

Report 3

Project name: Box

Group member: Klaus, Bruce, Anna,
Ivy, Olivia, Kathy

Content

1. Customer Statement of Requirement-----	P3-6
2. Glossary of Terms-----	P7
3. System Requirements-----	P8-9
4. Function Requirements Specification-----	P9-26
5. Effort Estimation using Use Case Points-----	P27-30
6. Domain Analysis-----	P31
7. Interaction Diagrams-----	P32-35
8. Class Diagram and Interface Specification-----	P36-40
9. System Architecture and System Design-----	P40-43
10. Algorithms and Data Structures-----	P43-44
11. User Interface Design and Implementation-----	P44-48
12. Design of Tests-----	P48-50
13. History of Work-----	P50-51
14. References-----	P51

1. Customer Statement of Requirements

I am a student of Sias. I met some problems in campus life that brought me great inconvenience. In order to better manage the campus express, the school had set up some express delivery points, so I had to pick up my expressage at the express delivery points. You know the campus of Sias is very big. The worst thing is that the delivery points in our school are very scattered, and each express delivery point has its own running time. Last semester, I bought a lot of books on the Internet. The seller sent me express from different courier companies, such as ZTO Express, YTO Express, SF Express and so on. I had to rush around at different express delivery points on campus to get my expressages. Especially in the peak period of online shopping, such as the Double 11 shopping carnival, every express point had a long queue. I even stood in line for 20 minutes at one delivery point last year. Sometimes it takes so long time to pick up the expressage that I'm late for class. Because of the conflict between my class time and business hours of express delivery points, I cannot get my express delivery in time. Such things always make me tired.

Because of these reasons, I very much hope that there is a 24-hour the self-pick-up cabinet under dormitory buildings. The courier will deliver it to the self-pick-up cabinet under each dormitory building according to the dormitory building number on the shipping address. All my expressage will be put in this cabinet. When my expressages is placed in self-pick-up cabinet, I will receive a message containing the pick-up password (which can also be viewed in small programs or applications). I can use the self-pick-up cabinet under the dormitory building at any time, and enter the

pick-up password to take away my expressages. I don't have to worry about waiting for a long time to pick up the express, losing the expressages or missing the business hours of the express point.

On the other hand, I have a very tight schedule this semester, 11:40 A.M. after class, and 1:00 P.M. will be class. This an hour is the peak time for dining in the canteen. In order to have lunch as soon as possible I often choose to order a good takeaway and let the deliver guy put the takeout downstairs so that when I get back to my dormitory, I can have lunch at once. That will give me more time to do some other things, instead of wasting my time waiting in line in the canteen. But these takeaways do not have a reasonable place to lay up them. I found that my takeaway was taken away by someone else when I returned to the dormitory after class. I cannot find my lunch so that I have to go to the afternoon class hungrily. This is not uncommon, I often hear complaints about this situation, but so far there is no good solution.

I hope the self-pick-up cabinet can also be used in takeaway storage, I can book a cupboard at the self-cabinet software in advance to store my takeaway and get a password for opening the self-pick-up cabinet, and then when I order the takeaway on line, just need to note my opening password to seller, so that when the deliver guy can put my takeaway into the self-pick-up cabinet. When I leave class at noon, I can go straight to the self-pick-up cabinet to get my takeaway and then end the rental of this self-pick-up cabinet. If this situation was true, I would no longer to worry about my takeaway being taken away by someone else.

In this software, I hope that its page design can be simple and clear, so that I can

know at a glance what it is expressing or what should I do. And the software is divided into three parts. The first part is to receive express or takeaway. First, courier or takeout personnel use the cabinet or application program (mobile phone number + password) to authenticate, then scan the express list to confirm my mobile phone number, and select the size of the cabinet (takeout can choose takeout cabinet with thermal insulation function), and finally the cabinet door opens and closes automatically to complete storage. I can input the pick-up code or two-dimensional code on the cabinet display screen. The system will automatically open the cabinet door after successful verification in the background, and then I can take my takeaway or expressages.

The second part is to send express, I can choose to send express in the cabinet or application program, enter my ID number + destination address information of express delivery, select the size of cabinet, and then cabinet automatically opens the cabinet door to complete storage. When my express is stored, the courier will get my express message and password. Then he used the courier code to open the cabinet door to collect the courier and sent the courier number to me by the software.

The third part is self-use, I can freely choose how long I wants to use, and it required advance reservation before each use. I can book the self-pick-up cabinet on the mobile phone application, and then choose the size and utility time of the self-pick-up cabinet I want to use. When making an appointment, it needs to input the user's cell phone number so that the application can send the cabinet password to me. During this period of time after the reservation successfully, I can through this

password to use self-pick-up cabinet at any time. We should pay special attention to that after exceeding the appointed time; the fee will be added in a certain proportion, so it require to pick up on time.



2. Glossary of Terms

Word	Explain
Quick landing	Users can enter their mobile phone number, authentication code and choose their identity to login directly without registering any more.
Account password login	Users login with registered mobile phone number and password
Register	User's Behavior of Obtaining Accounts
My	User center
Cabinet situation	Is the cabinet empty?
Time	Use time of cabinet
Express	What others have sent you use the cabinet
Confirm	Confirm the current page information
Verification	Open the corresponding cabinet by two-dimensional code or verification code
News	Use Information and Activity Notification of Cabinets
Cancel	Cancel the selected option
Button	Verification button for accessing things on the cabinet
Position	Location of the cabinet being used by the user

3. System Requirement

a. Enumerated Functional Requirements

REQ' T	Priority weight	Customer requirement description	Function realization
REQ-4	2	Users need to log out.	Logout
REQ-5	5	Users need a way to store packages.	Sending two-dimensional code
REQ-6	3	Users need to choose whether to use the incubator or not.	Fill Order
REQ-7	3	Users need to fill in recipient information.	Fill Order
REQ-8	5	Users need to open the cabinet to store packages.	Send code
REQ-9	5	Users need a way to pick up packages.	Scan code
REQ-12	8	The system requires data control center to record data.	Data processing

b. Enumerated Nonfunctional Requirements

We all know that the enumerated nonfunctional requirement contain FURPS. The F means function, the U means usability, the R means reliability, the P means performance and the S means supportability. First, I want to explain the Function, we just want to make an APP to help students and teachers to get the takeout, because some of the students and teachers would lose their takeout, and make their angry, causing the bad mood of having lunch. About the security, the system just sends the private messages to the users; other users cannot see the message. We ensure our system is security. We will build up the database to store our information about the students and teachers. Only users who need self-container can see only scream privately. Second, I want to explain the Usability. It is easy for students and teachers to use the APP, because we just receive the message and go to the self-container to

take the takeout and close the door of the self-container. We will buy some beautiful self-container to let user feel comfortable. When we finish the transaction, our system will record the information. We can record the time, times, and the honest and break. Third, I want to explain the Reliability. When the system is broken, we can fix it in the computer. The courier who put the takeout to the self-container, the message can send the user's phone, it can ensure the accuracy. Forth, I want to explain the Performance, our system can distinguish whether the self-container is empty or is full, also, our system can distinguish whether the self-container's door is open or is close, and our system can distinguish whether the message send is successful or is fail. When the courier put the takeout in the self-container, the message will send the users' phone as soon as possible, also, the courier can use the APP to check which self-container is empty or is full as soon as possible. Fifth, I want to explain the Supportability. We can test the APP for one week, and we will make a report to send the users' phone to know the suggestion and bug. We would create the Android and IOS, so we can ensure the compatibility. About the configurability, the intelligence phone can suppose the APP. We can upload the erection sequence to the Internet, so all of the users can find the APP's erection sequence. About the scalability, we can add the advertisement to help the business, and we can charge extra fees.

C. User Interface Requirements

For this practical project, user interface requirements are not are not complex. Basically, they can be divided into function keys and backgrounds. Function keys have higher priority than background. The background is basically a solid color background with some text guide. Function keys are displayed as text icons on the screen. Different icons correspond to different functions. There are nearly 30 icons. They are too many to list one by one. Here are two icons:



These two icons are displayed as two keys on the main interface, to lead users to select the two core functions of the system, receiving or delivering. All the keys are basically like these. Simple text description can make the function of keys clear at a glance.

4. Functional Requirements Specification

a. Stakeholders

Users: Students, teachers, courier and takeaway

Managers: "Campus Cabinet" Program Administrators

b. Actors and Goals

Actor	Actor's goal'	User case name
User	To quit using the software	Logout(UC-4)
User	Processing all data received	Data processing(UC-5)
User	To fill in information about using cabinets	Fill Order(UC-6)
User	To select the cabinet to use and obtain the storage code of the corresponding cabinet.	Send store code(UC-7)
User	Scanner Scanning storage Code on the Cabinet	Scan store code(UC-8)
User	To Get Pickup code for Use Cabinets	Get the Pickup code (UC-12)
Cabinet	To select the cabinet to use and obtain the storage code of the corresponding cabinet.	Send store code(UC-7)
Cabinet	The cabinet opens the corresponding cabinet door automatically	Open cabinet(UC-9)
Cabinet	Scanner Scanning storage Code on the Cabinet	Scan store code(UC-8)
Store code	To select the cabinet to use and obtain the storage code of the corresponding cabinet.	Send store code(UC-7)
Store code	Scanner Scanning storage Code on the Cabinet	Scan store code(UC-8)
Pickup code	To Get Pickup code for Use Cabinets	Get the Pickup code (UC-12)
Scanner	Scanner Scanning storage Code on the Cabinet	Scan store code(UC-8)
Timer	To counts how often cabinets are used	Open cabinet(UC-9)
Memorizer	To store all information	UC-(1~16)

Initiating actor: User

Participating actor: Cabinets、Store code、Scanner、Timer、Pickup code、Order、Software

c. Use Cases

i: Casual Description:

Login: The user logs in to the program by using the registered account password.

Exit: Users quits the function like delivery or user.

Data processing: Responsible for the storage and processing of all system information.

Fill Order: The sender fills in the order information (recipient's name, phone number, "box" ID), and choose whether you need an incubator.

Send store code: The system receives the order information, and sends the QR code back.

Scan store code: The cabinet scans the QR code.

Open cabinet: The cabinet recognizes the QR code successfully and opens the cabinet.

Close cabinet: The depositor puts the package and closes the door

Cabinet occupancy mark: The data processing center marks that the cabinet has been used.

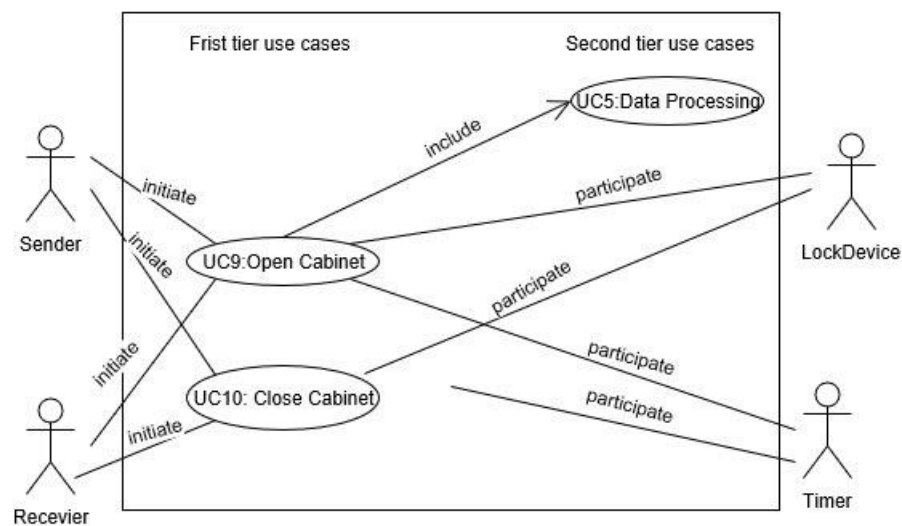
Send pick-up code: The system receives the order information, and sends the pickup QR code to the id according to the order information.

Scan pick-up code: The cabinet scans the QR code.

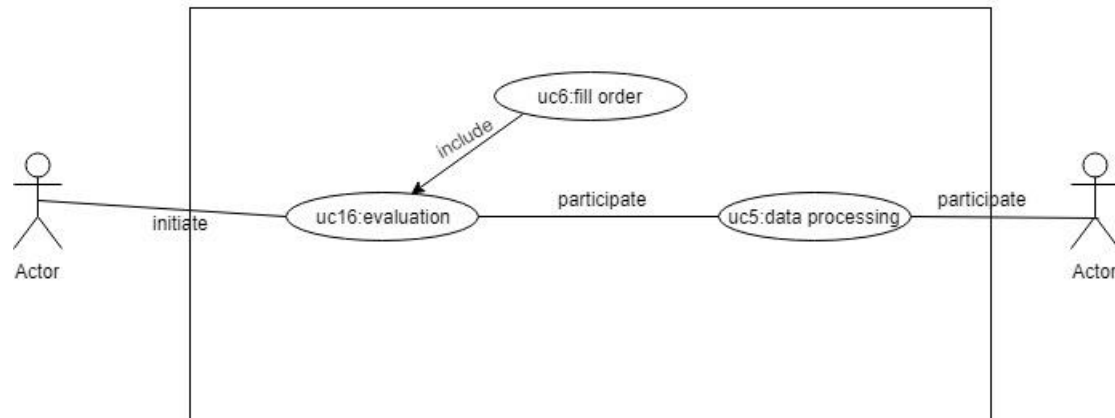
Cancel occupancy mark: After taking out the courier, the data control center cancels the occupancy mark.

ii: Use Case Diagram:

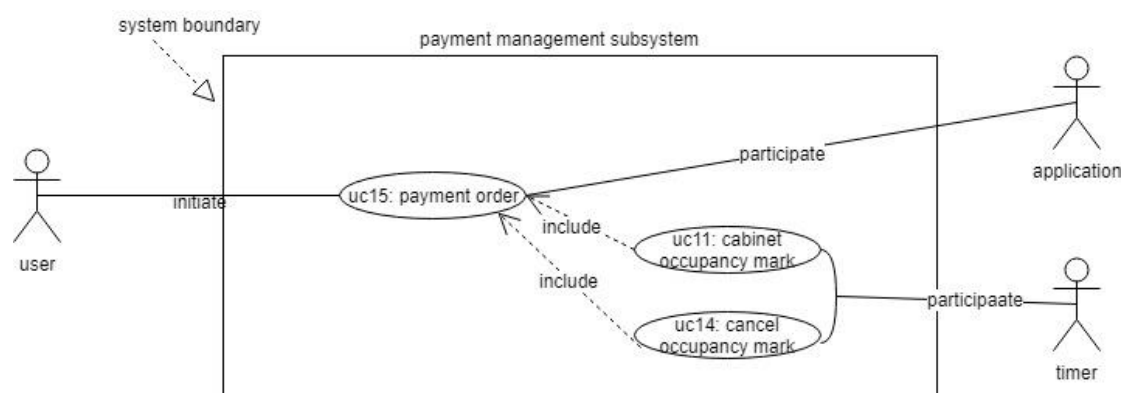
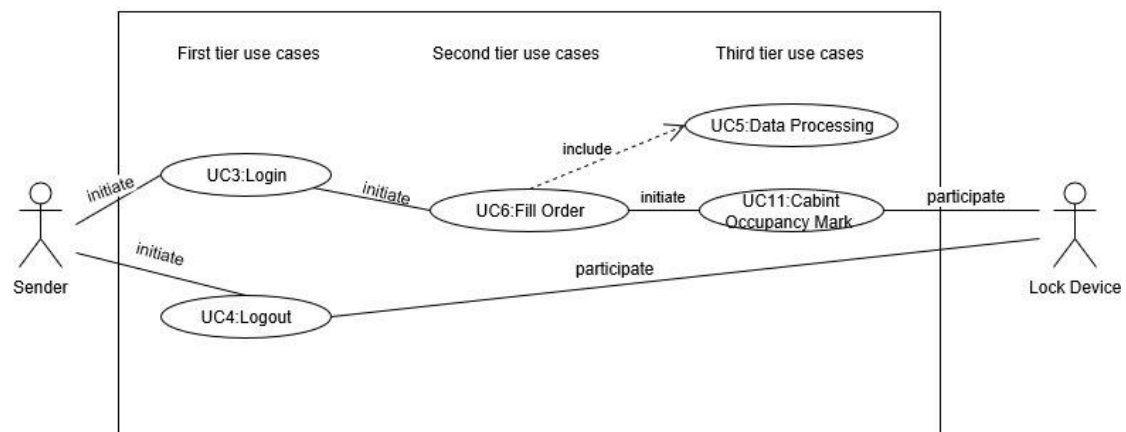
Use Case Diagram: Cabinet Control



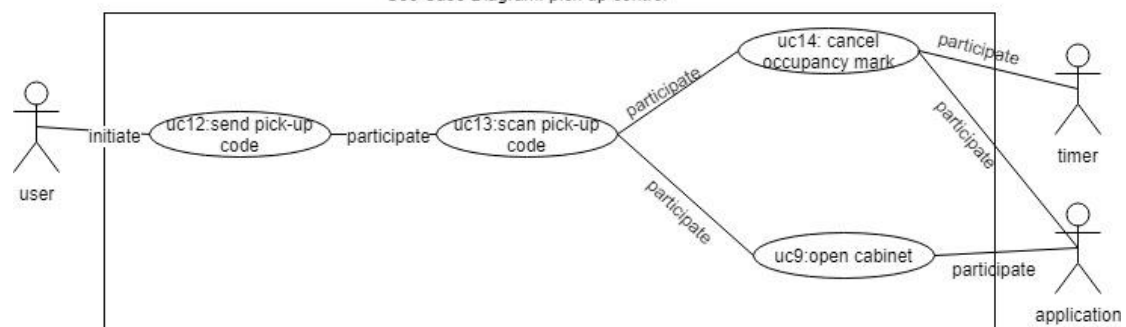
Use Case Diagram: evaluation control



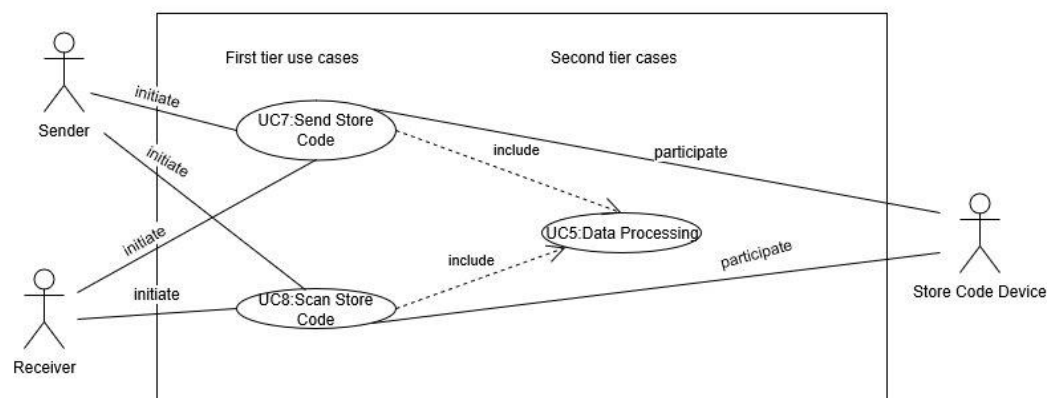
use case diagram:Order

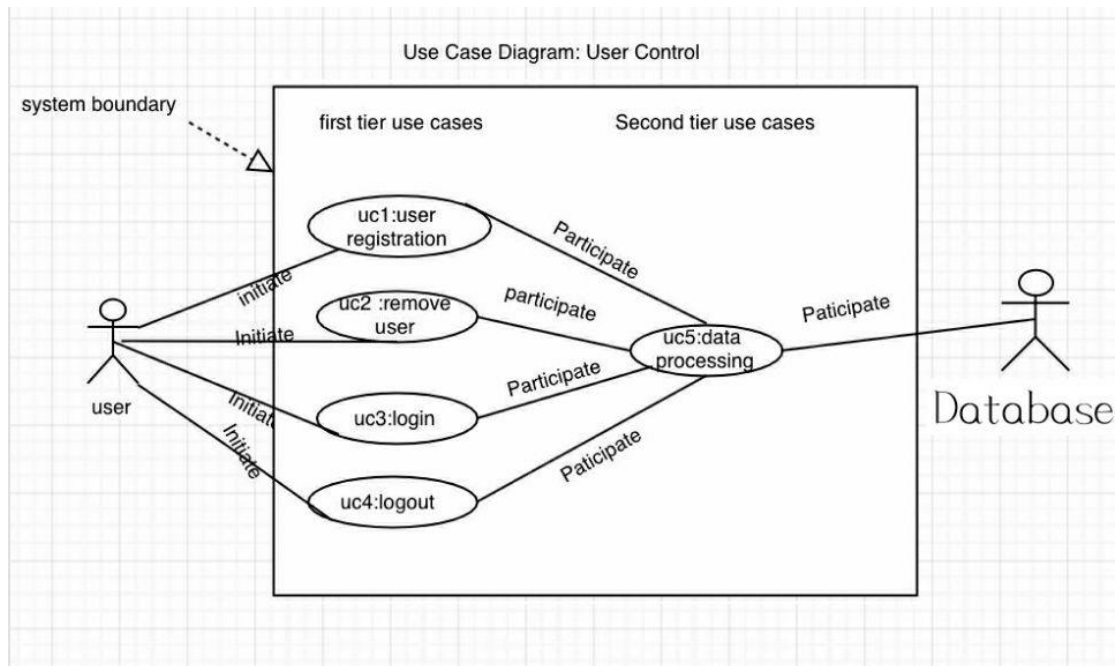


Use Case Diagram: pick-up control



Use Case Diagram: Store Code





iii: Traceability Matrix:

Traceability Matrix:

table1-2

Req't	PW	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14
req-4	2	x	x									
req-5	5		x	x	x	x	x					
req-6	3		x	x					x			
req-7	3		x	x					x			
req-8	5		x		x	x	x	x				
req-9	5		x					x		x	x	x
req-12	8		x									
Max PW	8	2	8	5	5	5	5	5	3	5	5	5
Max PW	43	2	43	11	10	10	10	10	6	5	5	5

iv: Fully-Dressed Description:

UC-3: Login: Users log in with registered account passwords.

UC-3: Login: Users log in with registered account passwords.
Related Requirement's: REQ1-REQ12.table1-2
Initiating Actor: Any of: senders, recipients
Actor's Goal: achieve the process of the log in, log out and registration.
Participating Actors: data process control, system, information system
Preconditions: none
Post conditions: achieve the process of what the users want to get.
Flow of Events for Main Success Scenario: ---->a. Users enter the login page and enter their account number and password for login. <---b. The system receives the order and feedback to the users.

UC-4: Exit

Use Case UC-4: Exit
Related Requirements: REQ-4-user need to log out Initiating Actor: senders and receivers Actor's goal: exit page of short time storage Participating Actors: data processor Preconditions: the system is working and the user completed the process of saving package in the cabinet Post conditions: none Flow of Events for Main Success Scenario: ----->a. User clicks the "MINE" button in the main interface. <-----b. System displays the personal center interface to the user. ----->c. User clicks the "exit" button. <-----d. System prompted the user to ensure to exit. ----->e. User clicks "Yes" to exit. <-----f. System stores the operate logs to database and shutdown.

UC-5: data processing

UC-5: data processing
<p>Related Requirement's: REQ1, REQ12, table1-2</p> <p>Initiating Actor: Any of: senders, receivers</p> <p>Actor's Goal: Let the database system to store the database of the users.</p> <p>Participating Actors: name's students or teacher, address of the teacher and students, name's food and takeout, phone number, senders' information</p> <p>Preconditions: The senders and receivers finished their order</p> <p>Post conditions: The data processing center store and arrange the database</p> <p>Flow of Events for Main Success Scenario:</p> <p>----→a. The receivers finished the information,</p> <p>←----b. The system receives this order, and request users to store the database</p> <p>----→c. The users choose "yes"</p> <p>←----d. The system stores the database and feedback to the receivers.</p>

Use Case UC- (6) : Users fill in order information

Use Case UC- (6) : Users fill in order information
<p>Related Requirement's: REQ5, REQ5, REQ7, table1-2</p> <p>Initiating Actor: Any of: senders</p> <p>Actor's Goal: Ensuring accuracy and consistency of information</p> <p>Participating Actors: name's students or teacher, address of the teacher and students, name's food and takeout, phone number, order information, data processing center.</p> <p>Preconditions: Senders confirm the type of food or takeaway</p> <p>Post conditions: Senders confirm order information again,</p> <p>Flow of Events for Main Success Scenario:</p> <p>---→ a. Senders determine the type of food and take-out,</p> <p>←---b. the system displays order information to users,</p> <p>---→c. users fill in order information.</p> <p>←---d. The system feedback to the users.</p>

Use Case UC- (7) Send storage code

Use Case UC- (7) Send storage code
<p>Related Requirement's: REQ5, REQ8, table1-2</p> <p>Initiating Actor: All of: senders</p> <p>Actor's Goal: let the sender to take the food or takeout into the cabinet.</p> <p>Participating Actors: data processing center, information system</p> <p>Preconditions: The receivers have finished the order information.</p>

Post conditions: The system get the behavior and send the storage code, and let the senders take the food of takeout into the cabinet.

Flow of Events for Main Success Scenario:

---→a. The receivers wrote the order information,

←---b. the system understood the meaning and send storage code to the senders

---→c. the senders will use the storage code to open the cabinet's door.

←---d. The door opened.

Use Case UC- (8) scan the storage code

Use Case UC- (8) scan the storage code

Related Requirement's: REQ5, REQ8, table1-2

Initiating Actor: All of: senders

Actor's Goal: the senders want to take the food or takeout into the cabinet.

Participating Actors: data processing center, canner, information system,

Preconditions: The senders get the storage code

Post conditions: The cabinet's door is opened.

Flow of Events for Main Success Scenario:

--→a. The senders get the code; he goes to the cabinet and scans the storage code.

←--b. The system confirms the behavior, and opens the door.

Use Case UC- (9) Open your own cabinet

Use Case UC- (9) Open your own cabinet

Related Requirement's:REQ5, REQ8, table1-2

Initiating Actor: Any of: recipients

Actor's Goal: Get the own food and takeout

Participating Actors: cabinet, door of the, scanner, system, data Processing Center

Preconditions: Recipients got the fetch code and scan the QR code.

Post conditions: The door opened.

Flow of Events for Main Success Scenario:

--→a. The recipients got the message and use the fetch code to scan the QR code.

←---b. The system will confirm the behavior. The recipients open the door.

Use Case UC- (10) Close own cabinet

Use Case UC- (10) Close own cabinet
<p>Related Requirement's: REQ8, REQ9, table1-2</p> <p>Initiating Actor: Any of: recipients</p> <p>Actor's Goal: close the own cabinet</p> <p>Participating Actors: cabinet, door of the cabinet, data processing center</p> <p>Preconditions: Recipients got the food or takeout successfully.</p> <p>Post conditions: The door is closed.</p> <p>Flow of Events for Main Success Scenario:</p> <p>--→a. The recipients got the food and takeout.</p> <p>←--b. The system confirmed the behavior. The recipients need to close the door.</p>

Use Case UC- (11) Tag occupation of self-contained cabinets

Use Case UC- (11) Tag occupation of self-contained cabinets
<p>Related Requirement's: REQ6, REQ7, table1-2</p> <p>Initiating Actor: Any of: senders, recipients, system</p> <p>Actor's Goal: the senders and recipients finished the behavior.</p> <p>Participating Actors: data processing center, cabinets</p> <p>Preconditions: The senders finished the behavior and close the door.</p> <p>Post conditions: The cabinets tag the occupation.</p> <p>Flow of Events for Main Success Scenario:</p> <p>--→a. The senders and recipients finished the process and close the door.</p> <p>←--b. The system confirmed the behavior and tag occupation of the cabinets</p>

Use Case UC- (12) System sending the fetch code

Use Case UC- (12) System sending the fetch code
<p>Related Requirement's: REQ9, table1-2</p> <p>Initiating Actor: System, fetching code</p> <p>Actor's Goal: Let users see their food or take-out information</p> <p>Participating Actors: recipients, system, data processing center</p> <p>Preconditions: System has received the successful order information, and sent the fetch code.</p> <p>Post conditions: Recipients received the fetch code.</p> <p>Flow of Events for Main Success Scenario:</p> <p>--→a. System saw the information and sent the fetch code</p> <p>←--b. the recipients got the fetch code.</p>

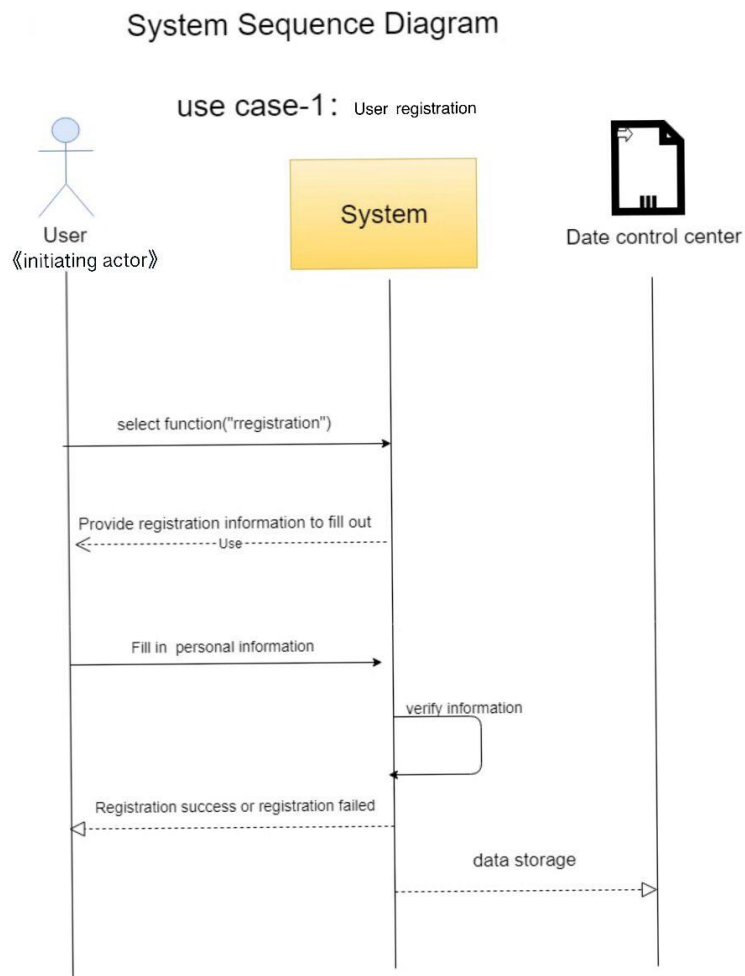
Use Case UC- (13) User scans the code

Use Case UC- (13) User scans the code
<p>Related Requirement's: REQ9, table1-2</p> <p>Initiating Actor: Any of: receivers</p> <p>Actor's Goal: Receivers get the food and takeout.</p> <p>Participating Actors: data processing center, system</p> <p>Preconditions: System got the access information about information code, and sent this code to the users</p> <p>Post conditions: Users scan the door, and the door will open</p> <p>Flow of Events for Main Success Scenario:</p> <p>---→a. The system got the information code and sent the access code to the recipients.</p> <p>←---b. The recipients got the access code.</p>

Use Case UC- (14) Cancel occupancy mark

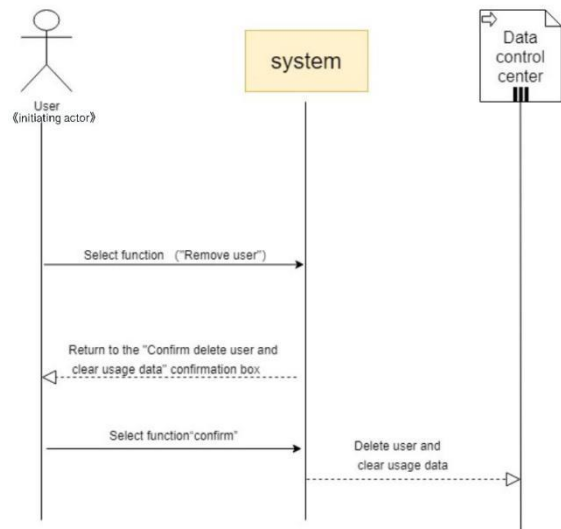
Use Case UC- (14) Cancel occupancy mark
<p>Related Requirements: REQ-6, REQ-9, REQ-12</p> <p>Initiating Actor: Any of :User, Software,</p> <p>Actor's Goal: Restoration of cabinet usability</p> <p>Participating Actors: LockDevice, Timer, Memorizer,</p> <p>Preconditions: The LockDevice and code verification device of the cabinet are natural.</p> <p>User has been sent the pick-up code.</p> <p>Flow of Events for Main Success Scenario:</p> <p>include::Send pick-up code(UC-12), Scan pick-up code(UC-13), Cancel occupancy mark(UC-14)</p> <p>----->a: User opens the application and selects the menu item"close the sale".</p> <p><-----b: system provide the pick-up code to user.</p> <p>c:the system access database to verify the account information-----></p> <p><-----d: system cancel occupancy mark successful or not</p> <p>e:system store the operation logs to database-----></p>

d. System Sequence Diagrams:



System sequence diagram

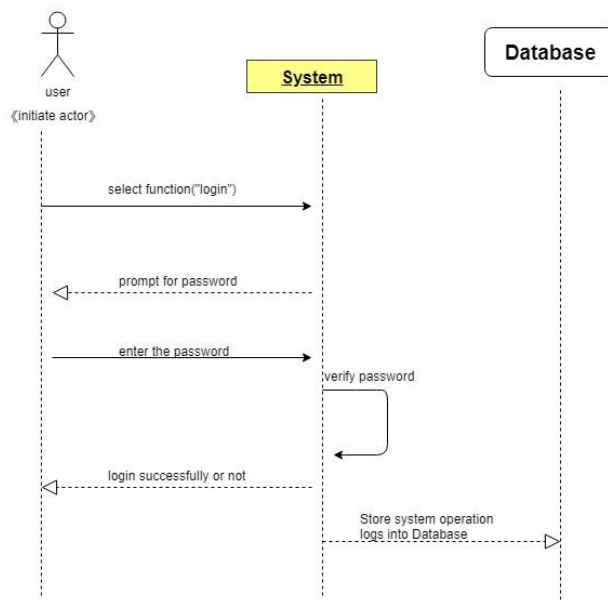
Use case-2: Remove user



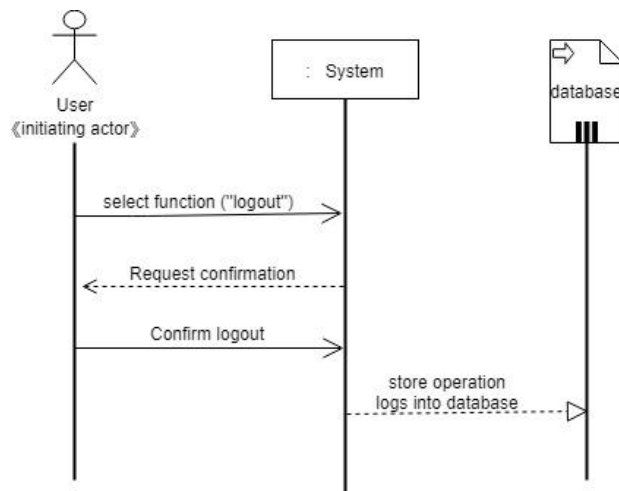
uc-3.

System Sequence Diagram

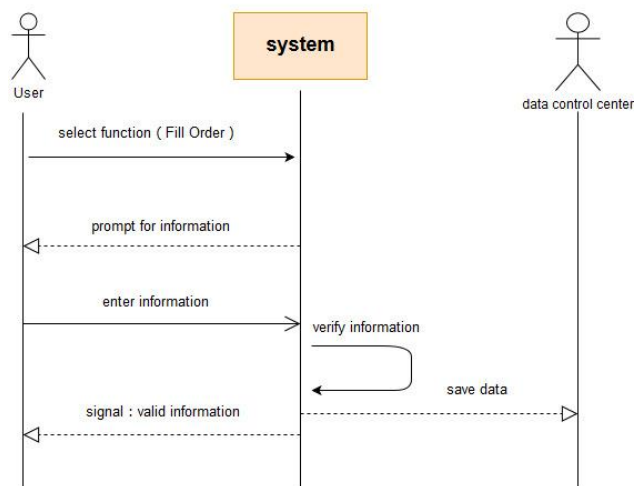
Use case: Login



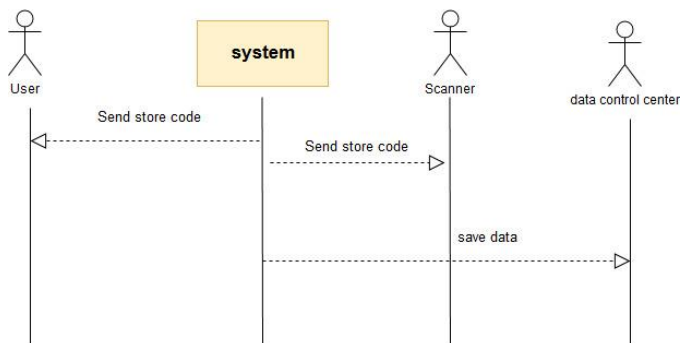
UC-4



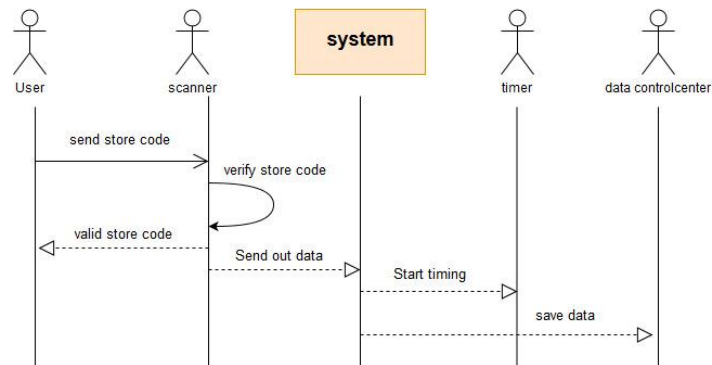
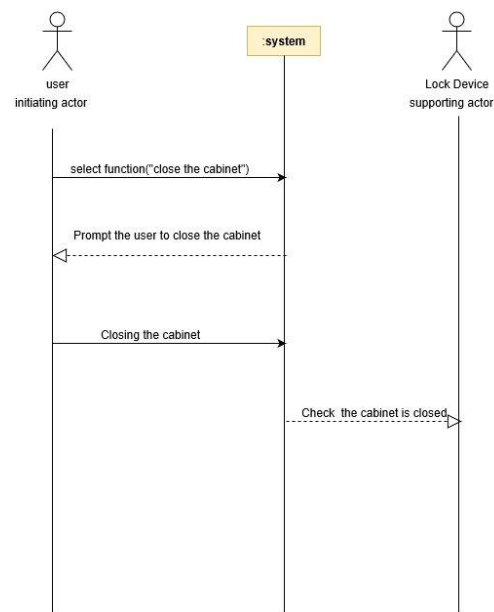
uc-6

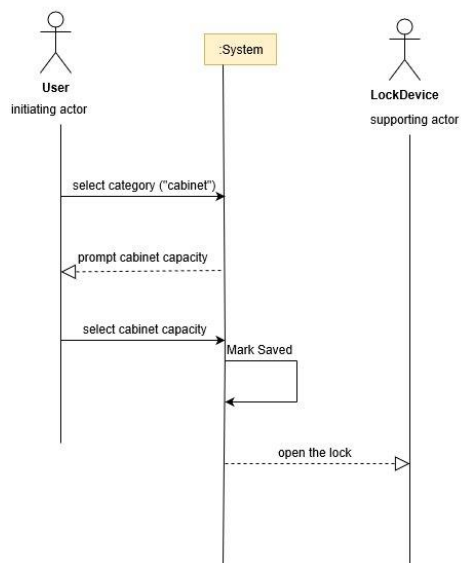
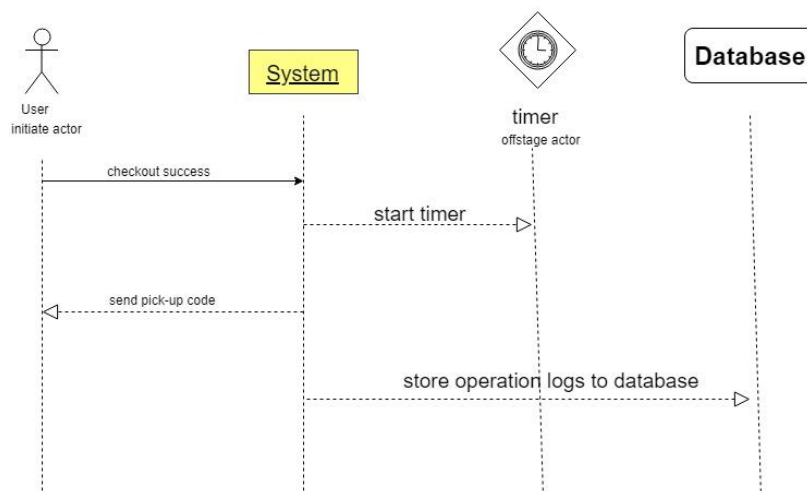
Fill Order

uc-7

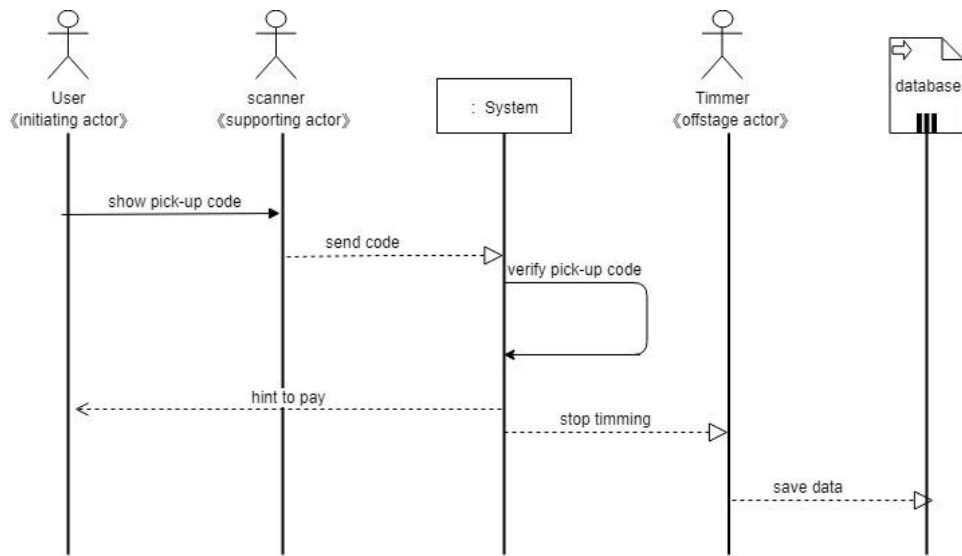
send store code

uc-8

scan store code**Use Case 10 :close cabinet**

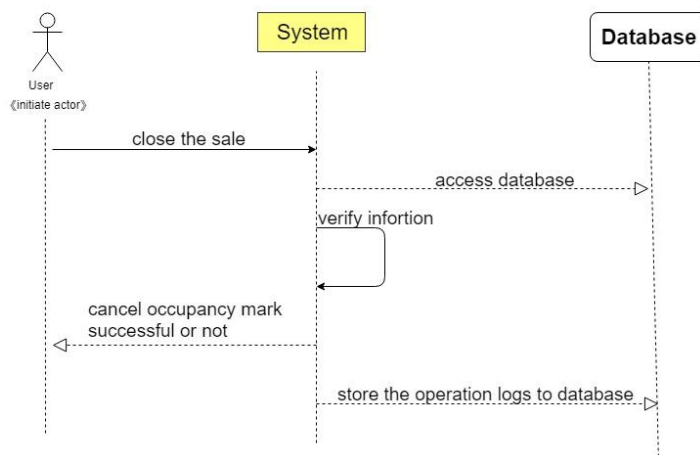
Use Case11:Cabinet occupancy mark**System Sequence Diagram****Use Case12: send pick-up code**

UC-13



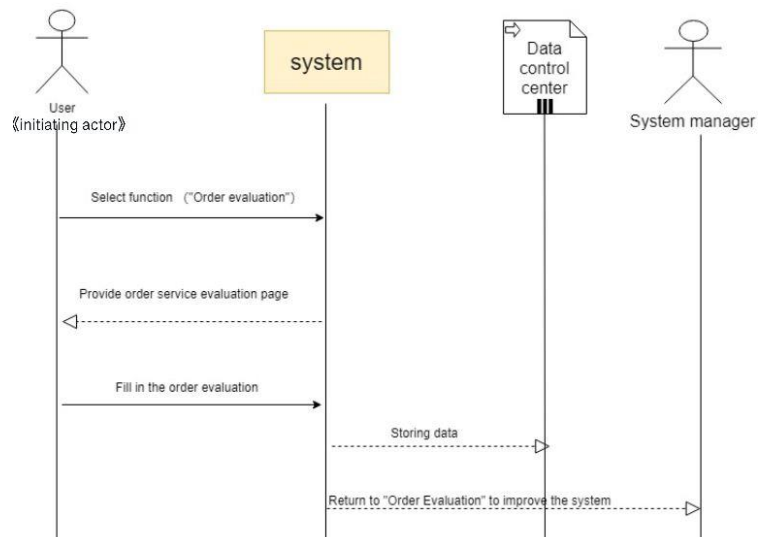
System Sequence Diagram

Use Case14:Cancel occupancy mark



System sequence diagram

Use case-15: Payment order



5. Effort Estimation using Use Case Points

Technical factor	Description	Weight	Perceived Complexity	Calculated Factor (Weight * Perceived Complexity) 计算因子 (权重*感知复杂度)
T1	Distributed system	2	3	$2*3 = 6$
T2	good performance in running	1	3	$1*3 = 3$
T3	End-user expects efficiency but there are no exceptional demands	1	3	$1*3 = 3$
T4	Internal processing is simple	1	1	$1*1 = 1$
T5	Reusable design	1	1	$1*1=1$
T6	It will probably be installed by technician	0.5	0	$0.5*0=0$
T7	Ease of use is very important	0.5	5	$0.5*5 = 2.5$
T8	No portability	2	0	$2*0=0$
T9	Easy to change	1	1	$1*1 = 1$
T10	Concurrent use	1	4	$1*4 = 4$
T11	Security is a significant concern	1	5	$1*5 = 5$
T12	Have direct access for third parties	1	2	$1*2=2$
T13	No unique training needs	1	0	$1*0 = 0$
Technical Factor Total:				28.5

TCF

$TCF = \text{Constant-1} + \text{Constant-2} * \text{Technical Factor Total} = 0.6 + 0.01 * 28.5 = 0.885$

Constant-1 (C1) = 0.6

Constant-2 (C2) = 0.01

ECF

Environmental factor	Description	Weight	Perceived Impact	Calculated Factor (Weight Perceived Impact)
E1	Beginner familiarity with the UML-based development	1.5	1	$1.5 * 1 = 1.5$
E2	Some familiarity with application problem	0.5	2	$0.5 * 2 = 1$
E3	Some knowledge of object-oriented approach	1	2	$1 * 2 = 2$
E4	Beginner lead analyst	0.5	1	$0.5 * 1 = 0.5$
E5	Highly motivated	1	6	$1 * 6 = 6$
E6	Stable requirements expected	2	4	$2 * 4 = 8$
E7	No part-time staff will be involved	-1	0	$-1 * 0 = 0$
E8	Programming language of average difficulty will be used 将使用平均难度的编程语言	-1	3	$-1 * 3 = -3$
Environmental Factor Total:				16

$$ECF = \text{Constant-1} + \text{Constant-2} * \text{Environmental Factor Total} = 1.4 - 0.03 * 16 = 0.92$$

$$\text{Constant-1 (C1)} = 1.4$$

$$\text{Constant-2 (C2)} = -0.03$$

UAW

Actor name	Description of relevant characteristics	Complexity	Weight
user	user is interacting with the system through the user interface	Complex	3
cabinet	The box interacts with the system and is controlled by the system.	Simple	1
LockDevice	LockDevice is another system which interacts with our system through a defined API.	Simple	1
scanner	Same as LockDevice.	Simple	1
Database	Database is another system interacting through a protocol.	Average	2
Timer	Same as LockDevice.	Simple	1
Data control center	another system interacting through a protocol	Average	2

$$\text{UAW(home access)} = 4 * \text{Simple} + 2 * \text{Average} + 1 * \text{Complex} = 4 * 1 + 2 * 2 + 1 * 3 = 11$$

UUAW

Use case	Description	Category	Weight
exit (UC-4)	Simple user interface. 2 steps for the main success scenario. No participating actors.	Simple	5
Data processing(UC-5)	Complex user interface. More than 7 steps for the main success scenario	Complex	15
Fill Order (UC-6)	Simple user interface. 3+3=6 steps for all scenarios. 2 participating actors (user, data control center).	Average	10
Send store code(UC-7)	Simple user interface 3 steps for all scenarios. 3 participating actors (user, scanner, data control center).	Average	10
Scan store code(UC-8)	Complex user interface. 6 steps for the main success scenario). 4 participating actors (user, scanner, data control center, timer).	Average	10
Open cabinet (UC-9)	Simple user interface 4 steps for all scenarios. 2 participating actors (user, LockDevice).	Average	10
Close cabinet(UC-10)	Same as open cabinet	Average	10
Cabinet occupancy mark(UC-11)	Simple user interface 5 steps for all scenarios. 3 participating actors (user, LockDevice, data control center).	Average	10
Send pick-up code(UC-12)	Simple user interface 4 steps for all scenarios. 3 participating actors (user, timer , database).	Average	10
Scan pick-up code(UC-13)	Complex user interface. 6 steps for the main success scenario). 4 participating actors (user, scanner, database, timer).	Average	10
Cancel occupancy mark(UC-14)	Simple user interface 5 steps for all scenarios. 2 participating actors (user, database).	Average	10

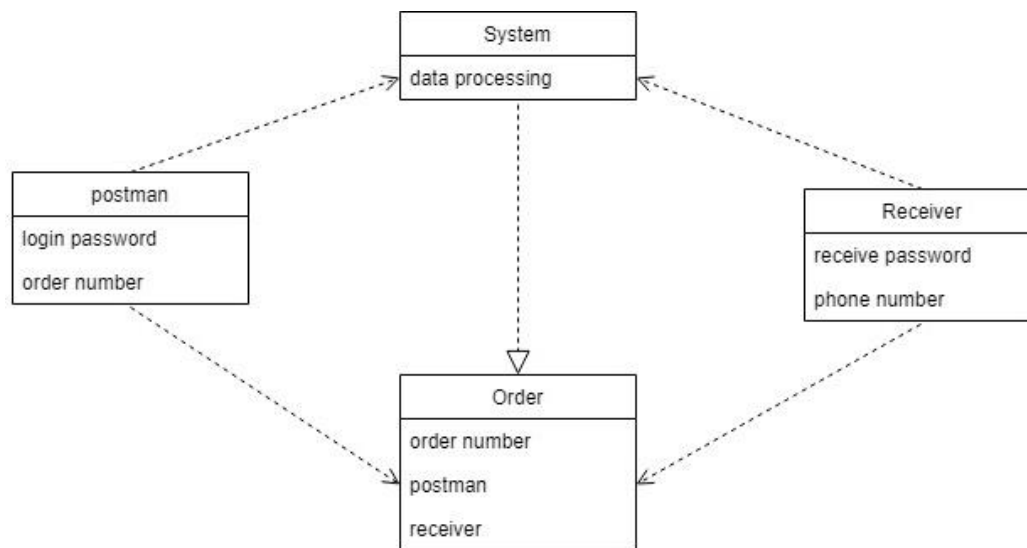
$UUCW(\text{home access}) = 1 * \text{Simple} + 9 * \text{Average} + 1 * \text{Complex} = 1 * 5 + 9 * 10 + 1 * 15 = 110$

$UUCP = UAW + UUCW = 11 + 110 = 121$

$TCF = 0.885$ $ECF = 0.92$

$UCP = UUCP * TCF * ECF = 121 * 0.885 * 0.92 = 98.5$

6. Domain Analysis



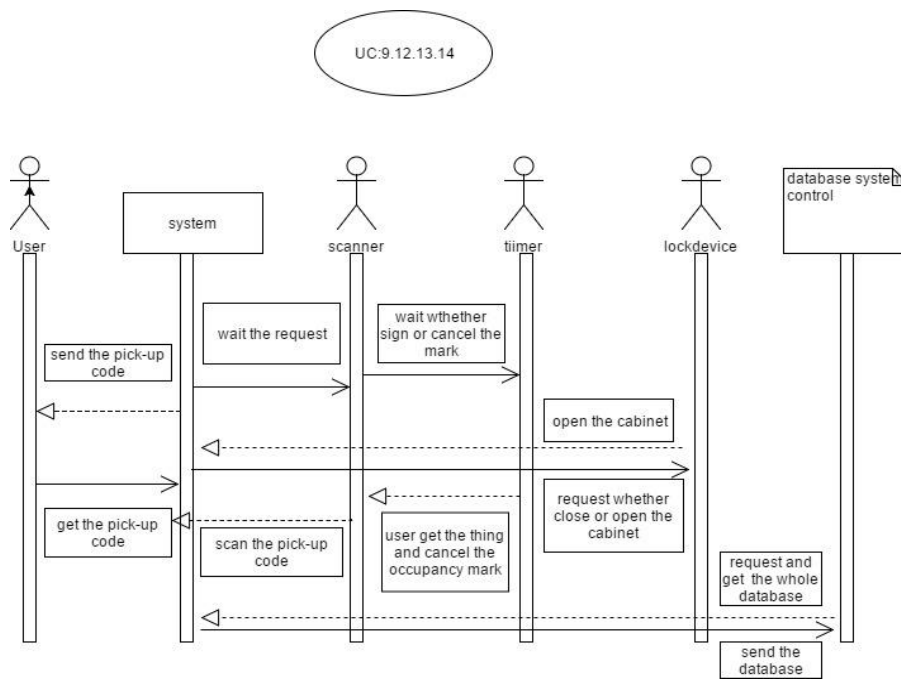
Association definitions

The order can be queried by the senders and the receivers, so the relationship between them is that the order depends on the senders and the receivers.

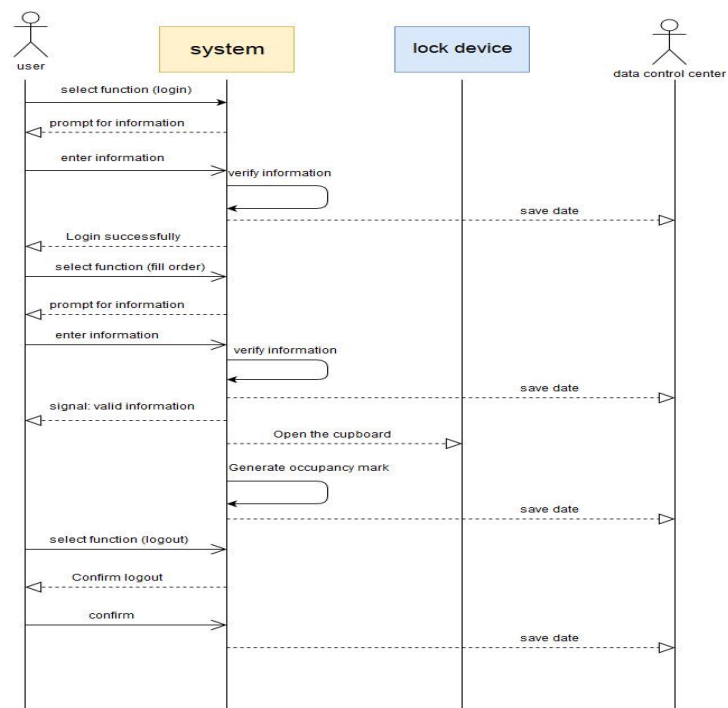
Attribute definitions

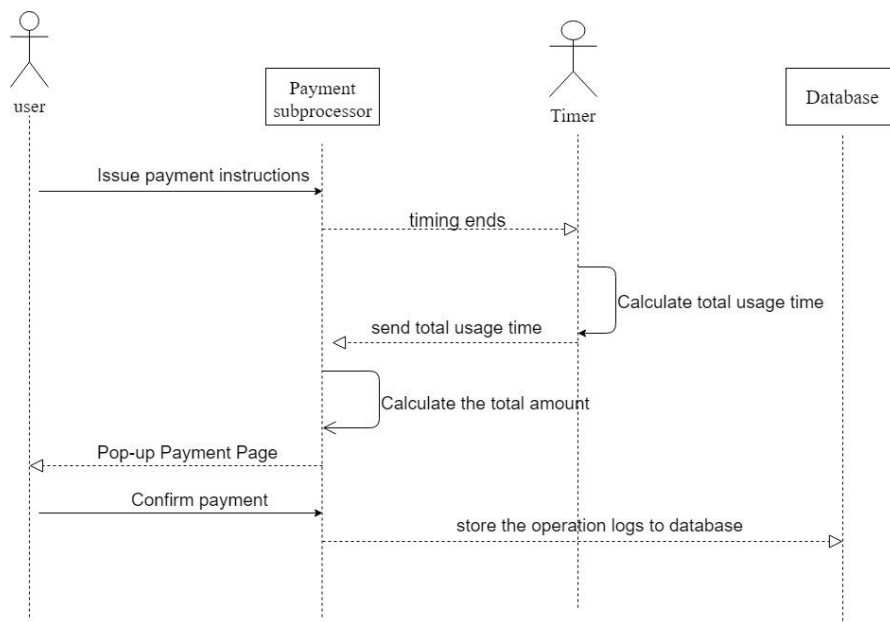
Order	order number, postman, receiver
Postman	login password, order number
Receiver	receive password, phone number
System	data processing

7. Interaction Diagrams

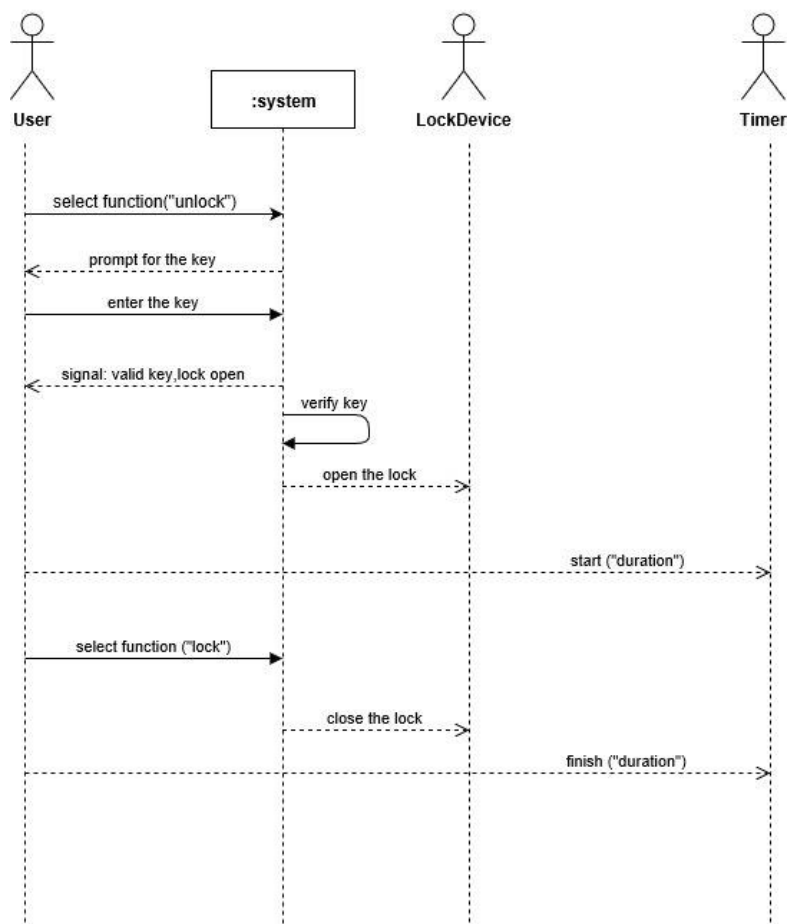


order

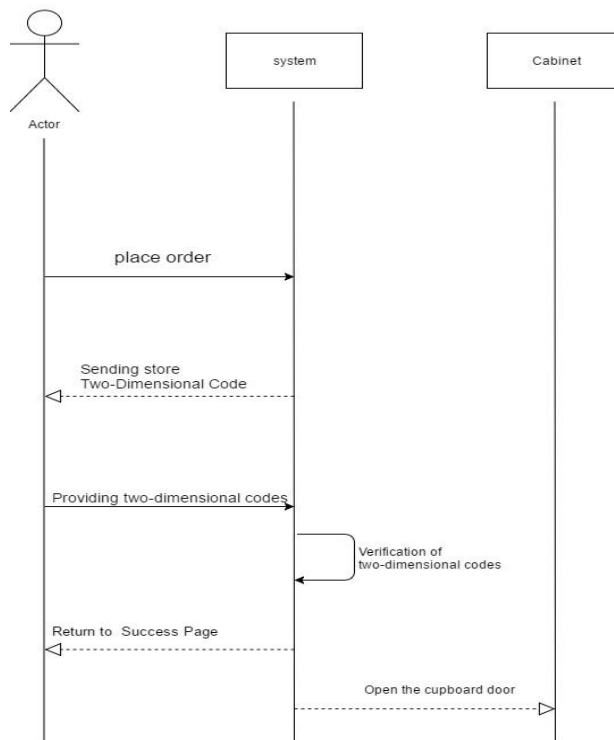




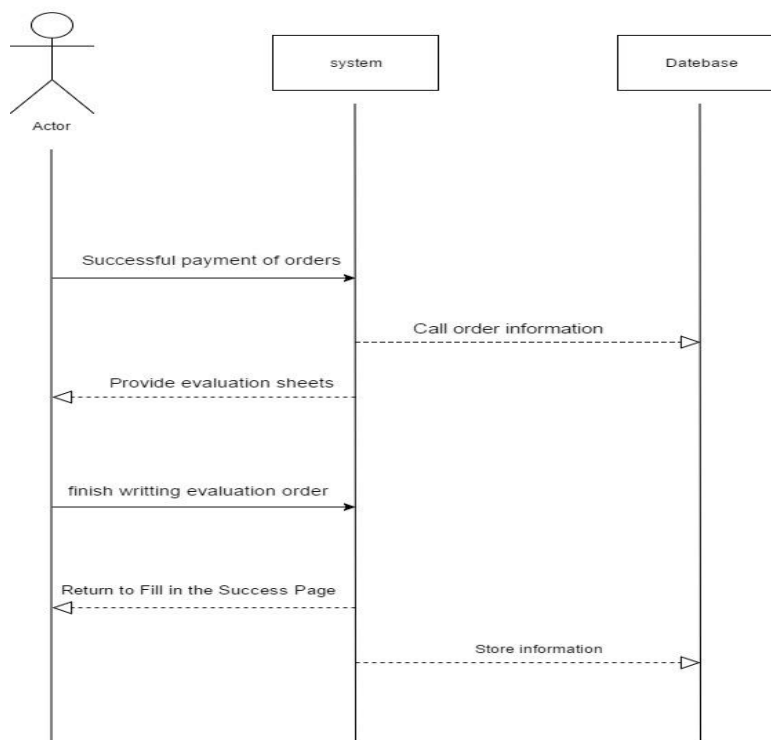
Cabinet Control



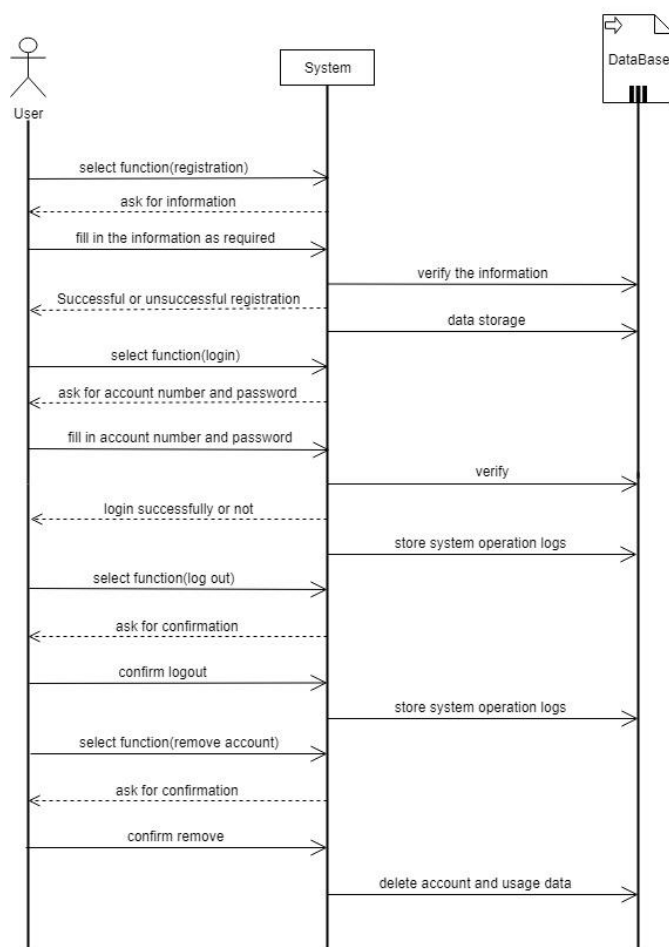
Store code



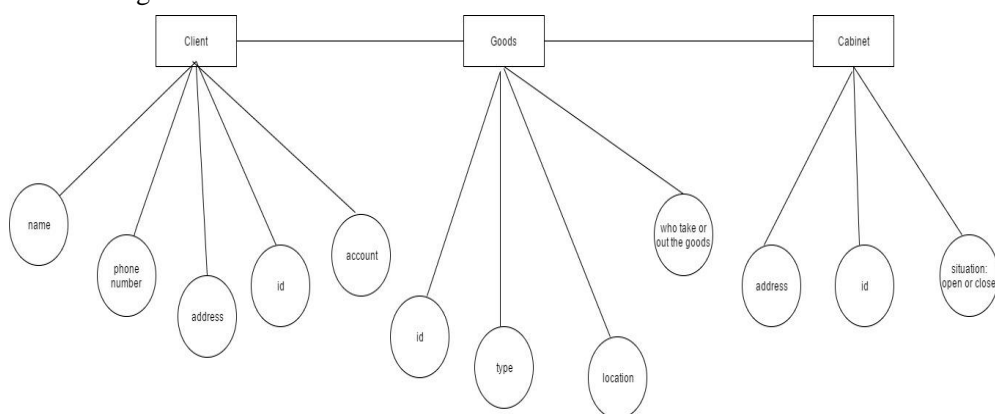
Evaluation control



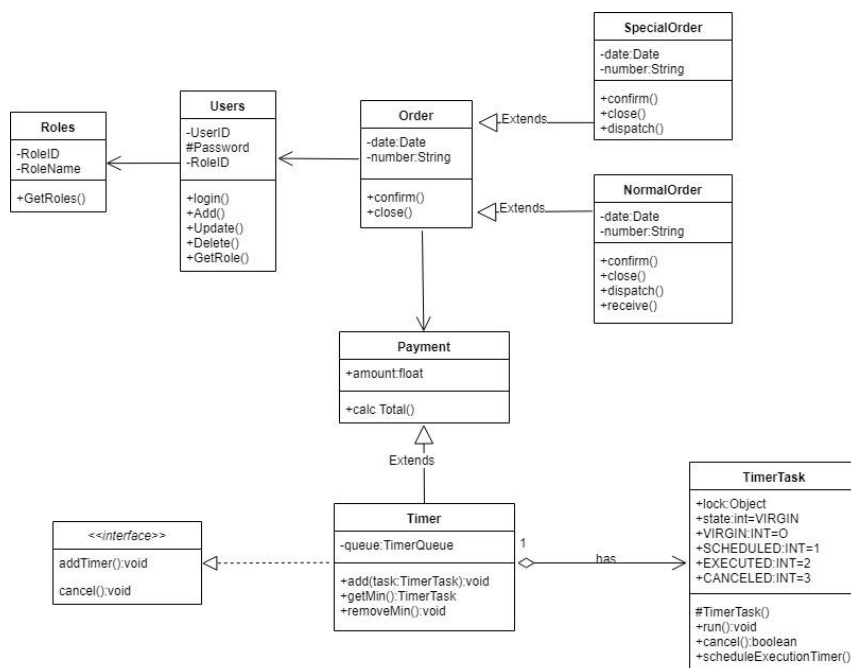
User Control



RE-diagram

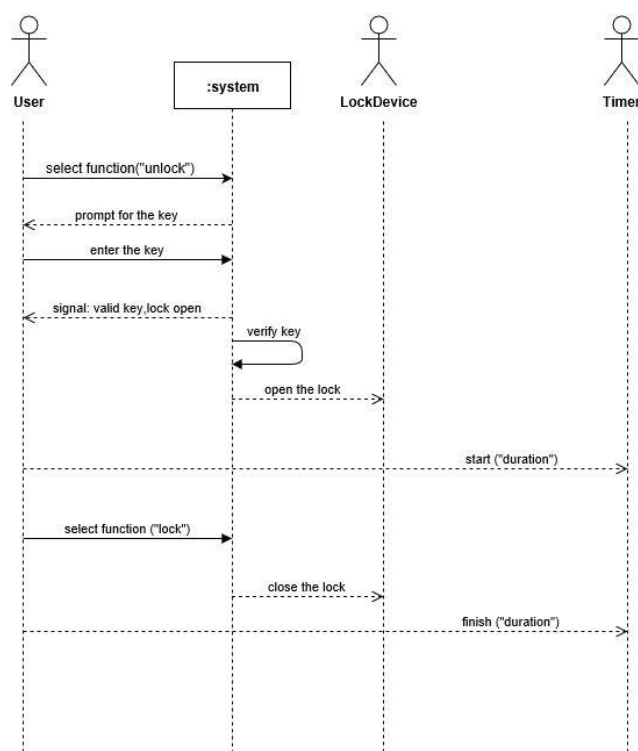


8. Class Diagram and Interface Specification



Design Patterns

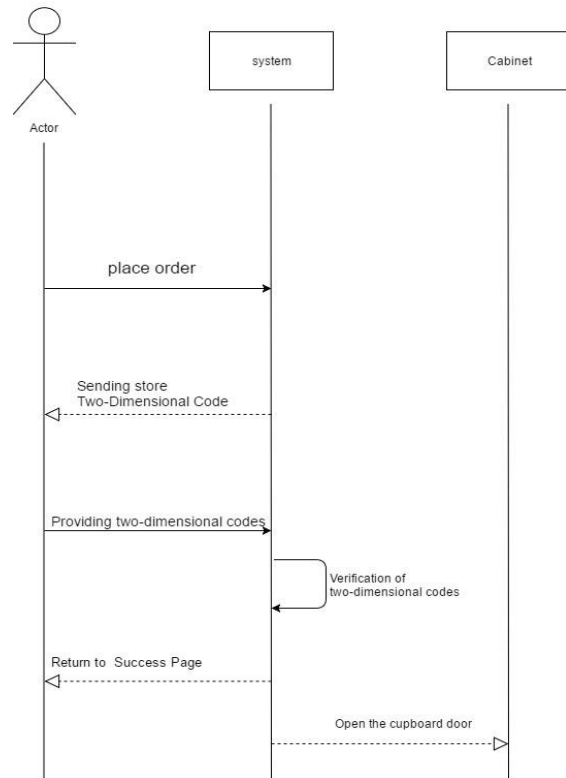
Cabinet Control



Cabinet control

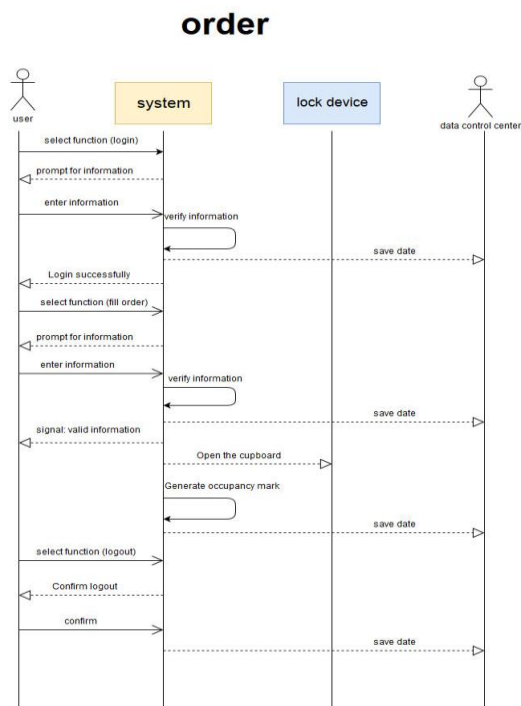
The user chooses to unlock, and the system prompts the user to enter the validation code. LockDevice opens the cabinet after the user enters the validation code system to verify that it is correct. Then the user closes the cabinet.

Store code



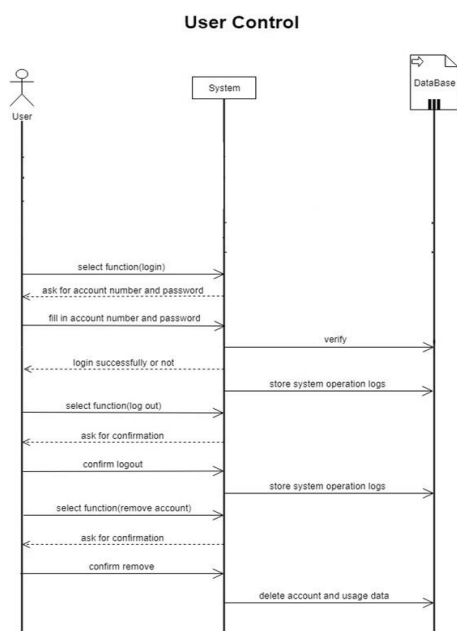
Store code

After users place orders, the system will send authentication codes to users. The user enters the authentication code and the system returns to the success page after verification. Cabinet opens the cabinet door.



Order

Log in to see if there are any orders. Some orders go to the order page. No orders directly show no orders. Users with orders enter validation codes and open the cabinet door after system validation is correct. Users can log out.



User control

Users choose to login, and the system displays the login page. The user enters the phone number and verification code, and the system verifies whether the login is successful or not. Database storage system operation log. Users can opt out and the

system requires confirmation. User can quit after confirmation.

We use the singleton pattern .

Our entire system only needs to have a global object, which is conducive to our coordination of the overall behavior of the system. Singleton pattern can be controlled by instances. The singleton pattern prevents other objects from instantiating copies of their own singleton objects, thus ensuring that all objects have access to unique instances and flexibility. Because classes control the instantiation process, classes can flexibly change the instantiation process.

But singleton pattern also has its drawbacks.

1. expenses

Although the number is small, it will still require some overhead if an instance of a class is checked every time an object requests a reference. This problem can be solved by using static initialization.

2. Possible development confusion

When using singleton objects (especially those defined in class libraries), developers must remember that they cannot instantiate objects using new keywords. Because the source code of the library may not be accessible, application developers may find themselves unable to instantiate such a class directly by accident.

3. Object Survival Period

The problem of deleting a single object cannot be solved. In languages that provide memory management, such as .NET Framework-based languages, only singleton classes can cause instances to be unassigned because they contain private references to that instance. In some languages (such as C++), other classes can delete object instances, but this can lead to suspended references in singleton classes.

Object Constraint Language (OCL) Contracts

1. Context Login

Inv: usersAccount. login

Pre: users registered accounts

Post: enter. password

2. Context courier delivers

Inv: choose box. storage expresses

Pre: courier. registered accounts

Post: box->select->size->is empty

3. Context retrieve the express

Inv: courier. retrieve

Pre: users->select->size->is empty

Post: courier. getOrder

4. Context timer

Inv: timer. Record operation time

Pre: box. Opening

Post: box. closed

5. Context unlocks

Inv: users and courier. unlock

Pre: system. send verification Code

Post: enter. verification code

6. Context lock

Inv: users and courier. lock

Pre: users and courier. put and take

Post: box. remind

7. Context scan verification code

Inv: box. scan.

Pre: users and courier. get verification code

Post: box. scan. Verify

8. Context choose the size of the box

Pre: users and courier. select->size->is empty

Post: box. closed

9. System Architecture and System Design

Architectural Styles

The software uses a three-tier client/server (C/S) architecture style. The three-tier C/S structure divides application functions into three parts: presentation layer, function layer and data layer. Our storage cabinet utilizes this architecture style, realizes three levels of independence logically, and makes the system more simple and clear.

The presentation layer is the user interface part of the application, which bears the function of dialogue between users and applications. It is used to check the data input from the keyboard and other users, and displays the data output from the application. In order to allow users to operate intuitively, graphical user interface (GUI) is generally used, which is simple to operate, easy to learn and easy to use. When changing the user interface, only the display control and data checking procedures needs to be rewritten, without affecting the other two layers. The content of the inspection is also confined to the form and value of the data, excluding the processing logic of the business itself. Visual programming tools are mainly used for the development of this layer.

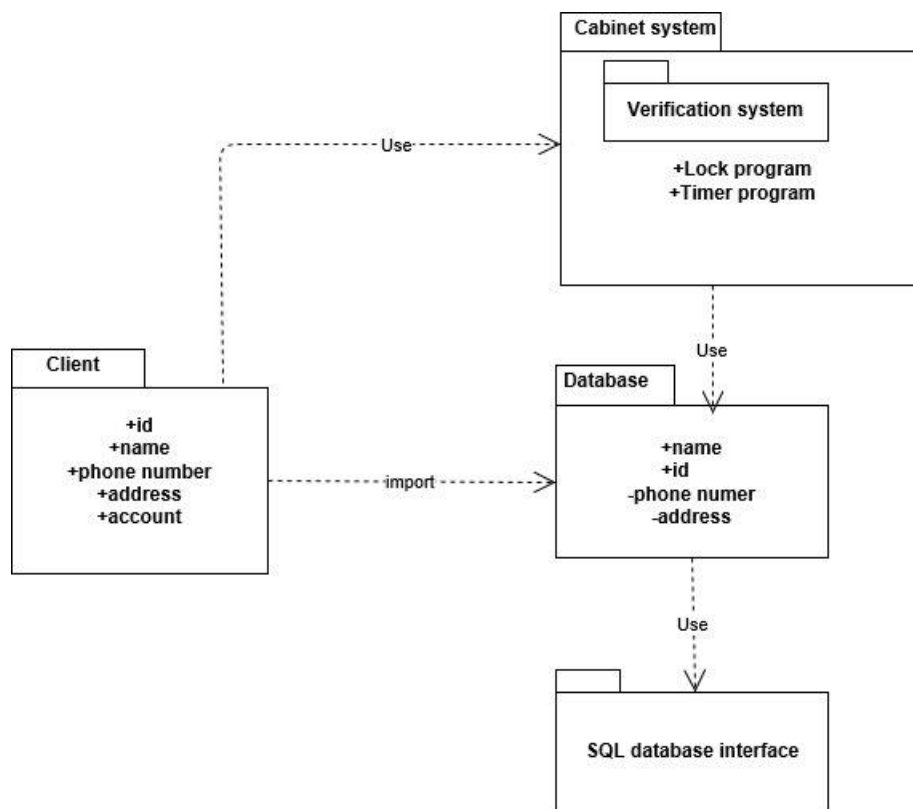
Functional layer is equivalent to the ontology of the application, which is to logically program specific business processing. In our storage cabinet system, when users access the order, they are required to calculate the order amount, configure the

data according to the given format, and output the order. The data was required to be processed here should be obtained from the presentation layer or the data layer. In the functional layer, it includes the function of confirming user's access to application and data inventory, and the function of recording system's log processing.

The data layer is DBMS, which handles the reading and writing of database data. DBMS can quickly update and retrieve large amounts of data. Therefore, customer information, order information, and evaluation will be kept in the data layer to facilitate system calls and reading and writing.

Using this three-tier C/S structure in the storage cabinet can improve the maintainability of the program and guarantee information security and user experience for customers.

b. Identifying Subsystems



c. Mapping Subsystems to Hardware

none

d: Persistent Data Storage

Our data processing cannot be instantaneous, because we need to record all of the information. When the users enroll their information and register in this system, our database process system needs to record this information every times. When the

courier put the takeout into the cabinets, our database process system needs to record this information every times, because we need to know who put the takeout into the cabinets. It means that we need ensure the food security and responsibility of every courier. When the users finished the whole process, our database process need to record all the information, because we need to know what kind of food users like. And in the next time, we will recommend the similar food to the users; it is benefit for the merchants to make some plans to the users. Our data storage must be permanent, not instantaneous through the above description. About the database schema, we need to record all of the information relate to the users and couriers.

3e: Network Protocol

none

3f: Global Control Flow

Execution orderness: our system procedure-driven and executes in a “linear” fashion, where every user every time has to go through the same steps, and every user can’t generate the actions in a different order.

Time dependency: our system has timers, it is real-time, it’s not periodic.

Concurrency: our system don’t use multiple threads.

3g: Hardware Requirements

We need lock control device module, card reading device module, bar code and QR code reader, video module, 32K horizontal needle bill printer (printing courier sheet), short message module, etc., the main cabinet door configuration is: electronic control on/off; item detection sensor, Windows 7 system with 21 inch touch LCD screen, WiFi wireless Internet module.

About the hardware selection and explanation:

CPU: Intel, Core i3, 4130

Memory: Kingston 4GB_DDR3_1600_

Hard Disk: Seagate Barracuda 2TB 7200 RPM 64MB SATA3

Wifi module: Waveshare_MX1081_stm32_wifi module_

POS Card Brushing Machine: Yibao POS Machine

Banknote Checking and Change Finding Module: Rongkang 866 Small Banknote Checking Machine

Voice prompt module: Blue Horse LMD107_

Digital keyboard module: kangaroo DS-9018A/9018_

Touch screen control module: DC80480B070_02TW_10.2 inch with touch_1280*720TFT_1G memory with configuration/USB/SD/5-26V power supply, etc.

Short Message Module: ATK-SIM900_GSM/GPRS Module Short Message Telephone Development Board_

Bill Printing Module: Jiabo GP5860III Thermal Small Bill Printer

Video equipment: Longshian 8-way all-D1 hard disk video recorder

Bar Code and Two-Dimensional Code Reader: Symbol DS-6708 Two-Dimensional Bar Code Scanner

Card Reader Module: 125KHz RFID Module Radio Frequency Electronic Card Reader

Lock control equipment: intelligent electric latch

10. Algorithms and Data Structures

The function of uploading offline data on time in this class

```
class AppRun3 implements Runnable {
    @Override
    public void run() {
        String chr1 = System.getProperty("line.separator"); String chr = "##";
        ArrayList<CoffDataProperty> lstOff = new ArrayList();
        int time = 5000 * 1;

        while (!m_bExit) {
            try { Thread.sleep(time); } catch (InterruptedException e) {}

            try {
                lstOff.clear();

                // Query whether rests need to be uploaded in the database
                String strSql = "select fi_ID,fi_Type,fs_Data from tb_OffLineData order by fi_ID asc limit 0,50";
                QueryEntity result = CDBHelper.getInstance().Query(strSql); // Query data
                if (result.hasData) {
                    ResultSet rs = result.dataRs;
                    try {
                        while (rs.next()) {
                            CoffDataProperty entity = new CoffDataProperty();

```

This ArrayList is the quantum structure.

This part is the function of background connection.

```
public void AddWebLog(int msgType, String msgContent) {
    String orderID = "";
    switch (CDataMgr.CurrentAction) {
        case CBaseEnum.Action_IDVCJ: orderID = CDataMgr.IDVOrderID; break;
        case CBaseEnum.Action_IDVQJ: orderID = CDataMgr.IDVOrderID; break;
        case CBaseEnum.Action_YHQJ: orderID = CDataMgr.UserOrderID; break;
    }
    TimeEntity entity = CDateHelper.GetNowTimeEntity();
    // String strSql = "insert into tb_DeviceLog (fi_DeviceID,fs_OrderID,fi_HappenTime,fs_HappenTime,fi_MsgType,fs_Note) values (" + CDataMgr.DeviceID + "','" + orderID + "','" + entity
    // func.CCommonFunc.AddRest_Executes(strSql);
    // CTabiniHandleRaca.Process6000(orderID, entity.TimeStart, entity.TimeValue, msgTime, msgContent);

```

This part is the function of get available ports

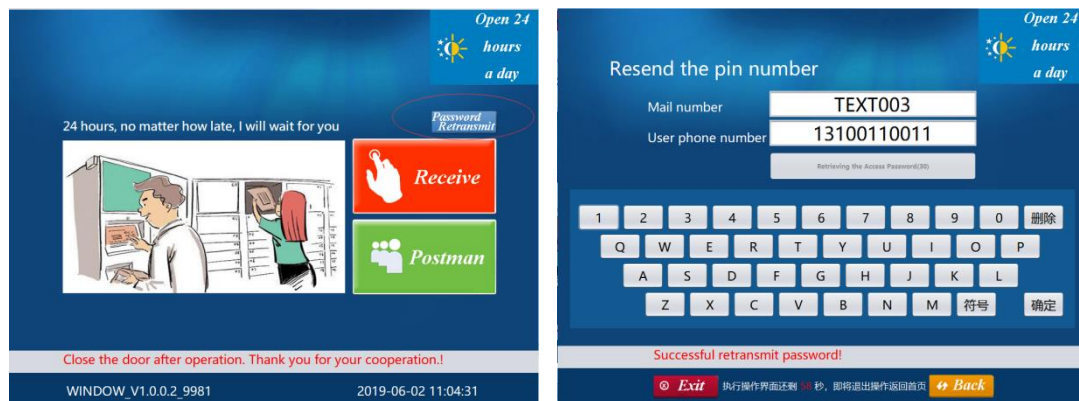
```
public static List<String> GetPortNames()
{
    List<String> lstRet = new ArrayList<String>();
    CommPortIdentifier portId;
    Enumeration<?> en = CommPortIdentifier.getPortIdentifiers();
    while (en.hasMoreElements())
    {
        portId = (CommPortIdentifier) en.nextElement();
        if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL)
        {
            lstRet.add(portId.getName());
        }
    }

    return lstRet;
}
```

11. User Interface Design and Implementation

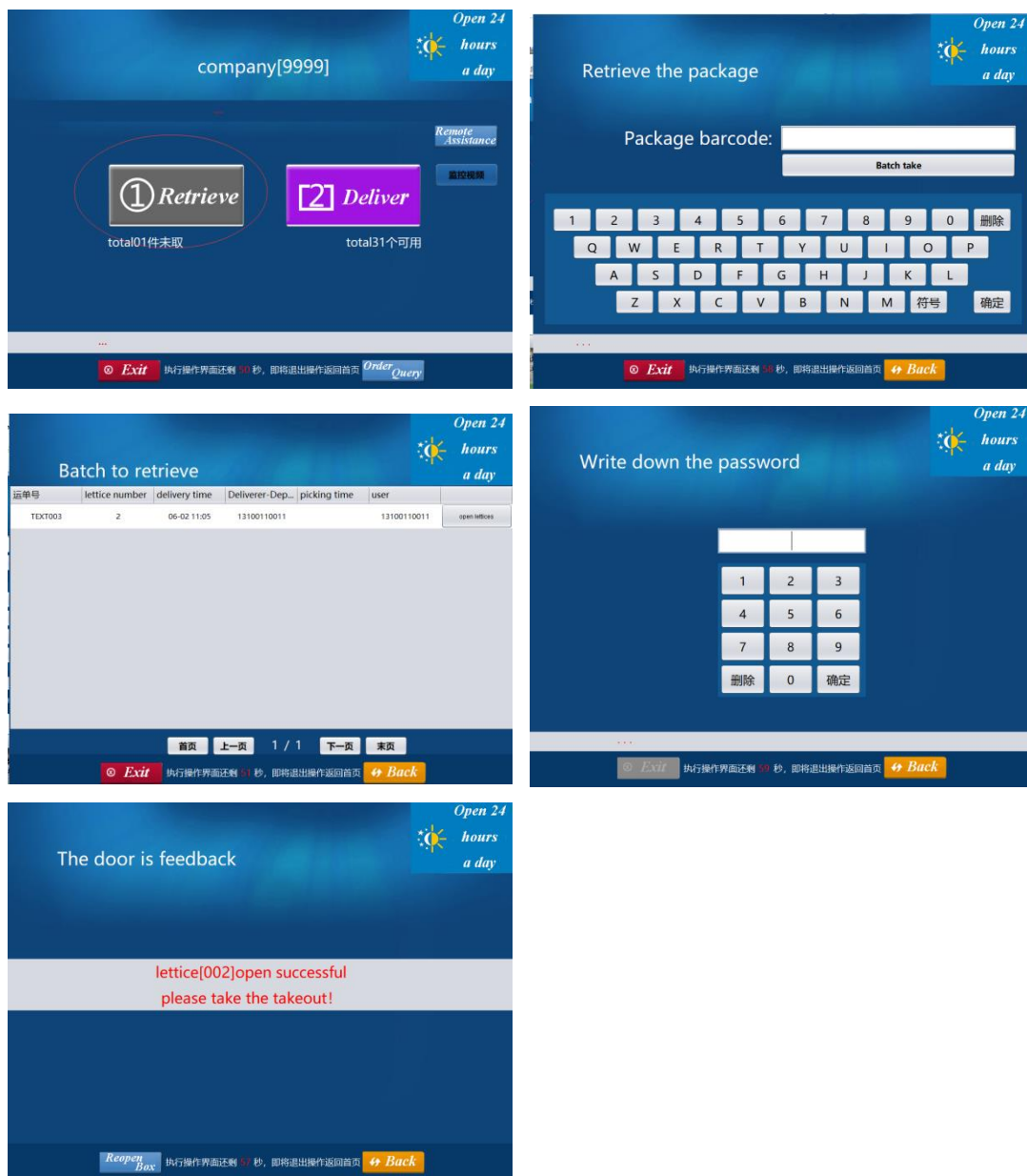
a. Preliminary Design:

Password Re-transmit



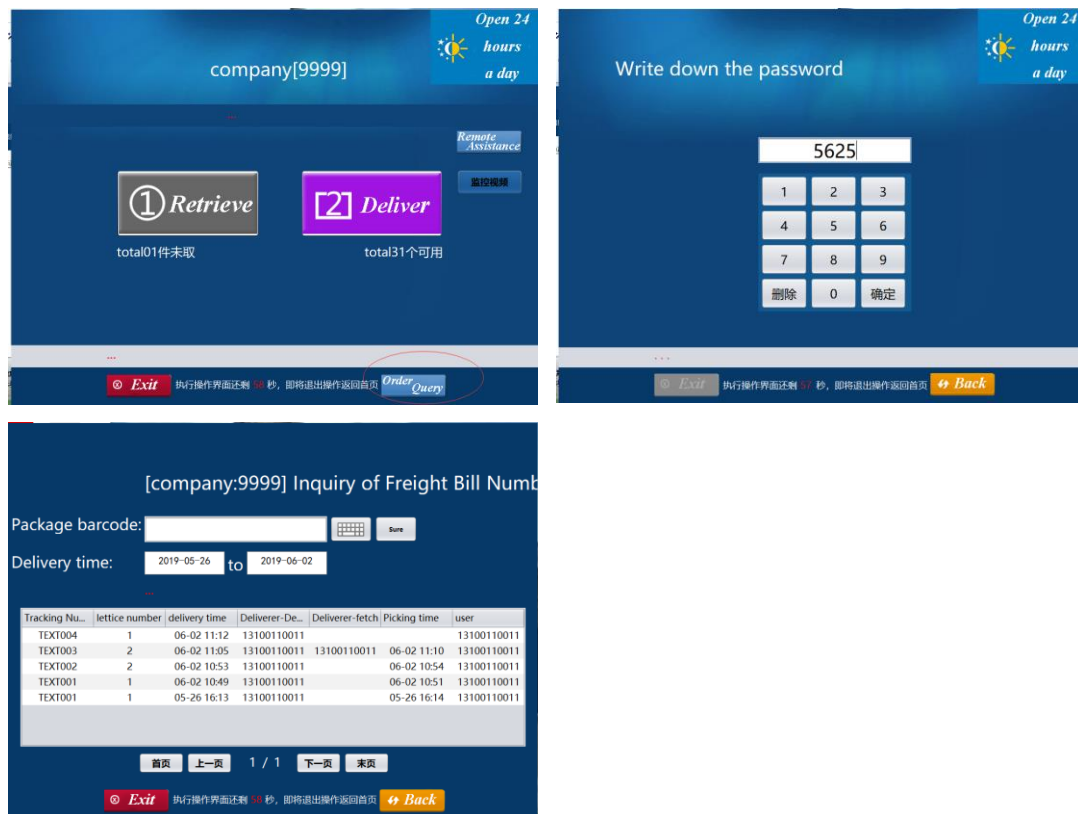
When the receiver needs to get the password again, click on the "Password Re-transmit" button on the main interface, and then enter the order information and the receiver's phone number.

Postman Retrieve



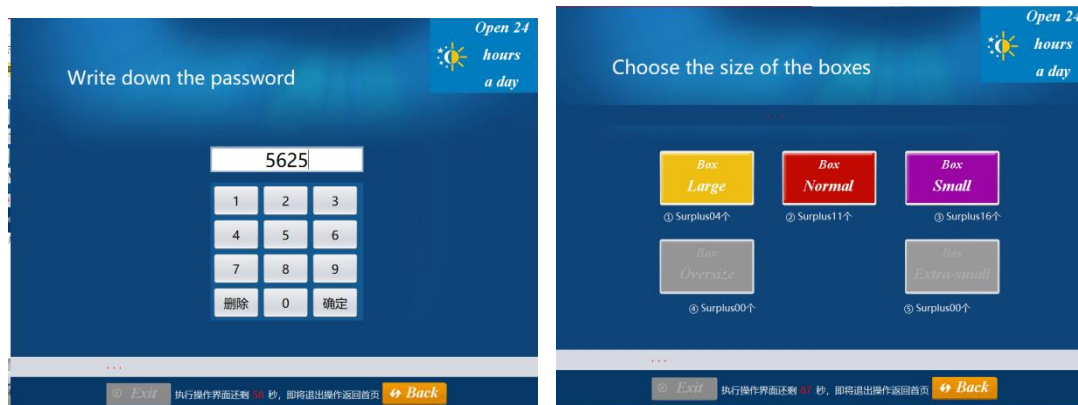
When the postman wants to retrieve the package, he clicks the "Retrieve" button, then enters the order number, selects the package to retrieve, and finally enters the password to submit.

Postman Order Query



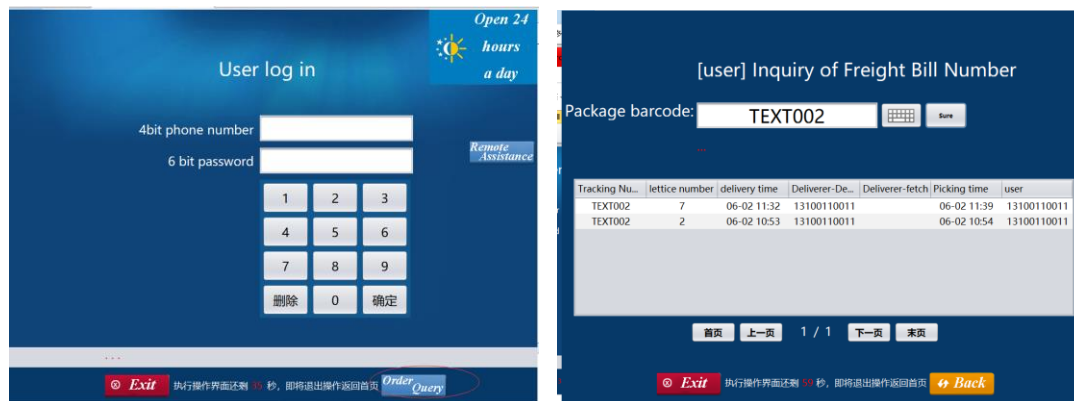
When a postman needs to inquire about his own delivery record, he clicks the "Order Query" button, he can browse all the records after entering his password. He can also enter an order number to search.

Re-select Cabinet



When the sender needs to re-select the cabinet, he needs to close the open cabinet, click the "Change the cabinet" button, and then select the cabinet he needs.

Receiver Order Query



When a receiver needs to inquire about his own receive record, he clicks the "Order Query" button, he can browse all the records after entering his password. He can also enter an order number to search.

b. User Effort Estimation:

Password Re-transmit

A total of 33 clicks on the screen, as follows

Click the "Password Re-transmit" button

Click the "Order number" input box

Click "x" "x" "x" "x" "x" "x" "x" "x"

Click the "Receiver's phone number" input box

Click "x" "x" "x" "x" "x" "x" "x" "x" "x" "x" "x" "x"

Click the "Confirm" button

Postman Retrieve

A total of 18 clicks on the screen, as follows

Click the "Retrieve" button

Click the "Order number" input box

Click "x" "x" "x" "x" "x" "x" "x" "x"

Click the "Confirm" button

Click the "Open cabinet" button

Click the "Password" input box

Click "x" "x" "x" "x"

Click the "Confirm" button

Postman Order Query

A total of 17 clicks on the screen, as follows

Click the "Order Query" button

Click the "Password" input box

Click "x" "x" "x" "x"

Click the "Confirm" button

Click the "Order number" input box

Click “x” “x” “x” “x” “x” “x” “x” “x”

Click the "Search" button

Re-select Cabinet

A total of 8 clicks on the screen, as follows

Click the "Change the cabinet" button" button

Click the "Password" input box

Click “x” “x” “x” “x”

Click the "Confirm" button

Click the "Normal Box" button

Receiver Order Query

A total of 17 clicks on the screen, as follows

Click the "Order Query" button

Click the "Password" input box

Click “x” “x” “x” “x”

Click the "Confirm" button

Click the "Order number" input box

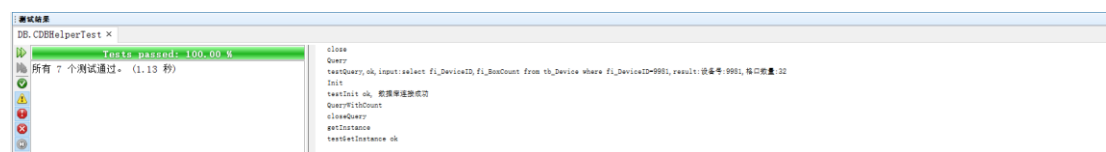
Click “x” “x” “x” “x” “x” “x” “x” “x”

Click the "Search" button

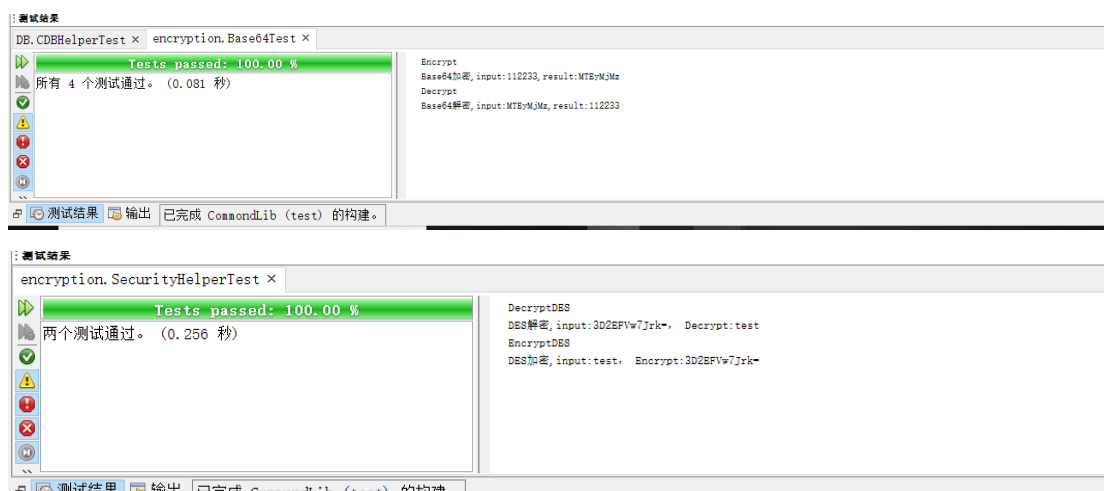
12. Design of Tests

In our whole program, we write the five Junit tests and one integration test. These tests are show in our program. The five Junit tests are CDBHelperTest.java, Base64Test.java, SecurityHelpTest.java, CPipeHelpTest.java, CTxtHelpTest.java, CBaseTimeTest.java, the one integration test is JUnitTestIT1.java.

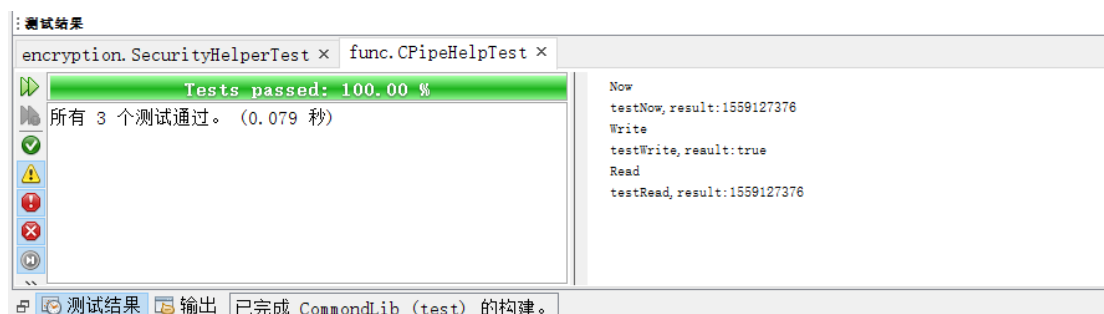
First I will explain the Database Operation test. This test mainly tests the function of the following aspects: the first is to test whether the database is connected successfully. We write a text on the D disk, the second is to test the number of device and the number of grid ports in the database. This is screenshot



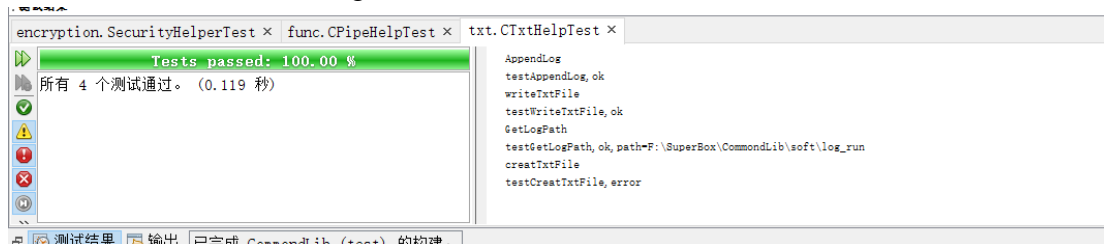
Second I will explain the Message Encryption and Decryption Function test. This test we write two functions, one is base 64 encryption and decryption, the other is des encryption and decryption. About the base 64 encrypt, we test the “112233”, we need to get the consequence is “112233”, too. About the base 64 decrypt, we test the “MTEyMjMz”, we need to get the consequence is “MTEyMjMz”. About the des encrypt, we test the key is "auaspp01" and the input is "test", we need to get the consequence is "auaspp01" and input is "test" . About the des decrypt, we test the key is "auaspp01" and input is "3D2EFVw7Jrk=", we need to get the consequence is "auaspp01" and input is "3D2EFVw7Jrk=". This is screenshot



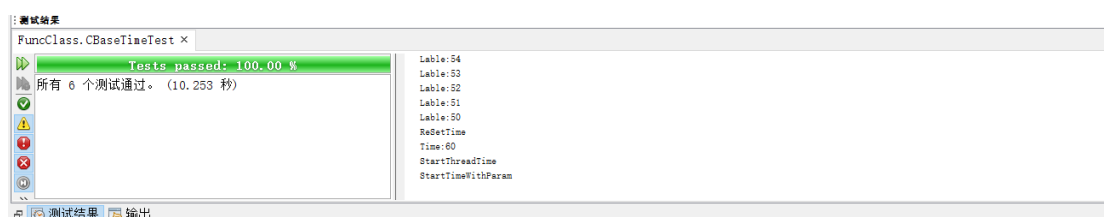
Third I will explain CPipeHelpTest. This function is that in order to restart the software when the system crashes, the terminal software writes the current timestamp to the pip. The software watchdog obtains this value for judgment. We can regard it as a file, a write, a read. This is screenshot



Forth I will explain CTxtHelpTest. This test is Logging function, Our main test is appendlog, we need to get the consequence of the appendlog is right or wrong, then, we test the createTxtFile, we need to get the consequence of the createTxtFile is right or wrong, finally, we test the path, and we get the consequence of the path is SuperBox\CommondLib\soft\log_run. This is screenshot

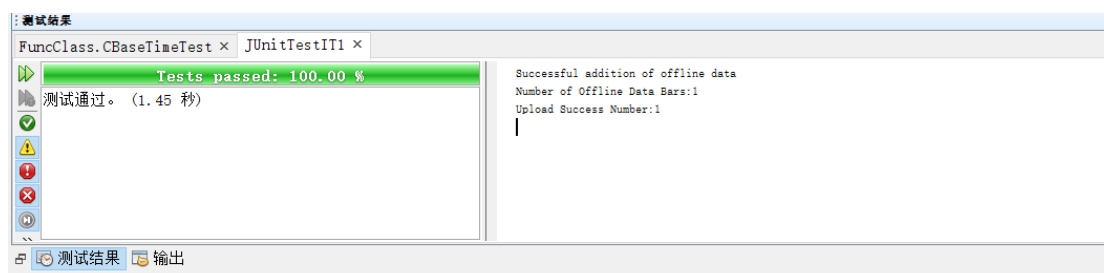


Fifth I will explain CBaseTimeTest. This function is that Countdown test. We design the 10 seconds to test, the 10 seconds is 50 to 59. This is screenshot



All these are Junit tests.

Then I will explain the integration test, we write the function is that Testing offline data upload function. We chose to add an offline data to see if the system could receive the data. This is screenshot



13. History of Work

History of Work, Current Status, and Future Work

At the beginning of the term, we started to design software about campus express cabinet. Firstly, we analyze the market demand and feasibility of express cabinet according to the real situation of campus daily life. Then, based on the interviews and questionnaires of college students and couriers, we found that the demand for courier cabinets is very high. We decided to start developing the "BOX (campus courier cabinet)" project. In the process of market demand analysis and investigation, we have completed the first task - Project Proposal.

Soon, we had our second task, Report One. After looking at the requirements of report one, we realized for the first time that it was not easy to implement the software. Our group began to carry out group discussion, division of labor in the library seminar room, and then completed their respective tasks. We discussed eight requirements realistically, and then summarized 16 use cases according to the requirements. According to these drawings, system diagrams, tracking matrices, use case diagrams, detailed descriptions of use cases and interface design are drawn.

Next, we started to work on the project code of express cabinet. At first, we simply implemented UC code for the user's main interface. But the computers of our team members cannot connect to the database. This problem has been bothering us for a long time. After many attempts, we turned to other students with high computer level. Through the joint efforts of the team members, we finally completed the Chinese version of the express cabinet project framework and code, and connected to the database.

With nearly a month of code implementation, we completed report two. Interaction diagrams and class diagrams have been drawn, and some data structures and system architectures have been completed. But we also found some problems through this code production. We deleted six use cases (registration, login, cancellation, payment, evaluation) because in the process of code implementation, we think that the routine means of registration, login and cancellation are too cumbersome, which makes it difficult to use the cabinet, and takes up too much memory space, but the actual effect of registration is not great, so we delete them.

Express courier can directly use cabinet storage through unified certification. Users can inquire for the order by inputting mobile phone number, and quickly pick up the parts. This can give users a better sense of experience. At the same time, in order to adapt to the fast-paced campus life and learning, and more convenient for students to use, we decided to let the express cabinet provide services free of charge, while canceling the evaluation function, making the user experience more simple and direct.

After completing the above work, the express cabinet project has gradually become clear in the eyes of our team members. We started to optimize the details and modify the areas that had not been achieved or changed in the previous plan. At the same time, report three is completed. We also modified the software to English version.

So far, we have basically completed the production of campus express cabinet.

About the future of this system: because our current capabilities cannot meet our assumptions, we will still work harder to learn relevant knowledge in the future, and apply it to this project, and finally achieve a nearly perfect and usable express cabinet system.

14. References

URL: <https://github.com/magical-group/Report3>