|  |
| --- |
| **Detail Design** |
| nju-ipay |
|  |
|  |
|  |
| popcorn team |
| **2011/7/18** |
|  |

catalogue

[1. introduction 4](#_Toc303082282)

[1.1. aim 4](#_Toc303082283)

[1.2. background 4](#_Toc303082284)

[1.3. reference 4](#_Toc303082285)

[2. Framework 5](#_Toc303082286)

[2.1. client 5](#_Toc303082287)

[2.2. SERVER 5](#_Toc303082288)

[3. CLIENT DESIGN 6](#_Toc303082289)

[3.1. FUNCTION 6](#_Toc303082290)

[3.2. performance 6](#_Toc303082291)

[3.3. input 6](#_Toc303082292)

[3.4. output 7](#_Toc303082293)

[3.5. algorithm 7](#_Toc303082294)

[3.6. process logic 7](#_Toc303082295)

[3.7 main interface 8](#_Toc303082296)

[3.8 comment design 12](#_Toc303082297)

[3.9 limit 13](#_Toc303082298)

[3.10 test plan 13](#_Toc303082299)

[4. market Server design 13](#_Toc303082300)

[4.1 program description 13](#_Toc303082301)

[4.2 function 13](#_Toc303082302)

[4.3 performance 16](#_Toc303082303)

[4.4 input 16](#_Toc303082304)

[4.5 output 18](#_Toc303082305)

[5. bank server design 21](#_Toc303082306)

[5.1 program description 21](#_Toc303082307)

[5.2 function 21](#_Toc303082308)

[5.3 performance 21](#_Toc303082309)

[5.4 input 21](#_Toc303082310)

[5.5 output 22](#_Toc303082311)

# introduction

## aim

The document is to detail the project, make it close to the final program. The document describes the whole framework, program interfaces, logical calls and so on.

## background

Project name : nju-ipay

Team : Popcorn Team

Users: client user – customer

Server user - market

## reference

1. Detail design（GB8567——88）
2. Medium-report.docx

# Framework

## client

UI

ENTITY

COMMUNICATION

BARCODE SCAN

## SERVER

领域层（Entity）

安

全

模

块（Security）

数据链路层（Dao）

业务逻辑（Service）

网络控制（Web）

Ajax客户端接口

后台界面Module

Database

客户端（client）

Jquery

Jsp

验证

# CLIENT DESIGN

## FUNCTION

|  |  |  |  |
| --- | --- | --- | --- |
| FUNCTION | INPUT | PROCESS | OUTPUT |
| Barcode scan | The bitmap from the camera | Barcode scan and handle, get product information from server | Product information |
| Input barcode | Product barcode | Submit to server | Product information |
| Pay | Password and shopping cart information | Submit to server | Toast for whether pay successfully |

## performance

* Look up product information should have multi-ways, barcode scanning or keyboard input.
* Inquiries and payment should be tolerable. Inquiry time should be less than 20 seconds and payment should be less than 60 seconds.

## input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | Domain | Input way | Frequency |
| Barcode picture | Bitmap | Judge by program | Camera scan | High |
| Barcode | String | Number and character | Handy input | Medium |
| User name | String | Number and character | Handy input | Low |
| Login password | String | Number and character | Handy input | Low |
| Payment password | String | Number and character | Handy input | High |
| Market name | String | No limits | Handy input | Medium |

## output

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Output style | Frequency |
| Product information | Product | List | High |
| Market information | Market | List | High |
| User information | UserInfo | List | Low |

## algorithm

Communication using MD5 and AES encryption algorithm.

## process logic

Graphically ( flow charts, decision tables, etc.) combined with the necessary instructions to represent the program's logic flow.

## main interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | instance | class | CommunicationManager | package | communication |
| Input | void | | | | |
| Output | CommunicationManager instance | | | | |
| Function | Get instance（singleton） | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | login | Class | CommunicationManager | Package | communication |
| Input | String username, String password | | | | |
| Output | int status | | | | |
| Function | Login | | | | |
| Detail description | According to the returned status code to determine the login status | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | logout | Class | CommunicationManager | Package | communication |
| Input | void | | | | |
| Output | Boolean | | | | |
| Function | Logout | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | getUserInfo | Class | CommunicationManager | Package | communication |
| Input | void | | | | |
| Output | UserInfo info | | | | |
| Function | View profile | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | setUserInfo | Class | CommunicationManager | Package | communication |
| Input | UserInfo info | | | | |
| Output | Boolean | | | | |
| Function | Set personal information | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | setPassword | Class | CommunicationManager | Package | communication |
| Input | String oldPassword, String newPassword | | | | |
| Output | Boolean | | | | |
| Function | Set password | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | searchMarket | Class | CommunicationManager | Package | communication |
| Input | String name, int pageNum | | | | |
| Output | ArrayList<Market> markets | | | | |
| Function | Search marcket | | | | |
| Detail description | PageNum >= 1 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | getEncryptPrivateKey | Class | CommunicationManager | Package | communication |
| Input | Context context | | | | |
| Ouput | Byte[] | | | | |
| Function | Get the private key | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | getMarketInfo | Class | CommunicationManager | Package | communication |
| Input | Int marketed | | | | |
| Output | MarketInfo | | | | |
| Function | Get market information | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | getProductInfo | Class | CommunicationManager | Package | communication |
| Input | String barcode | | | | |
| Output | ProductInfo | | | | |
| Function | Get detail product information | | | | |
| Detail description |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | initConnection | Class | CommunicationManager | Package | communication |
| Input | void | | | | |
| Output | void | | | | |
| Function | Get the id of the market to be connected | | | | |
| Detail description | Each client has successfully connected to the shopping network should first call this method after | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | pay | Class | CommunicationManager | package | communication |
| Input | Session session， String payPassword | | | | |
| Output | Order order | | | | |
| Function | Pay money | | | | |
| Detail description | If username in session equals null, it means not logged in, the client should first login in,  Failure to pay, returns null. | | | | |

## comment design

* Method function
* Parameter
* Return value
* Exception to throw

Detail :

/\*\*

\* *method function and description for using*

\* **@param** paramName

\* **@param** …

\* **@return**

\* **@throws** Exception

\* **@throws** …

\*/

## limit

* This procedure relies on the shopping network, enter the mall with the need to access the network
* Camera acquisition of goods rely on bar code information, so they requested that mobile phones with camera

## test plan

Refer to “test plan”

# market Server design

## 4.1 program description

Ipay’s market server.

## 4.2 function

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Input | Process | Output |
| Provide the client with the login interface | Account and password | Query the database and verify password | Whether login is successful |
| Provide the client with the searching market interface | Name or part of the shopping mall name | Query the database to meet the requirements of the mall | Shopping list |
| Provide the client interfaces to look up for detail market information | Shopping mall’s id | Get more information about shopping malls in the database | Shopping details |
| Provides the client interface to view Specials | Shopping mall’s id | Query corresponds to the mall in specials | A list of specials |
| Provides the client interface to view hot commodity | Shopping mall’s id | Query corresponds to the largest shopping malls in the sale of goods | Hot Products List |
| Provide client interfaces to search product | Shopping mall’s id and product name | Queries to meet the requirements under the trade names of goods | Product List |
| Provides the client interface to provide access to product information | Shopping mall’s id and product’s barcode | Query the database for the corresponding goods | Product Details |
| Provide payment interface for the client | Orders, that is, product id and quantity | Calculate the total price, payment requests sent to the bank | Bank payment response |
| Provide client interfaces to view personal information | None (provided that the first login) | Query the database | Personal details |
| Provide client interfaces to view credit card information | None (provided that the first login) | Request to the bank server, access to the results | Credit Details |
| Provides the client interface to view purchase history | None (provided that the first login) | Query the database | Purchase history list |
| Provide client interfaces to modify personal information | To modify personal information | Verify and save the changes to the database | Whether change successfully |
| Client / mall / Administrator Login | Account and password | Query the database and verify | Whether change is successful |
| Customer Registration | User information | Verify information and to modify the saved database | Whether login is successful |
| Shopping registration | Mall information | Verify information and to modify the saved database | Whether login is successful |
| View, modify, delete customer information | Modification to modify the data to be submitted | Query the database, change information to modify delete | Customer Information |
| View, modify, delete, shopping information | Modification to modify the data to be submitted | Query the database, change information to modify delete | Shopping Information |
| View, modify, delete product information | Modification to modify the data to be submitted | Query the database, change information to modify delete | Product Information |
| Show shopping day / week / month Income Statistics | None (mall to log in the background) | Query the database and generate charts | Product information statistics and charts |
| Display merchandise sales charts | None (mall to log in the background) | Query the database and statistics | Product List |

## 4.3 performance

* In the standard configuration to be able to support 500 people online while still able to quickly process the request (0.1s respond within), in 1000, when people can handle the request line (1s respond within)
* To ensure an error is encountered in the wrong machine, such as user error input, the database disorders. After the crash response system can be based on the log.
* Processing payment requests to respond within 0.5s.

## 4.4 input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Format | Domain | Input | Frequency |
| Provide the client with the login interface | Json Data Format | Number and string | https Network Access | Medium |
| Provide the client with the searching market interface | String | Characters with no punctuation | http network access | Medium |
| Provide the client interfaces to look up for detail market information | Number | Number | http network access | Medium |
| Provides the client interface to view Specials | Number | Number | http network access | Medium |
| Provides the client interface to view hot commodity | Number | Number | http network access | Medium |
| Provide client interfaces to search product | String | String with no punctuation | http network access | Medium |
| Provides the client interface to provide access to product information | Number or string | Number or string | http network access | Medium |
| Provide payment interface for the client | Json data format | String | https network access | Medium |
| Provide client interfaces to view personal information | None | None | https network access | Medium |
| Provide client interfaces to view credit card information | None | None | https network access | Medium |
| Provides the client interface to view purchase history | None | None | https network access | Medium |
| Provide client interfaces to modify personal information | None | None | https network access | Medium |
| Client / mall / Administrator Login | String | Number and string | Web Access | Medium |
| Customer Registration | Json data format | String | Web Access | Medium |
| Shopping registration | Json data format | String | Web Access | Medium |
| View, modify, delete customer information | Json data format | String | Web Access | Medium |
| View, modify, delete, shopping information | Json data format | String | Web Access | Medium |
| View, modify, delete product information | Json data format | String | Web Access | Medium |
| Show shopping day / week / month Income Statistics | None | None | Web Access | Medium |
| Display merchandise sales charts | None | None | Web Access | None |

## 4.5 output

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data format | Output | Frequency |
| Provide the client with the login interface | Json data format | String | Medium |
| Provide the client with the searching market interface | Json data format | String | Medium |
| Provide the client interfaces to look up for detail market information | Json data format | String | Medium |
| Provides the client interface to view Specials | Json data format | String | Medium |
| Provides the client interface to view hot commodity | Json data format | String | Medium |
| Provide client interfaces to search product | Json data format | String | Medium |
| Provides the client interface to provide access to product information | Json data format | String | Medium |
| Provide payment interface for the client | Encrypted bank data | Byte stream | Medium |
| Provide client interfaces to view personal information | Json data format | String | Medium |
| Provide client interfaces to view credit card information | Json data format | String | Medium |
| Provides the client interface to view purchase history | Json data format | String | Medium |
| Provide client interfaces to modify personal information | Json data format | String | Medium |
| Client / mall / Administrator Login | String | Page display | Medium |
| Customer Registration | String | Page display | Medium |
| Shopping registration | String | Page display | Medium |
| View, modify, delete customer information | String | Page display | Medium |
| View, modify, delete, shopping information | String | Page display | Medium |
| View, modify, delete product information | String | Page display | Medium |
| Show shopping day / week / month Income Statistics | String | Page display | Medium |
| Display merchandise sales charts | String | Page display | Medium |

# bank server design

## 5.1 program description

Ipay’s bank server.

## 5.2 function

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Input | Process | Output |
| Customer register | User information | Verify the information and save to the database | Whether register is successful |
| Process payment requests | OI，PI，OIMD。PIMD，DS | Data validation and save the modified query | Payment result |
| View card details | Bank card number | Query the database | Bank card details |

## 5.3 performance

* In standard configuration, can support 100 concurrent access while the response time less than 0.2 second.
* Error to ensure that improper machine, and to record error log; crash recovery to recover all the data

## 5.4 input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data format | Domain | Input | Frequency |
| Customer register | Json data format | String (not special symbols) | Client input | Medium |
| Process payment requests | Encrypted byte stream | RSA，AES，SHA Data Encryption Standard | httpNetwork Access | High |
| View card details | Encrypted byte stream | RSA，AES，SHA Data Encryption Standard | httpNetwork Access | High |

## 5.5 output

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data format | Output | Frequency |
| Customer register | String | Client display | Medium |
| Process payment requests | Encrypted byte stream | RSA，AES，SHA Data Encryption Standard | High |
| View card details | Encrypted byte stream | RSA，AES，SHA Data Encryption Standard | High |