Project name: Self pick-up cabinet

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

URL:

https://github.com/klaus668/project-2

https://github.com/kzh0726/repott2

https://github.com/yumao-anna/report2

https://github.com/Hexbruce/project-two

https://github.com/OliviaWY29/software-engineering-/blob/master/pickup.zip

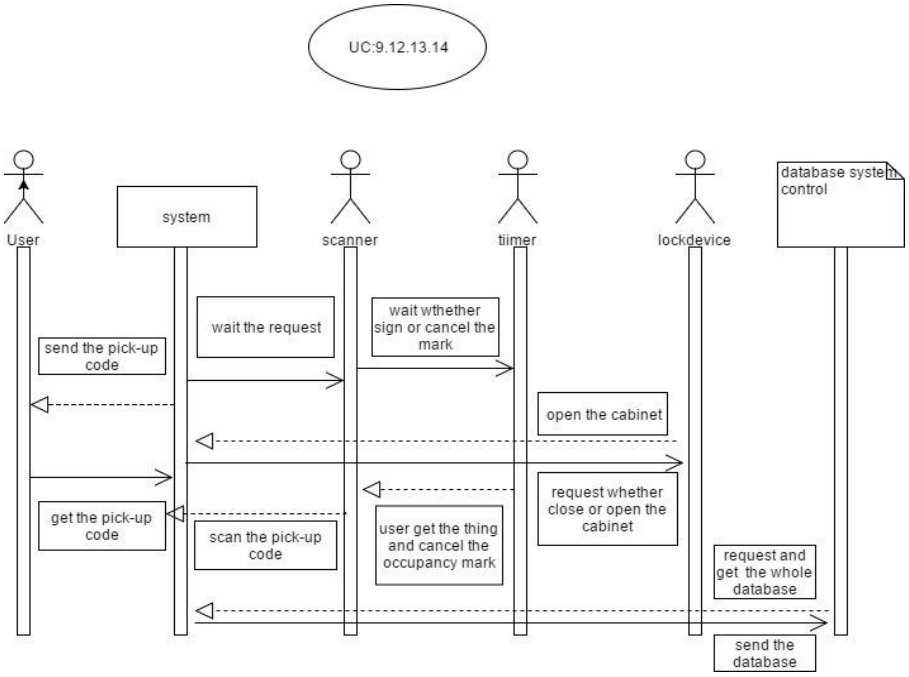
https://github.com/Ivy915/project-one/blob/master/pickupSucc.zip

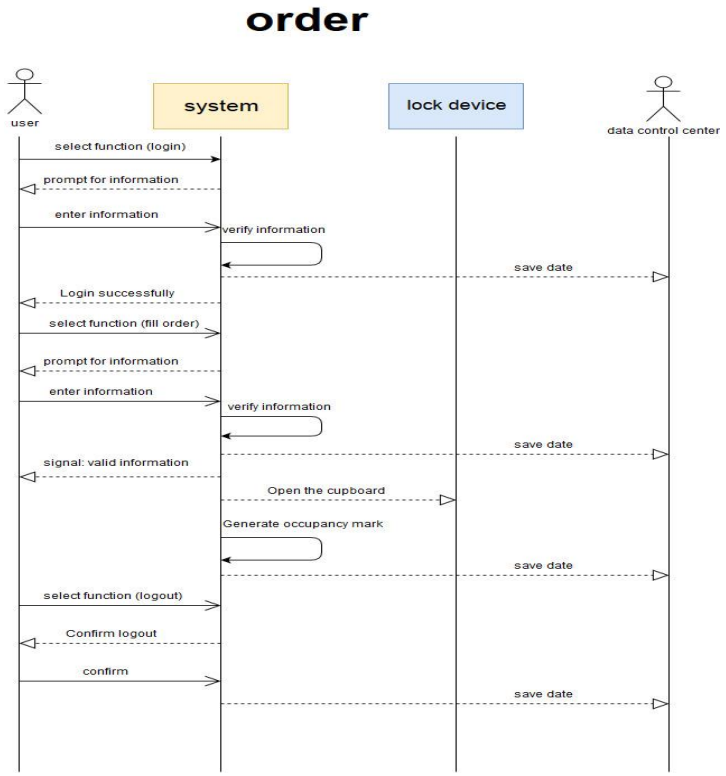
https://github.com/magical-group/project-two

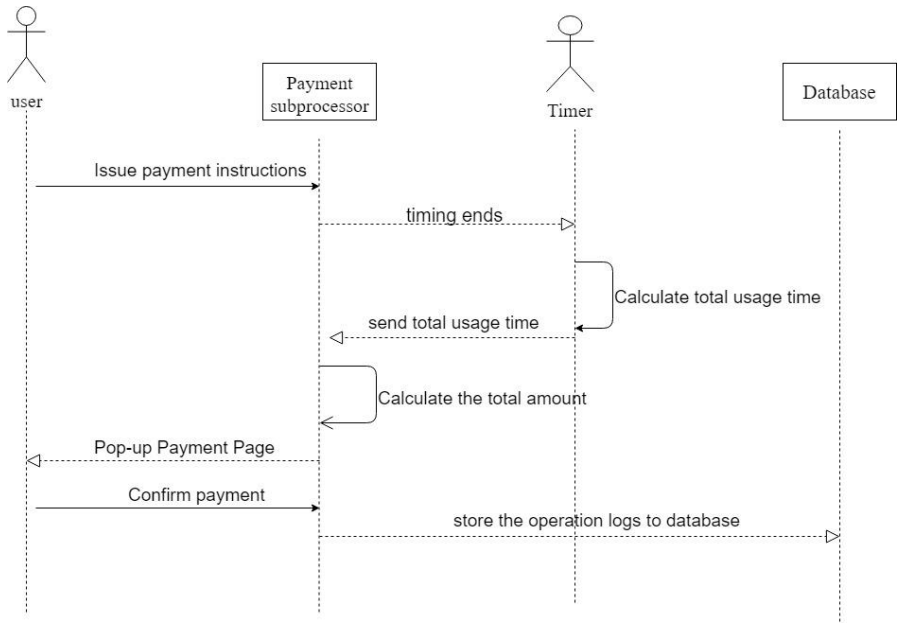
Table of Contents

Part 1:

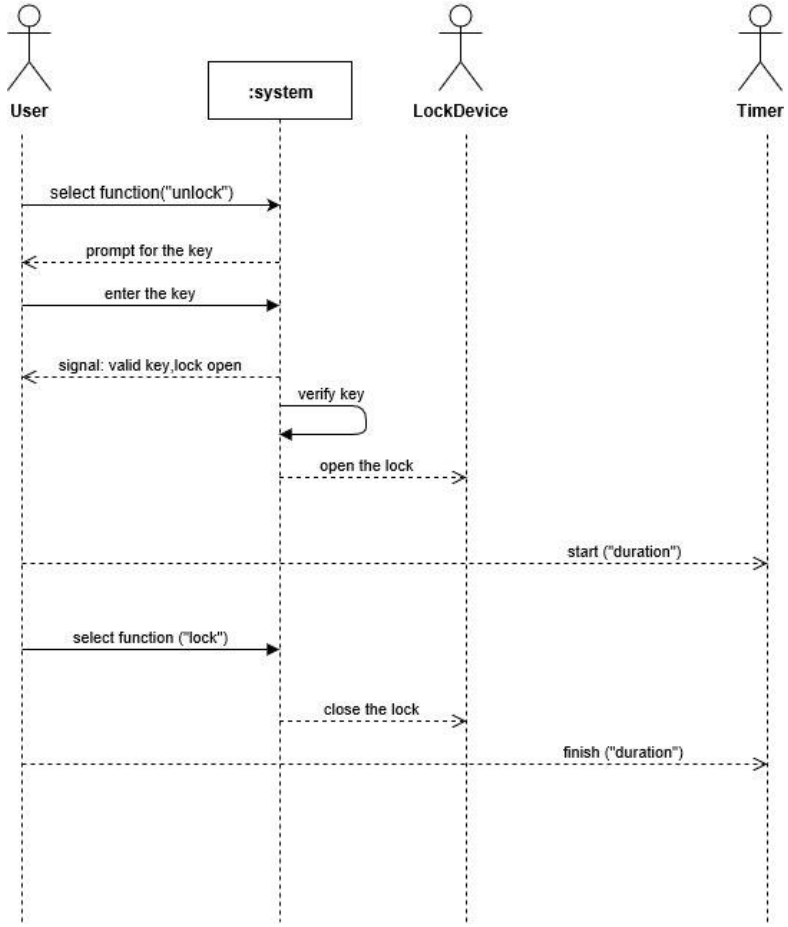
Interaction Diagrams



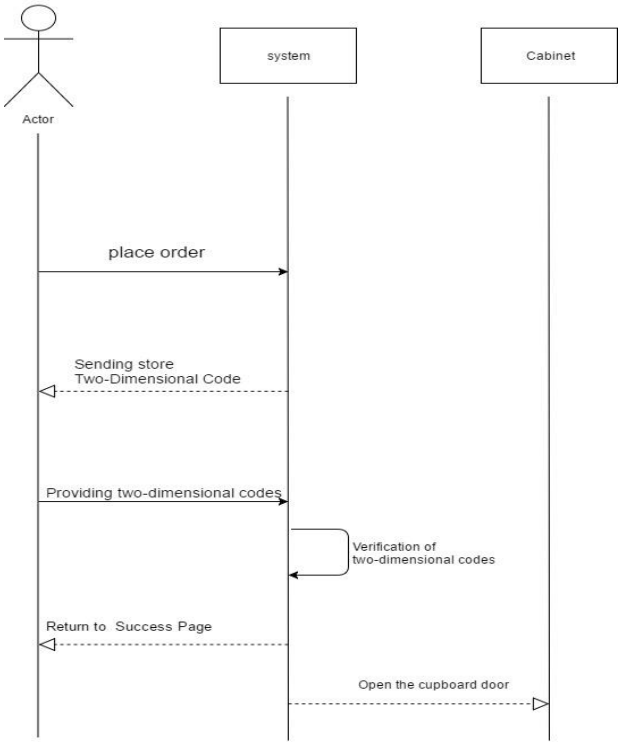




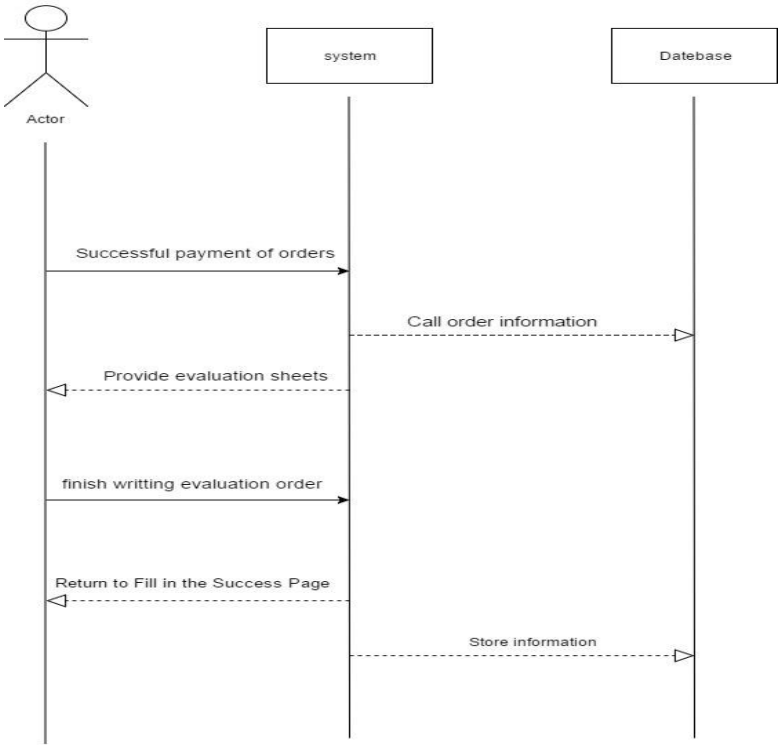
Cabinet Control

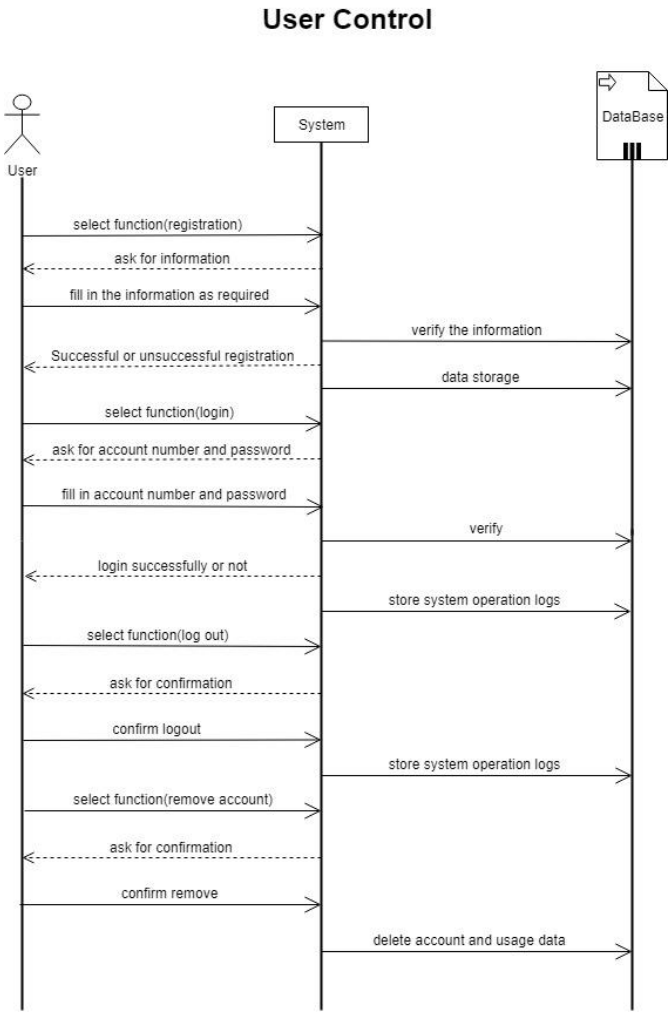


Store code

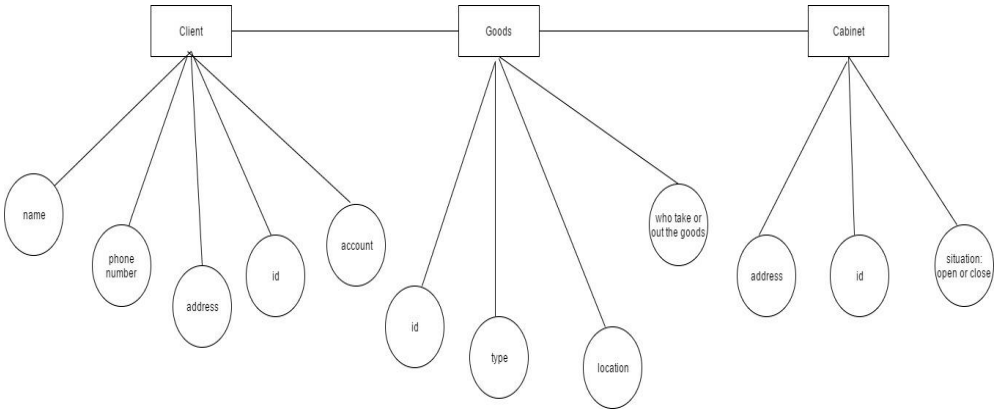


Evaluation control




User Control
User
System
DataBase
select function(registration)
ask for information
fill in the information as required
verify the information
Successful or unsuccessful registration
data storage
select function(login)
ask for account number and password
fill in account number and password
verify
login successfully or not
store system operation logs
select function(log out)
ask for confirmation
confirm logout
store system operation logs
select function(remove account)
ask for confirmation
confirm remove
delete account and usage data

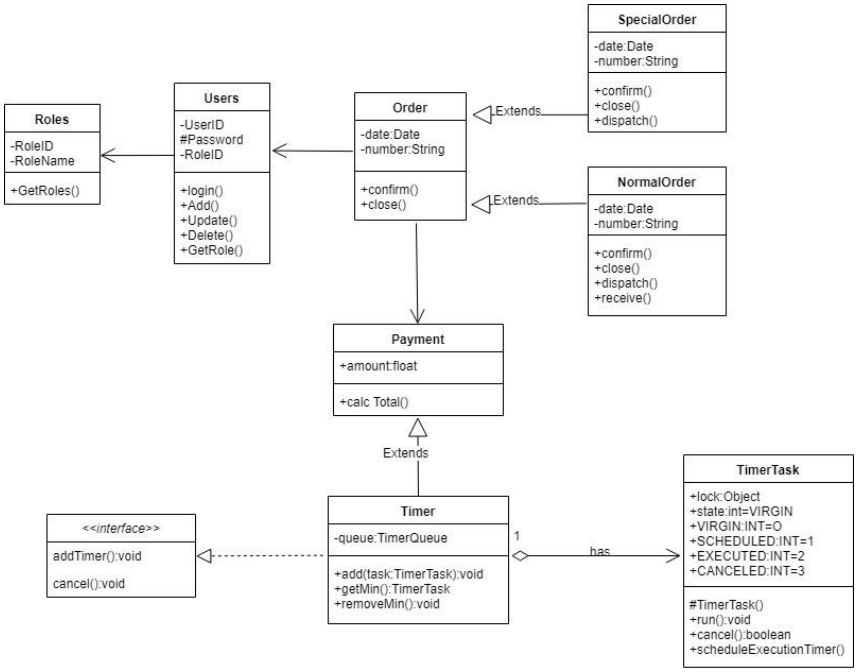
RE-diagram


Client
Goods
Cabinet
name
phone number
address
id
account
id
type
location
who take or out the goods
address
id
situation: open or close

Part 2:

2. Class Diagram and Interface Specification

a. Class Diagram



b. Data Types and Operation Signatures

Roles
-Role ID: char -Role Name: int
+Get Roles()

The Roles class is used to represent all the roles in use cases.

Users
-User ID: int #Password: char -Role Name: char
+Login() +Add() +Update() +Delete() +Get Roles()

The User class represents all users of the system.

Order
-date: Date -number: String
+confirm() +close()

The Order class are all created orders in the system.

Special Order
-date: Date -number: String
+confirm() +close() +dispatch()

The Special Order class is the orders that are incomplete and has not been submitted.

Normal Order
-date: Date -number: String
+confirm() +close() +dispatch() +receive

The Normal Order class are the orders with complete information and submitted.

Payment
+amount: float
+calc Total()

The Payment class refers to the payment status of user's order.

Timer
-queue: TimerQueue
+add(task: TimerTask): void +getMin(): TimerTask +removeMin(): void

The Timer class are all timers created by the the users.

TimerTask
+lock: Object +state: int=VIRGIN +VIRGIN: int=0 +SCHEDULED: INT=1 +EXECUTED: INT=2 +CANCELED: INT=3
#TimerTask +run(): void +cancel(): boolean +schedule Execution Timer()

The TimerTask class is used to represent all the timer states.

c. Traceability Matrix

Domain Concepts:

1. People need to register an account to become users. Users can be senders or receivers.
2. To open the cabinet for send, users need to create a new order. Fill in the order information to initiate the order, and the cabinet will be opened. Orders with irregular or incorrect information cannot open the cabinet.
3. After the sender locks the cabinet, the receiver will get a message with the t verification code for opening the cabinet. The receive opens the cabinet with the verification code and takes the goods.
4. After the receiver locks the cabinet, the system will prompt him to pay for the length of time he occupies the cabinet.
5. When the user has completed the payment, the order information will be supplemented and provided to the user for inquiry.
6. The length of cabinet occupancy needs to be calculated by a timer. Timers correspond to different states at different stages.

The User class is derived from Domain Concept 1.
The Order class is derived from Domain Concept 2.
The Special Order class is derived from Domain Concept 2.
The Normal Order class is derived from Domain Concept 2.
The Payment class is derived from Domain Concept 4.
The Timer class is derived from Domain Concept 6.
The Timer Task class is derived from Domain Concept 6.

3. System Architecture and System Design

a. 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.

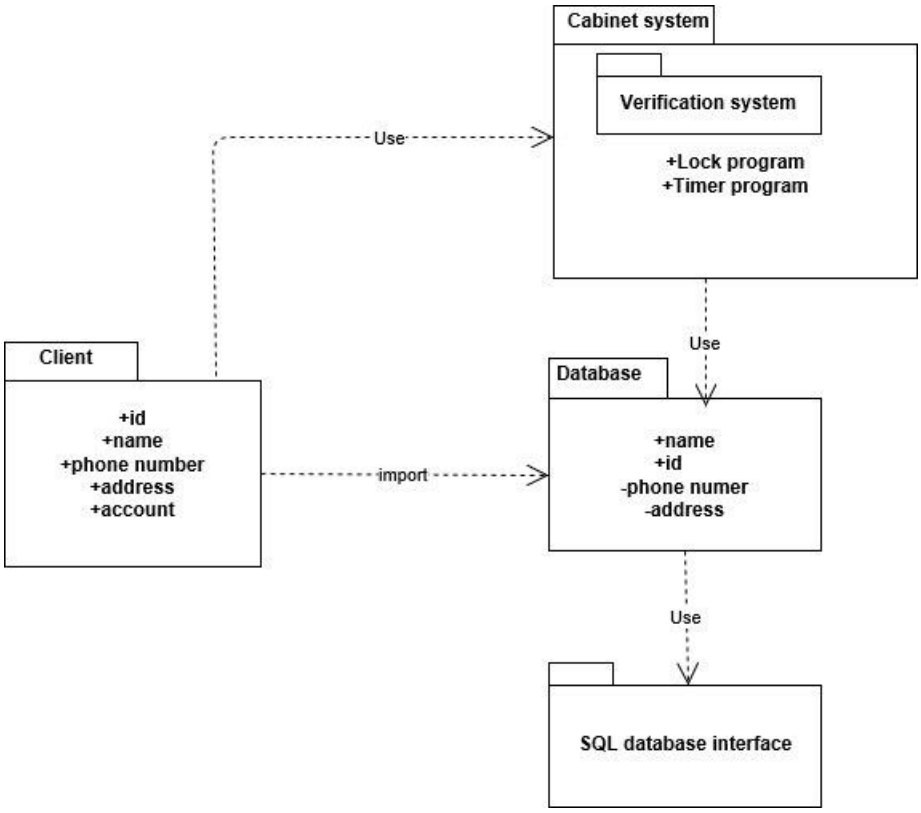
The presentation layer is the user interface part of the application, which bears the function of dialogue between users and applications. It is used to check the data input from the keyboard and other users, and displays the data output from the application. In order to allow users to operate intuitively, 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. Out 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