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

[1. 引言 3](#_Toc299394804)

[1.1. 编写目的 3](#_Toc299394805)

[1.2. 背景 3](#_Toc299394806)

[1.3. 定义 3](#_Toc299394807)

[1.4. 参考资料 3](#_Toc299394808)

[2. 程序系统的结构 4](#_Toc299394809)

[2.1. 客户端 4](#_Toc299394810)

[2.2. 商场服务器 4](#_Toc299394811)

[2.3. 银行服务器 4](#_Toc299394812)

[3. 客户端设计说明 4](#_Toc299394813)

[3.1. 程序描述 4](#_Toc299394814)

[3.2. 功能 4](#_Toc299394815)

[3.3. 性能 4](#_Toc299394816)

[3.4. 输人项 4](#_Toc299394817)

[3.5. 输出项 5](#_Toc299394818)

[3.6. 算法 5](#_Toc299394819)

[3.7. 流程逻辑 5](#_Toc299394820)

[3.8. 接口 5](#_Toc299394821)

[3.9. 存储分配 5](#_Toc299394822)

[3.10. 注释设计 5](#_Toc299394823)

[3.11. 限制条件 5](#_Toc299394824)

[3.12. 测试计划 5](#_Toc299394825)

[3.13. 尚未解决的问题 6](#_Toc299394826)

[4. 商场服务器端设计说明 6](#_Toc299394827)

[5. 银行服务器端设计说明 6](#_Toc299394828)

# 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

JQUERY

CLIENT

JSP

JSON DATA

Module

DASHBOARD

AJAX INTERFACE

DAO

ENTITY

Database

SECURITY

MODULE

NET TRANSMISSION CONTROL

VERIFY

TRANSACTION

URL

KEY

# 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

1. 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”

# 服务器端设计说明

## 程序描述

iPay的商场服务端处理程序

## 功能

## 性能

* 服务端应能够通过管理GUI页面进行方便的操作，进行数据库的操作，与客户端的交互等。
* 对于来自客户端的数据请求等，服务端应能够在可以忍受的时间内给出相应，初步调查为返回数据时间小于20s，返回操作结果（即支付等操作）为小于1min。

## 输入项

* 服务端管理员的操作：主要是后台管理的一些操作。对数据库的增、删、查、改，记录查看商品信息，购物信息等。
* 来自客户端的操作：来自客户端主要分为两类操作，查看数据以及递交操作。向服务端发送请求，一是查看相关数据，二是发送修改，确认等操作请求。

## 输出项

* 对于服务端管理员的操作：对应于输入操作，输出对应的结果。对数据库的修改，会给出最终的结果。
* 对于来自客户端的操作：给出相应的结果。对于查看数据等请求，服务端返回需要的数据，对于修改，确认等，服务端进行操作后，返回得到的结果，如扫描条形码，支付等。并返回到客户端。

## 算法

## 逻辑流程

## 接口

## 存储分配

## 注释设计

## 限制条件

本服务端需要如下的条件

1. 良好的网络访问。

## 测试计划

## 尚未解决的问题

## 尚未解决的问题