详细设计说明书

Revision History:

|  |  |  |
| --- | --- | --- |
| **Who** | **When** | **What** |
| 1.0 | 2018/2/6 | Draft |

目录

[详细设计说明书 1](#_Toc35685103)

[1 引言 4](#_Toc35685104)

[1.1 编写目的 4](#_Toc35685105)

[1.2 背景 4](#_Toc35685106)

[1.3 定义 4](#_Toc35685107)

[1.4 参考资料 4](#_Toc35685108)

[2 程序系统的组织结构 5](#_Toc35685109)

[3 关键类设计 7](#_Toc35685110)

[3.1 nbr测试代码体系类图 8](#_Toc35685111)

[4 模块实现：同步 8](#_Toc35685112)

[4.1 程序描述 9](#_Toc35685113)

[4.2 功能 9](#_Toc35685114)

[4.3 性能 9](#_Toc35685115)

[4.4 输入项 9](#_Toc35685116)

[4.5 输出项 9](#_Toc35685117)

[4.6 算法 9](#_Toc35685118)

[4.7 流程逻辑 10](#_Toc35685119)

[4.8 接口 10](#_Toc35685120)

[4.9 存储分配 11](#_Toc35685121)

[4.10 注释设计 11](#_Toc35685122)

[4.11 限制条件 11](#_Toc35685123)

[4.12 测试计划 11](#_Toc35685124)

[4.12.1 单元测试计划 11](#_Toc35685125)

[4.12.2 自动化测试计划 11](#_Toc35685126)

[4.12.3 集成测试方案 11](#_Toc35685127)

[4.13 尚未解决的问题 12](#_Toc35685128)

[5 NBR支付流程实现 12](#_Toc35685129)

[5.1 程序描述 12](#_Toc35685130)

[5.2 功能 12](#_Toc35685131)

[5.3 性能 12](#_Toc35685132)

[5.4 输入项 13](#_Toc35685133)

[5.5 输出项 13](#_Toc35685134)

[5.6 算法 13](#_Toc35685135)

[5.7 流程逻辑 13](#_Toc35685136)

[5.8 接口 13](#_Toc35685137)

[5.9 存储分配 13](#_Toc35685138)

[5.10 注释设计 13](#_Toc35685139)

[5.11 限制条件 14](#_Toc35685140)

[5.12 测试计划 14](#_Toc35685141)

[5.12.1 单元测试计划 14](#_Toc35685142)

[5.12.2 自动化测试计划 14](#_Toc35685143)

[5.12.3 集成测试方案 14](#_Toc35685144)

[6 Pos支付流程实现 16](#_Toc35685145)

# 引言

## 编写目的

本文档的主要目的：

* 指导开发
* 指导测试
* 估计工作量和工期
* 后续版本迭代的依据

## 背景

## 定义

## 参考资料

《概要设计说明书》

# 程序系统的组织结构

包

类

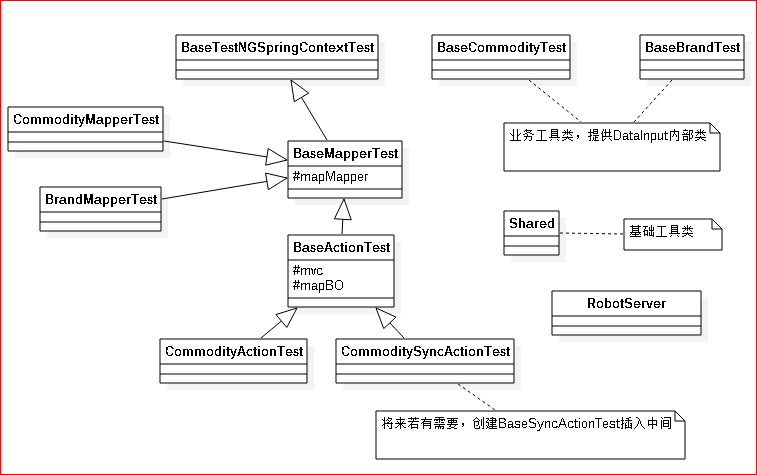
源文件

JSP

SP

# 关键类设计

## nbr测试代码体系类图



# 模块实现：同步

## 程序描述

## 功能

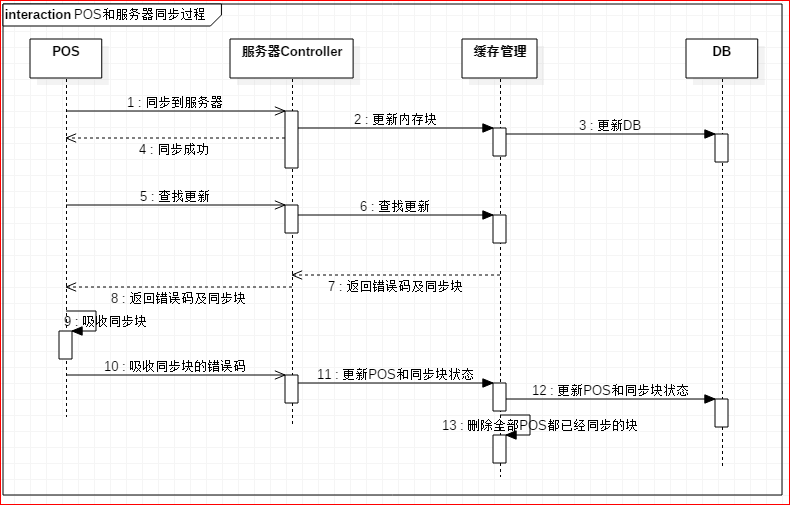
## 性能

## 输入项

## 输出项

## 算法

## 流程逻辑



## 接口

## 存储分配

## 注释设计

## 限制条件

## 测试计划

### 单元测试计划

### 自动化测试计划

### 集成测试方案

## 尚未解决的问题

# NBR支付流程实现

## 程序描述

## 功能

## 性能

## 输入项

## 输出项

## 算法

## 流程逻辑

## 接口

## 存储分配

## 注释设计

## 限制条件

## 测试计划

### 单元测试计划

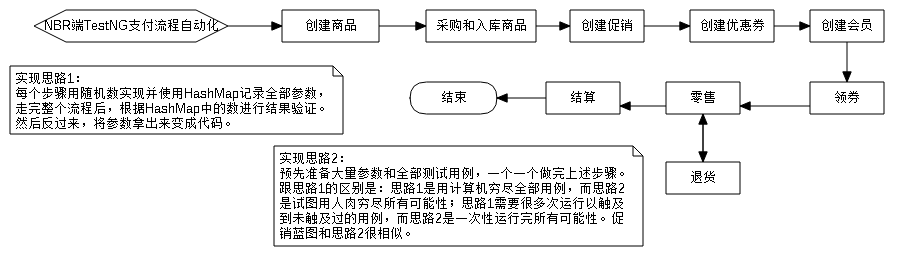
### 自动化测试计划

### 集成测试方案

#### 可变因子

促销、优惠券、是否Vip、时间、张数、优惠券种类、商品种类、是否已经核销、是否中促销/优惠券范围？

#### 流程和实现思路



采用实现思路1，基于以下原因：

1. 几乎不可能穷尽所有测试用例。
2. 人手不足。即使有人手，也要花费大量的精力，比如促销计算蓝图那边，消耗了大量人力，但是也无法穷尽。
3. 实现思路1，不等于放弃思路2。将来可以就发现的Bug，按思路2去实现去弥补。

#### 具体实现

1. 实现流程中的每个步骤，每个细节都采用随机数，并且用HashMap或其它数据结构，存储每个步骤的输入和输出。
2. 像实现检查点一样，在流程结束后，通过读取数据结构中的数据，进行结果验证。一旦发现运行过程中出现结果验证失败，则打印出现场，进行修复，可能需要用思路2弥补漏洞。

#### 运行计划

理论上足够长时间的运行，可以覆盖所有用例，所以本测试要独立为一个TestNG suite，放testngSIT\_Pay.xml中，新建Jenkins任务nbr\_dev\_runSITTest\_Pay运行。

# Pos支付流程实现