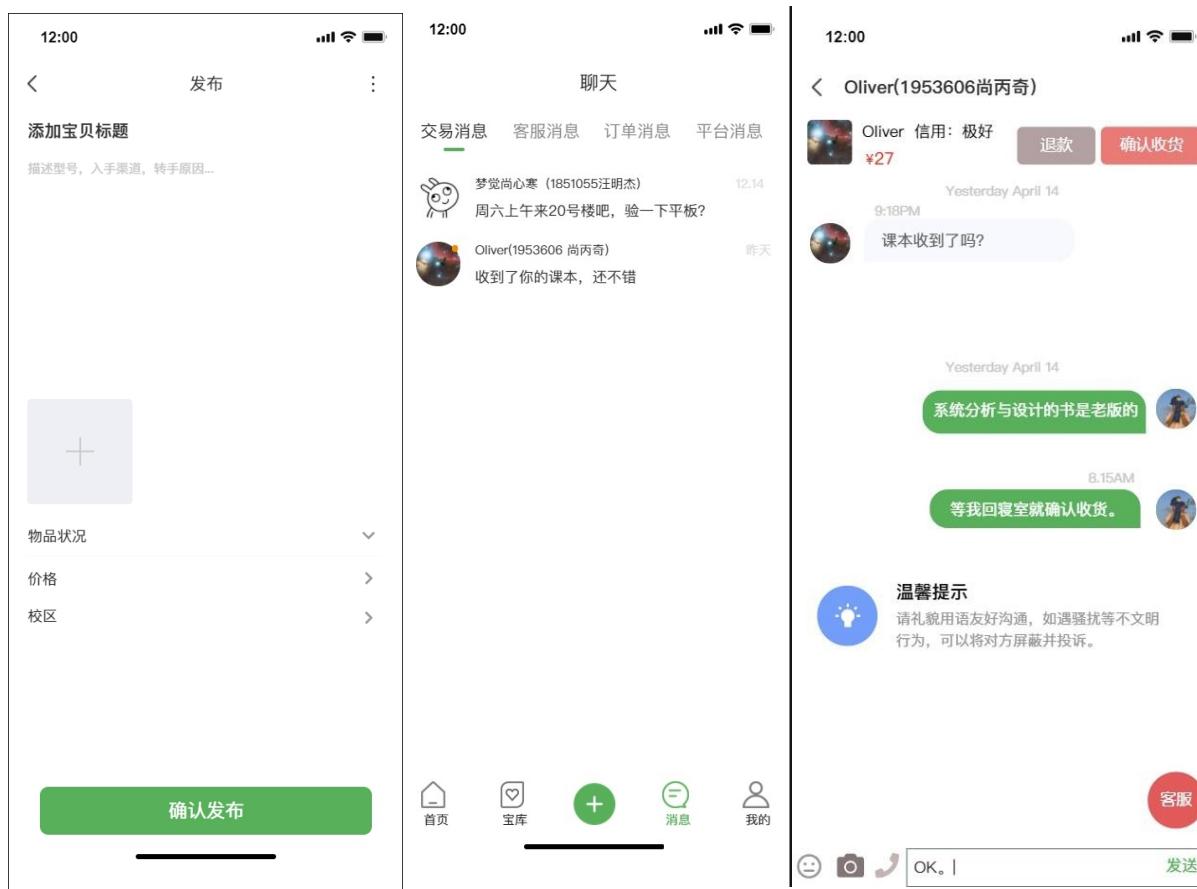
The above three UI prototype snapshots are the interface for individual users to register by uploading the all-in-one card, the login interface, and the personal homepage interface.

Firstly, the user selects the campus, enters the student number, sets the password, and registers the account by uploading the corresponding one-card picture.

Secondly, the user enters the login interface and enters the account password to log in to the personal homepage. The personal homepage mainly displays the user's credit, public elective courses that have been taken (which can be automatically generated by importing the academic system timetable), transaction records, browsing records, personal data and other modules.

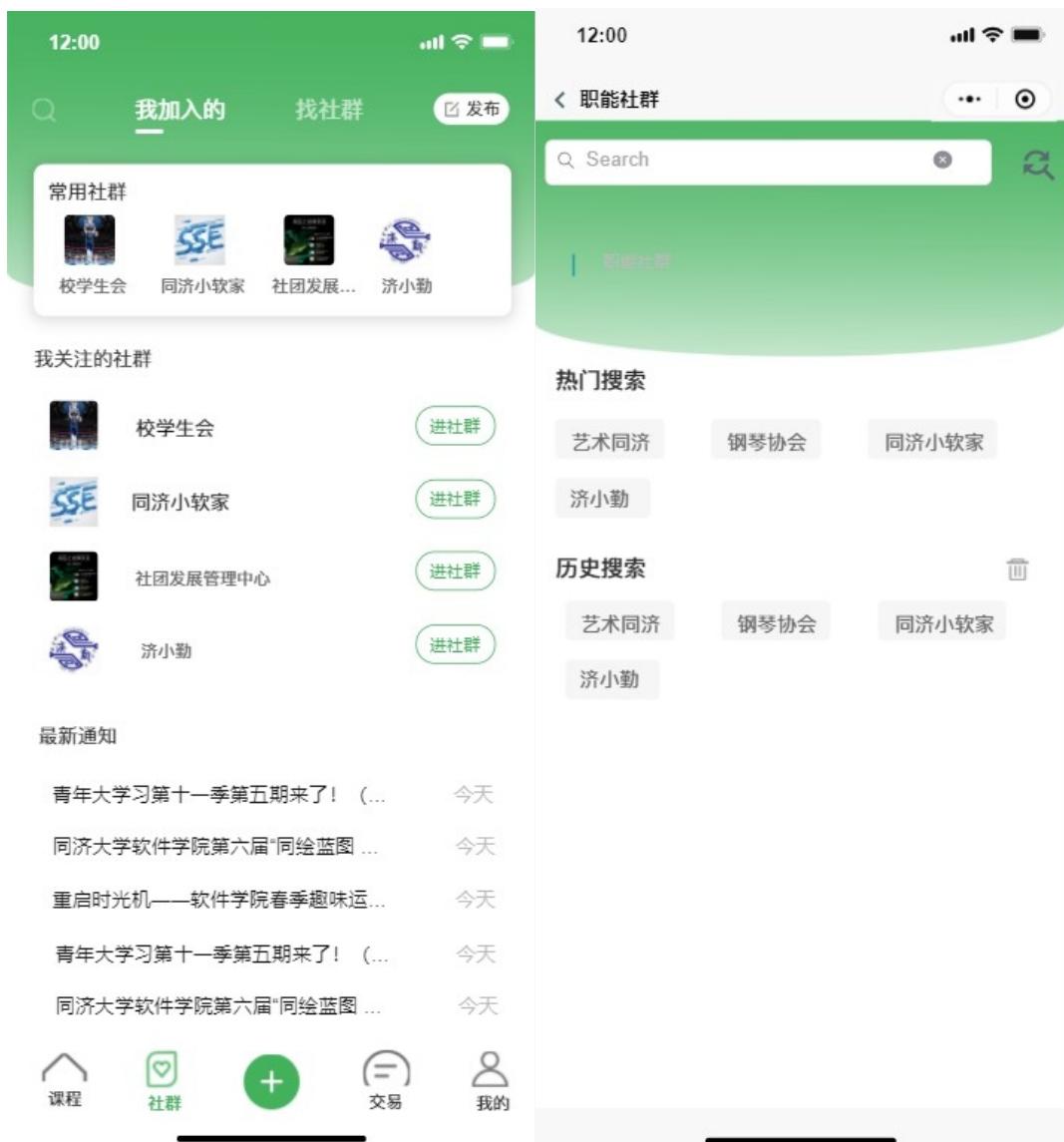


Then, the user can click the "Transaction" button at the bottom of the personal homepage to enter the **campus second-hand trading platform**. The homepage of the trading platform can intelligently generate recommended commodities based on the user's browsing history and purchase history, as well as provide a ranking of commodities with the most recent transactions. Users can use the precise search function and fuzzy search function to search for the items they want, and the search results are displayed in the default order. Users can filter products and change the sorting method according to their needs. After browsing to the favorite item, the user can click the "Buy" button to enter the order filling page and make payment.

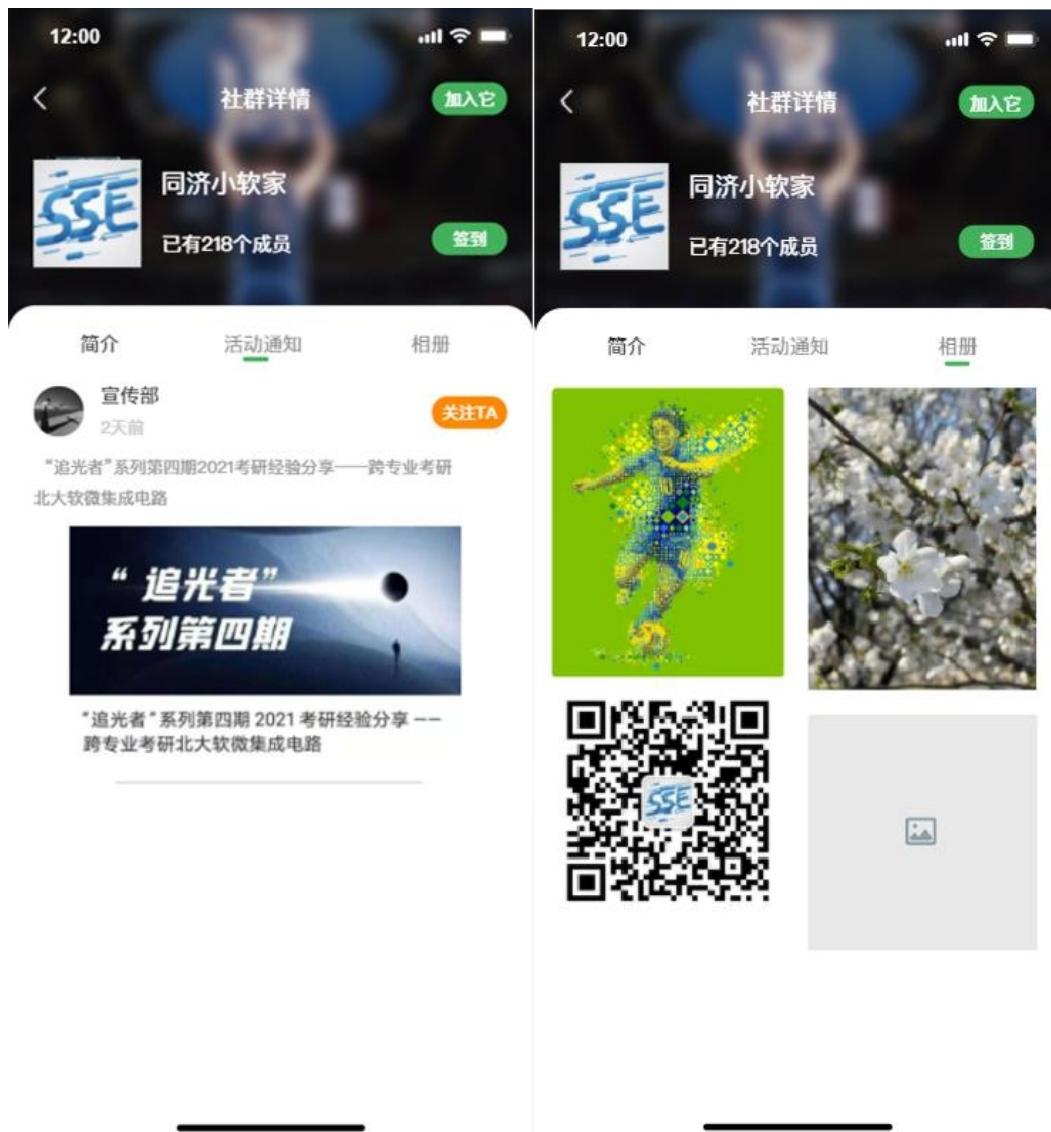


In addition, users can click the “+” in the middle to publish their products. During the second-hand transaction, the user can click on the “message” on the personal homepage to enter the real-time chat system. The real-time chat system includes transaction messages to communicate with other users; customer service messages to facilitate communication between users and customer service regarding after-sales service and refund related matters; order messages to check order status; platform messages to receive system notifications. The user clicks on any chat bar to enter the chat window, which displays the current transaction status, sent historical messages, and so on. Users can make audio and video calls with other users, or contact customer service on this page.

As for the Association system, the user clicks the "Community" tag in the bottom column to enter the community information interface. Click "I joined" in the top column to see the notification activities of the communities you joined, the communities you follow, and the latest community notifications. In "Communities I follow", click on the community to enter the community home page, where you can view the community profile, history notifications, photo albums, etc. In "Latest notifications", you can view notifications in chronological order. In the top bar, "Find a community" or click the search tab to find a community, so that we can enter the community search interface and enter the community name to search.



On the community homepage, you can click "join" in the upper right corner to follow the community and receive notifications; you can sign in, view the community profile, historical activity notifications and community photo album.



On the community homepage, you can click "join" in the upper right corner to follow the community and receive notifications; you can sign in, view the community profile, historical activity notifications and community photo album.

If you are an association administrator, you can also post notifications, edit your home page, etc., and share to WeChat and QQ, as well as sync with your community.

7. List of references

- [1] Dr. William S. Chao, *Software Requirements Specification (SRS) 2.0: The Structure-Behavior Coalescence Approach*. Create Space Independent Publishing, 2015.
SBC facilitates an integrated whole. Therefore, we conclude that software requirements specification (SRS) 2.0 using the SBC approach, which contains three fundamental diagrams: a) architecture hierarchy

diagram, b) component operation diagram, and c) interaction flow diagram, is highly adequate in specifying a software system.

- [2] Davis M A, *Just Enough Requirements Management: Where Software Development Meets Marketing*, New York, Dorset House Publishing, 2005.

This book is about how to discover, prune, and document requirements when people are subjected to tight schedule constraints.

- [3] Liu Bo, Chen Guanyi, Ma Yunlong. *Research on energy efficiency management system of smart hospital based on Cloud Service [J]*. Computer application and software, 2017 (5)

Based on cloud service and Internet of technology, this paper mainly studies the cloud service architecture for smart hospital energy efficiency management, as well as the key issues in management evaluation. And will effectively promote the development of green energy conservation in medical institutions.

- [4] Wang Shue. *Building information service management system based on hospital information workflow model [J]*. Fujian computer, 2017, 033 (009): 130-131

Introduce information automation management, change the traditional way of information service, improve the quality and efficiency of hospital information service management.

- [5] Ying Peng. *Overview of Object-Oriented System Development Method and UML Modeling Language[C]*. Institute of Management Science and Industrial Engineering. Proceedings of 2019 9th International Conference on Information and Social Science(ICISS 2019). 2019:38-44

This paper outlines the object-oriented methods and related technologies, and describes the object-oriented system development methods, including object-oriented analysis(OOA), object-oriented design.

8. Contribution of team members

Name	Contribution
1853572 Qiao Liang	<ul style="list-style-type: none"> ·Assignment of tasks ·UI design of login, homepage and trading system ·The empathy map canvas of agile requirements analysis ·Typography and introduction ·Use case modeling of 3.2.3 home page system and 3.2.4 transaction system
1854062 Zhibo Xu	<ul style="list-style-type: none"> ·Use case modeling of 3.2.1.Personal Information system, 3.2.2.Authority management system ·User journey map of agile analysis ·Glossary of terms
1854117 Zhenyu Li	<ul style="list-style-type: none"> ·Use case modeling of 3.2.5 Chatting and After-sale Service subsystem ·Draw and revise general use case diagram ·Supplementary specification
1953803 Yixi Zheng	<ul style="list-style-type: none"> ·Use case modeling of 3.2.6 evaluation system ·Draw the UI interface of the course evaluation system and the community information system ·Write a description of the corresponding UI interface
1954106 Wenchao Chen	<ul style="list-style-type: none"> ·Use case modeling of 3.2.7 the association management system ·Mapping the user story map of the main business of "@Ji" ·Write agile analysis document for user story map