



**趣袋**

**Fun Bags**

**项目测试文档**

**Version [1.0.0]**

**2020.7.28**

**Written by 蓝色一袋**



**All Rights Reserved**

# 目录

<b>1</b>	<b>测试计划.....</b>	<b>1</b>
1.1	测试策略与目标.....	1
1.1.1	测试策略.....	1
1.1.2	测试用例与执行分析.....	1
1.1.3	测试难点.....	1
1.1.4	测试目标.....	1
1.2	测试范围.....	2
1.3	测试环境.....	2
1.3.1	软件环境.....	2
1.3.2	硬件环境.....	2
<b>2</b>	<b>单元测试.....</b>	<b>3</b>
2.1	订单管理模块.....	3
2.1.1	获取订单列表.....	3
2.1.2	获取订单详细信息.....	6
2.1.3	结束订单.....	7
2.2	机器管理模块.....	7
2.2.1	获取机器列表.....	7
2.2.2	添加机器信息.....	10
2.2.3	修改机器信息.....	10
2.2.4	获取机器详情.....	10
2.2.5	获取机器列表.....	11
2.3	用户管理模块.....	11
2.3.1	获取用户列表.....	11
2.3.2	添加用户信息.....	14
2.3.3	修改用户信息.....	15
2.3.4	查看用户详情.....	15
2.3.5	移除用户.....	15
2.4	商户管理模块.....	16
2.4.1	获取商户列表.....	16
2.4.2	添加商户.....	21
2.4.3	移除商户.....	21
2.5	角色管理模块（获取角色列表）.....	22
2.6	机器位置分布管理模块.....	25
2.6.1	获取机器位置列表.....	25
2.6.2	按地区查看机器位置分布.....	27
2.6.1	按状态查看机器位置分布.....	27
2.7	员工管理模块.....	28
2.7.1	获取员工列表.....	28
2.7.2	添加员工信息.....	30
2.7.3	修改员工信息.....	31
2.7.4	查看员工详细信息.....	31

2.7.5	移除员工.....	31
2.8	测试结果综合分析及建议.....	32
2.9	测试经验总结.....	32
<b>3</b>	<b>功能测试.....</b>	<b>32</b>
3.1	****功能.....	32
3.1.1	测试用例与执行分析.....	32
	测试用例: .....	32
	分析结果: .....	32
3.1.2	测试结果综合分析及建议.....	33
3.1.3	测试经验总结.....	33
<b>4</b>	<b>系统测试.....</b>	<b>33</b>
4.1	模型性能测试.....	33
4.1.1	测试用例与执行分析.....	33
	预期性能指标测试用例: .....	33
	分析结果: .....	33
4.1.2	测试结果综合分析及建议.....	33
4.1.3	测试经验总结.....	33

## 记录更改历史

[illegible]

# 1 测试计划

## 1.1 测试策略与目标

### 1.1.1 测试策略

采取黑盒测试，根据功能模块进行测试单元分割，对不同的功能模块采取不同的测试技术。

### 1.1.2 测试用例与执行分析

- A. 测试快应用是否正确运行
- B. 测试快应用能否对二维码进行准确识别，向服务器发送订单信息，并正确返回提示信息。
- C. 测试 MySQL 数据库是否正确搭建，能否被服务器访问。
- D. 测试 PHP 服务器能否正确连接 Mysql 数据库并对数据进行函数调用。
- E. 测试 PHP 服务器能否正确接收快应用端发送的订单请求，并对 MySQL 数据库中的订单信息进行添加和修改。
- F. 测试 PHP 服务器能否正确向吐袋机发送吐袋信号。
- G. 测试 Unity 模拟的吐袋机能否正确访问服务器，获取订单信息。
- H. 测试 Unity 模拟的吐袋机能否正确获取并判断服务器上的吐袋状态标记，并完成吐袋动作（吐袋三维动画）。
- I. 测试 CMS 后台管理系统能否正确通过 PHP 服务器对后台 MySQL 数据库中的数据进行读取显示、调用和修改。
- J. 测试 CMS 后台管理系统能否根据用户角色、权限的不同，显示对应不同的数据页面。
- K. 测试 CMS 后台管理系统的上传图片功能能否正确运行。
- L. 对 PHP 服务器与 CMS 后台管理系统进行运行压力测试。

### 1.1.3 测试难点

- A. 测试 PHP 服务器能否正确接收快应用端发送的订单请求，并对 MySQL 数据库中的订单信息进行添加和修改。
- B. 测试吐袋机能否正确接收服务器端发送的吐袋执行信号。
- C. 测试吐袋机能否在限定响应时间内完成吐袋动作，并回复到预吐袋工作状态。（现阶段暂用 Unity 模拟吐袋机的搭建和吐袋动作）

### 1.1.4 测试目标

- A. 初赛阶段：
  - a. 快应用能否对二维码进行准确识别，向服务器发送订单信息。
  - b. PHP 服务器能否正确连接 MySQL 数据库。（MySQL 数据库的搭建暂用 easy-mock 代替模拟）

- c. Unity 能否正确模拟真实吐袋机完成吐袋动作。
- d. CMS 后台管理系统能否正确通过服务器调取显示 MySQL 数据库中数据。(数据暂用 easy-mock 模拟生成)

B. 复赛阶段:

测试在使用 Unity 模拟的情况下, 能否完成由用户使用快应用扫码发送订单请求, 到服务器接收订单请求对数据库中的订单信息进行修改, 再到吐袋机接收服务器端的吐袋请求完成吐袋动作, 到最后 CMS 后台管理系统能显示订单信息的整个完整流程。

## 1.2 测试范围

A. 测试对象

- a. 程序
- b. 相关文件 (需求文档、使用手册)

B. 需要测试的产品功能

- a. 产品使用功能
- b. 产品资料库

C. 测试覆盖

- a. 功能 (用例覆盖)
- b. 输入域覆盖
- c. 输出域覆盖

## 1.3 测试环境

### 1.3.1 软件环境

- A. Postman 7.14.0(Web API & HTTP 请求调试工具)
- B. Easy-Mock (模拟数据平台)
- C. Chrome 浏览器
- D. Window10 X64

### 1.3.2 硬件环境

- A. 处理器:i5-7300
- B. 内存:12GB
- C. 显示适配器:GTX 1050

## 2 单元测试

### 2.1 订单管理模块

#### 2.1.1 获取订单列表

用例编号	001
接口名称	getOrderList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 OrderList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的订单数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```
"item_list": [  
  {  
    "id": 2959165,  
    "order_id": "O201947884",  
    "machine_id": "M106421",  
    "user_id": "C602927",  
    "user_name": "汤勇",  
    "shop_id": "S023171",  
    "shop_name": "商户 4",  
    "order_time": "2017-01-14 19:43:30",  
    "machine_type": "小型机",  
    "machine_status": 0,  
    "order_fee": 1.5  
  },  
  {  
    "id": 2959166,  
    "order_id": "O201927755",  
    "machine_id": "M102283",  
    "user_id": "C602058",  
    "user_name": "武平",  
    "shop_id": "S024319",  
    "shop_name": "商户 4",  
    "order_time": "2010-09-12 10:23:19",  
    "machine_type": "小型机",  
    "machine_status": 1,  
  }  
]
```

```
"order_fee": 1.5
},
{
  "id": 2959167,
  "order_id": "O201908446",
  "machine_id": "M106636",
  "user_id": "C601538",
  "user_name": "贾丽",
  "shop_id": "S025784",
  "shop_name": "商户 1",
  "order_time": "2007-06-03 19:34:31",
  "machine_type": "大型机",
  "machine_status": 0,
  "order_fee": 1.5
},
{
  "id": 2959168,
  "order_id": "O201972475",
  "machine_id": "M102361",
  "user_id": "C604782",
  "user_name": "毛杰",
  "shop_id": "S029535",
  "shop_name": "商户 2",
  "order_time": "1975-10-10 00:17:35",
  "machine_type": "小型机",
  "machine_status": 1,
  "order_fee": 1.5
},
{
  "id": 2959169,
  "order_id": "O201932310",
  "machine_id": "M102383",
  "user_id": "C608345",
  "user_name": "邓艳",
  "shop_id": "S022450",
  "shop_name": "商户 2",
  "order_time": "2004-06-23 11:01:01",
  "machine_type": "小型机",
  "machine_status": 2,
  "order_fee": 1.5
},
{
  "id": 2959170,
  "order_id": "O201968353",
```



```
"machine_id": "M102664",
"user_id": "C605541",
"user_name": "秦秀英",
"shop_id": "S026354",
"shop_name": "商户 4",
"order_time": "2010-02-11 22:58:04",
"machine_type": "小型机",
"machine_status": 0,
"order_fee": 1.5
},
{
  "id": 2959171,
  "order_id": "O201959236",
  "machine_id": "M107732",
  "user_id": "C607767",
  "user_name": "江刚",
  "shop_id": "S025566",
  "shop_name": "商户 3",
  "order_time": "1980-03-30 19:35:59",
  "machine_type": "大型机",
  "machine_status": 1,
  "order_fee": 1.5
},
{
  "id": 2959172,
  "order_id": "O201918104",
  "machine_id": "M105766",
  "user_id": "C609553",
  "user_name": "程秀兰",
  "shop_id": "S025216",
  "shop_name": "商户 2",
  "order_time": "1974-05-20 23:51:55",
  "machine_type": "大型机",
  "machine_status": 1,
  "order_fee": 1.5
},
{
  "id": 2959173,
  "order_id": "O201922358",
  "machine_id": "M105822",
  "user_id": "C603325",
  "user_name": "石刚",
  "shop_id": "S023018",
  "shop_name": "商户 4",
```

```

"order_time": "1972-05-02 03:52:55",
"machine_type": "小型机",
"machine_status": 2,
"order_fee": 1.5
},
{
"id": 2959174,
"order_id": "O201984432",
"machine_id": "M103337",
"user_id": "C608554",
"user_name": "秦强",
"shop_id": "S028383",
"shop_name": "商户 2",
"order_time": "2003-01-22 12:32:12",
"machine_type": "小型机",
"machine_status": 0,
"order_fee": 1.5
}
]

```

### 2.1.2 获取订单详细信息

用例编号	002
接口名称	getOrderDetail
参数	order_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 OrderDetail 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中某一订单的详细信息
前提条件	管理人员登录后台管理系统,并选中其中一条订单
特殊的规程说明	无
用例间的依赖关系	<pre> graph LR     A([get0rderDetail]) -- "&lt;&lt;uses&gt;&gt;" --&gt; B([get0rderList]) </pre>

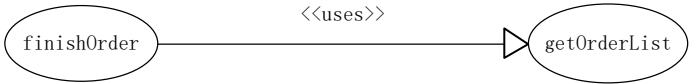
返回结果:

```

"status": 2,
"order_sn": "O201957809",
"bike_sn": "M105835",
"mode": 2,
"location": "北京市工业大学奥运餐厅",
"owner": "商户 1",
"position": "116.488551,39.879605"

```

### 2.1.3 结束订单

用例编号	003
接口名称	finishOrder
参数	order_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 OrderList 接口,查看被结束的订单信息是否存在
用例目的	在后台管理系统删除服务器中某一订单的信息
前提条件	管理人员登录后台管理系统,并选中其中一条订单
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     A((finishOrder)) -- "&lt;&lt;uses&gt;&gt;" --&gt; B(((getOrderList))) </pre>

## 2.2 机器管理模块

### 2.2.1 获取机器列表

用例编号	004
接口名称	getMachineList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 MachineList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的机器数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```


"item_list": [
{
"id": "M105216",
"machine_type": "小型机",
"status": 1,
"service_time": "1978-08-14",
"owner": "商户 3",
"address": "北京市海淀区",
"remain_num": 68,
"usetime": 50
},
{
"id": "M103508",
"machine_type": "小型机",

```


```
"status": 0,
"service_time": "1983-01-06",
"owner": "商户 2",
"address": "北京市石景山区",
"remain_num": 211,
"usetime": 66
},
{
  "id": "M106575",
  "machine_type": "大型机",
  "status": 1,
  "service_time": "1973-05-30",
  "owner": "商户 3",
  "address": "北京市丰台区",
  "remain_num": 139,
  "usetime": 232
},
{
  "id": "M107122",
  "machine_type": "大型机",
  "status": 3,
  "service_time": "1989-09-09",
  "owner": "商户 1",
  "address": "北京市昌平区",
  "remain_num": 179,
  "usetime": 72
},
{
  "id": "M108250",
  "machine_type": "小型机",
  "status": 2,
  "service_time": "2004-01-15",
  "owner": "商户 3",
  "address": "北京市石景山区",
  "remain_num": 117,
  "usetime": 336
},
{
  "id": "M105175",
  "machine_type": "大型机",
  "status": 0,
  "service_time": "1998-06-27",
  "owner": "商户 3",
  "address": "北京市通州区",
```

```
"remain_num": 85,
"usetime": 280
},
{
"id": "M106394",
"machine_type": "小型机",
"status": 1,
"service_time": "1995-07-08",
"owner": "商户 2",
"address": "北京市朝阳区",
"remain_num": 206,
"usetime": 31
},
{
"id": "M109763",
"machine_type": "大型机",
"status": 3,
"service_time": "1974-10-26",
"owner": "商户 3",
"address": "北京市西城区",
"remain_num": 231,
"usetime": 52
},
{
"id": "M105446",
"machine_type": "小型机",
"status": 1,
"service_time": "1987-07-28",
"owner": "商户 2",
"address": "北京市大兴区",
"remain_num": 153,
"usetime": 265
},
{
"id": "M102782",
"machine_type": "大型机",
"status": 0,
"service_time": "1978-12-18",
"owner": "商户 1",
"address": "北京市东城区",
"remain_num": 218,
"usetime": 136
}
]
```

### 2.2.2 添加机器信息

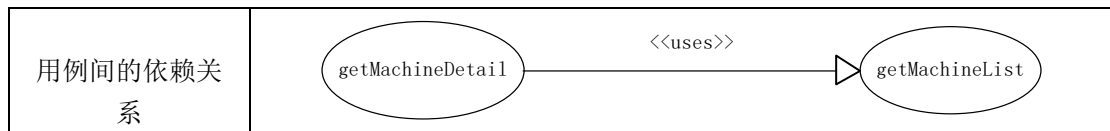
用例编号	005
接口名称	addMachine
参数	machine_type, service_time, machine_owner, address, remain_num
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 MachineList 接口,查看已添加的机器信息是否存在
用例目的	通过后台管理系统, 向服务器添加新的机器信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     addMachine((addMachine)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getMachineList((getMachineList))           </pre>

### 2.2.3 修改机器信息

用例编号	006
接口名称	modifyMachine
参数	Machine_id, machine_type, service_time, machine_owner, address, remain_num
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 MachineList 接口,查看已修改的机器信息是否正确
用例目的	在后台管理系统修改服务器中某一机器的信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     modifyMachine((modifyMachine)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getMachineList((getMachineList))           </pre>

### 2.2.4 获取机器详情

用例编号	007
接口名称	getMachineDetail
参数	machine_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 MachineDetail 接口,查看返回的机器详细信息是否正确
用例目的	在后台管理系统获取服务器中某一机器的详细数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无



### 2.2.5 获取机器列表

用例编号	008
接口名称	deleteMachine
参数	machine_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 MachineList 接口,查看返回的机器列表信息是否正确
用例目的	在后台管理系统移除服务器中某一机器的数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	<pre> graph LR     A([deleteMachine]) -- "&lt;&lt;uses&gt;&gt;" --&gt; B([getMachineList])           </pre>

## 2.3 用户管理模块

### 2.3.1 获取用户列表

用例编号	009
接口名称	getUserList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 UserList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的用户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```

"item_list": [
{
"id": "C607881",
"username": "于娟",
"sex": 2,
"phone": "15879553087",
"birthday": "1979-12-10",
"age": 2019,

```

```
"address": "北京市大兴区",
"buildtime": "1979-05-22",
"usetime": 77,
"interest": 3,
"totaluse": 76
},
{
  "id": "C602653",
  "username": "蔡明",
  "sex": 1,
  "phone": "12525026571",
  "birthday": "1998-05-30",
  "age": 2019,
  "address": "北京市怀柔区",
  "buildtime": "2003-11-10",
  "usetime": 257,
  "interest": 2,
  "totaluse": 96
},
{
  "id": "C604874",
  "username": "卢娜",
  "sex": 1,
  "phone": "16742116875",
  "birthday": "2011-07-03",
  "age": 2019,
  "address": "北京市顺义区",
  "buildtime": "1974-01-10",
  "usetime": 52,
  "interest": 5,
  "totaluse": 44
},
{
  "id": "C609888",
  "username": "孙杰",
  "sex": 2,
  "phone": "15183513588",
  "birthday": "1977-07-25",
  "age": 2019,
  "address": "北京市延庆区",
  "buildtime": "2017-10-04",
  "usetime": 56,
  "interest": 8,
  "totaluse": 36
```

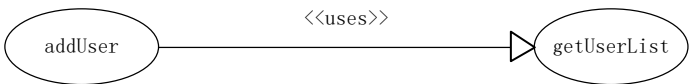


```
},
{
  "id": "C605858",
  "username": "袁秀兰",
  "sex": 2,
  "phone": "18368286111",
  "birthday": "2019-03-27",
  "age": 2019,
  "address": "北京市昌平区",
  "buildtime": "2019-09-16",
  "usetime": 192,
  "interest": 2,
  "totaluse": 36
},
{
  "id": "C602184",
  "username": "丁丽",
  "sex": 2,
  "phone": "17281290872",
  "birthday": "1991-11-07",
  "age": 2019,
  "address": "北京市石景山区",
  "buildtime": "1994-02-22",
  "usetime": 230,
  "interest": 2,
  "totaluse": 51
},
{
  "id": "C600355",
  "username": "钱洋",
  "sex": 1,
  "phone": "17642172142",
  "birthday": "1992-04-08",
  "age": 2019,
  "address": "北京市门头沟区",
  "buildtime": "2000-11-22",
  "usetime": 365,
  "interest": 3,
  "totaluse": 93
},
{
  "id": "C603764",
  "username": "易秀英",
  "sex": 2,
```

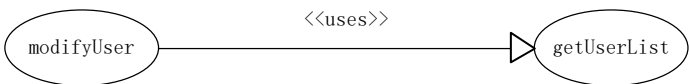
```
"phone": "12415882565",
"birthday": "1977-04-29",
"age": 2019,
"address": "北京市丰台区",
"buildtime": "2018-01-09",
"usetime": 16,
"interest": 8,
"totaluse": 57
},
{
  "id": "C605453",
  "username": "万强",
  "sex": 2,
  "phone": "11261751315",
  "birthday": "2001-12-15",
  "age": 2019,
  "address": "北京市昌平区",
  "buildtime": "2008-01-05",
  "usetime": 5,
  "interest": 3,
  "totaluse": 47
},
{
  "id": "C606218",
  "username": "范秀英",
  "sex": 2,
  "phone": "14637427375",
  "birthday": "2008-10-17",
  "age": 2019,
  "address": "北京市密云区",
  "buildtime": "2003-07-13",
  "usetime": 162,
  "interest": 7,
  "totaluse": 36
}
]
```

### 2.3.2 添加用户信息

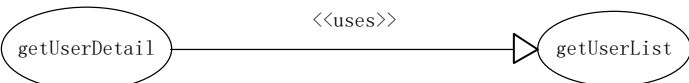
用例编号	010
接口名称	addUser
参数	user_name, user_sex, user_phone, user_birthday, user_age, user_address, user_buildname, user_interest, user_totaluse

测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 UserList 接口,查看已添加的用户信息是否正确。
用例目的	通过后台管理系统, 向服务器添加用户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     addUser((addUser)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getUserList((getUserList))           </pre>

### 2.3.3 修改用户信息

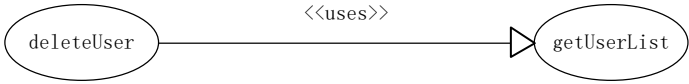
用例编号	011
接口名称	modifyUser
参数	user_name, user_sex, user_phone, user_birthday, user_age, user_address, user_buildname, user_interest, user_totaluse
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 UserList 接口,查看已修改的用户信息是否正确。
用例目的	通过后台管理系统, 修改服务器中某一用户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     modifyUser((modifyUser)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getUserList((getUserList))           </pre>

### 2.3.4 查看用户详情

用例编号	012
接口名称	getUserDetail
参数	user_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 UserDetail 接口,查看返回的用户详细信息是否正确。
用例目的	在后台管理系统获取服务器中某一用户的详细数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     getUserDetail((getUserDetail)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getUserList((getUserList))           </pre>

### 2.3.5 移除用户

用例编号	013
------	-----

接口名称	deleteUser
参数	user_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 UserList 接口,查看返回的用户信息是否正确。
用例目的	通过后台管理系统, 移除服务器中某一用户的数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     deleteUser((deleteUser)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getUserList((getUserList)) </pre>

## 2.4 商户管理模块

### 2.4.1 获取商户列表

用例编号	014
接口名称	getShopList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 ShopList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的商户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```

"item_list": [
{
"id": "S021637",
"name": "商户 3",
"location": "北京市通州区",
"machine_num": 4,
"order_num": 2561,
"mode": 1,
"op_mode": 2,
"franchisee_id": 77,
"franchisee_name": "松果自营",
"city_admins": [
{
"user_name": "韩磊",
"user_id": 10001
},

```

```
{
  "user_name": "戴芳",
  "user_id": 10002
},
{
  "open_time": "1970-04-28 10:25:37",
  "sys_user_name": "廖磊",
  "update_time": 1520476737000
},
{
  "id": "S027606",
  "name": "商户 3",
  "location": "北京市西城区",
  "machine_num": 5,
  "order_num": 3272,
  "mode": 1,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "吴磊",
      "user_id": 10003
    }
  ],
  "open_time": "1984-06-17 12:37:43",
  "sys_user_name": "杜敏",
  "update_time": 1520476737000
},
{
  "id": "S028878",
  "name": "商户 2",
  "location": "北京市大兴区",
  "machine_num": 4,
  "order_num": 4308,
  "mode": 1,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "易霞",
      "user_id": 10004
    }
  ],
}
```

```
{
  "user_name": "唐平",
  "user_id": 10005
},
{
  "open_time": "2009-11-29 15:21:00",
  "sys_user_name": "汤平",
  "update_time": 1520476737000
},
{
  "id": "S029473",
  "name": "商户 3",
  "location": "北京市延庆区",
  "machine_num": 6,
  "order_num": 5763,
  "mode": 2,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "杜勇",
      "user_id": 10006
    }
  ],
  "open_time": "1993-04-14 05:05:46",
  "sys_user_name": "黄丽",
  "update_time": 1520476737000
},
{
  "id": "S022562",
  "name": "商户 2",
  "location": "北京市丰台区",
  "machine_num": 4,
  "order_num": 5328,
  "mode": 1,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "武秀兰",
      "user_id": 10007
    }
  ]
}
```

```
],
"open_time": "1977-09-03 23:30:52",
"sys_user_name": "姚磊",
"update_time": 1520476737000
},
{
"id": "S026868",
"name": "商户 1",
"location": "北京市延庆区",
"machine_num": 1,
"order_num": 6418,
"mode": 1,
"op_mode": 2,
"franchisee_id": 77,
"franchisee_name": "松果自营",
"city_admins": [
{
"user_name": "傅霞",
"user_id": 10008
}
],
"open_time": "2009-05-31 18:04:44",
"sys_user_name": "潘娟",
"update_time": 1520476737000
},
{
"id": "S024456",
"name": "商户 1",
"location": "北京市通州区",
"machine_num": 1,
"order_num": 3112,
"mode": 2,
"op_mode": 1,
"franchisee_id": 77,
"franchisee_name": "松果自营",
"city_admins": [
{
"user_name": "程丽",
"user_id": 10009
}
],
"open_time": "1990-11-22 13:52:46",
"sys_user_name": "尹洋",
"update_time": 1520476737000
```

```
},
{
  "id": "S023461",
  "name": "商户 1",
  "location": "北京市昌平区",
  "machine_num": 2,
  "order_num": 6005,
  "mode": 2,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "常杰",
      "user_id": 10010
    }
  ],
  "open_time": "2003-03-19 05:30:18",
  "sys_user_name": "易勇",
  "update_time": 1520476737000
},
{
  "id": "S021247",
  "name": "商户 1",
  "location": "北京市通州区",
  "machine_num": 2,
  "order_num": 4448,
  "mode": 2,
  "op_mode": 1,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "尹平",
      "user_id": 10011
    },
    {
      "user_name": "谭静",
      "user_id": 10012
    }
  ],
  "open_time": "2013-08-18 00:51:17",
  "sys_user_name": "苏秀英",
  "update_time": 1520476737000
}
```

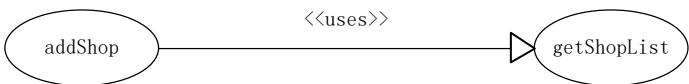


```

},
{
  "id": "S025596",
  "name": "商户 2",
  "location": "北京市顺义区",
  "machine_num": 4,
  "order_num": 5818,
  "mode": 2,
  "op_mode": 2,
  "franchisee_id": 77,
  "franchisee_name": "松果自营",
  "city_admins": [
    {
      "user_name": "陆刚",
      "user_id": 10013
    }
  ],
  "open_time": "1995-09-14 03:59:30",
  "sys_user_name": "叶强",
  "update_time": 1520476737000
}
]

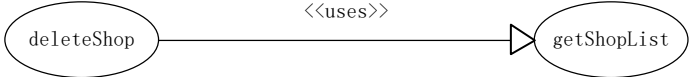
```

### 2.4.2 添加商户

用例编号	015
接口名称	addShop
参数	shop_id, shop_name, shop_location, machine_num, order_num, open_time, shop_admins
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 ShopList 接口,查看已添加的商户信息是否存在
用例目的	通过后台管理系统,向服务器添加新的商户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     addShop((addShop)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getShopList((getShopList)) </pre>

### 2.4.3 移除商户

用例编号	016
接口名称	deleteShop
参数	shop_id

测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 ShopList 接口,查看返回的商户数据信息是否正确
用例目的	通过后台管理系统,删除服务器中某一商户数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     deleteShop([deleteShop]) -- "&lt;&lt;uses&gt;&gt;" --&gt; getShopList([getShopList]) </pre>

## 2.5 角色管理模块（获取角色列表）

用例编号	017
接口名称	getRoleList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 RoleList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的角色数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```

"item_list": [
{
"id": 1,
"role_name": "普通用户",
"status": 0,
"authorize_user_name": "田洋",
"authorize_time": 1521270166000,
"create_time": 1499305790000,
"menus": [
"/home",
"/ui/buttons",
"/ui/modals",
"/ui/loadings",
"/ui/notification",
"/ui/messages",
"/ui/tabs",
"/ui/gallery",
"/ui/carousel",
"/ui"
]
}
]

```

```
},
{
  "id": 2,
  "role_name": "普通员工",
  "status": 0,
  "authorize_user_name": "孙涛",
  "authorize_time": 1521270166000,
  "create_time": 1499305790000,
  "menus": [
    "/home",
    "/ui/buttons",
    "/ui/modals",
    "/ui/loadings",
    "/ui/notification",
    "/ui/messages",
    "/ui/tabs",
    "/ui/gallery",
    "/ui/carousel",
    "/ui"
  ]
},
{
  "id": 3,
  "role_name": "普通员工",
  "status": 0,
  "authorize_user_name": "曹超",
  "authorize_time": 1521270166000,
  "create_time": 1499305790000,
  "menus": [
    "/home",
    "/ui/buttons",
    "/ui/modals",
    "/ui/loadings",
    "/ui/notification",
    "/ui/messages",
    "/ui/tabs",
    "/ui/gallery",
    "/ui/carousel",
    "/ui"
  ]
},
{
  "id": 4,
  "role_name": "商户负责人",
```

```
"status": 0,
"authorize_user_name": "吴勇",
"authorize_time": 1521270166000,
"create_time": 1499305790000,
"menus": [
"/home",
"/ui/buttons",
"/ui/modals",
"/ui/loadings",
"/ui/notification",
"/ui/messages",
"/ui/tabs",
"/ui/gallery",
"/ui/carousel",
"/ui"
],
},
{
"id": 5,
"role_name": "普通用户",
"status": 0,
"authorize_user_name": "韩芳",
"authorize_time": 1521270166000,
"create_time": 1499305790000,
"menus": [
"/home",
"/ui/buttons",
"/ui/modals",
"/ui/loadings",
"/ui/notification",
"/ui/messages",
"/ui/tabs",
"/ui/gallery",
"/ui/carousel",
"/ui"
],
},
{
"id": 6,
"role_name": "普通用户",
"status": 1,
"authorize_user_name": "蔡伟",
"authorize_time": 1521270166000,
"create_time": 1499305790000,
```

```
"menus": [  
  "/home",  
  "/ui/buttons",  
  "/ui/modals",  
  "/ui/loadings",  
  "/ui/notification",  
  "/ui/messages",  
  "/ui/tabs",  
  "/ui/gallery",  
  "/ui/carousel",  
  "/ui"  
]  
},  
{  
  "id": 7,  
  "role_name": "普通员工",  
  "status": 0,  
  "authorize_user_name": "刘娟",  
  "authorize_time": 1521270166000,  
  "create_time": 1499305790000,  
  "menus": [  
    "/home",  
    "/ui/buttons",  
    "/ui/modals",  
    "/ui/loadings",  
    "/ui/notification",  
    "/ui/messages",  
    "/ui/tabs",  
    "/ui/gallery",  
    "/ui/carousel",  
    "/ui"  
  ]  
}  
]
```

## 2.6 机器位置分布管理模块

### 2.6.1 获取机器位置列表

用例编号	018
接口名称	getMachinePosition
参数	无

测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 Map_MachineList 接口, 查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的每台机器对应的经纬度信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

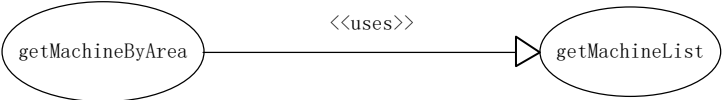
```
"bike_list": [
  "116.356619,40.017782",
  "116.437107,39.975331",
  "116.34972,40.070808",
  "116.323849,39.964714",
  "116.404912,40.015129",
  "116.365243,39.958078",
  "116.336429,39.99082",
  "116.253641,39.889473",
  "116.254216,39.974456",
  "116.378398,39.934633",
  "116.501429,39.886816",
  "116.501429,39.886816",
  "116.284686,39.855362",
  "116.644583,39.912943",
  "116.56352,39.972686",
  "116.227195,39.916485",
  "116.32723,39.822564",
  "116.32723,39.822564",
  "116.305599,39.968926",
  "116.365678,39.890332",
  "116.365678,39.890332",
  "116.299706,40.054113",
  "116.614185,39.889114",
  "116.325577,39.962152",
  "116.488314,39.881875",
  "116.488314,39.881875",
  "116.51282,39.89512",
  "116.480337,39.962622",
  "116.480337,39.962622",
  "116.466252,39.993363",
  "116.509371,39.887813",
  "116.436356,39.899327",
  "116.430895,39.986951",
  "116.702542,39.938506",
```

```


"116.702542,39.938506",
"116.502184,40.022101",
"116.406748,39.7923",
"116.389501,39.857245",
"116.467977,39.850155",
"116.425433,39.869208",
"116.425433,39.869208",
"116.51282,39.843508",
"116.494423,39.944038",
"116.474301,39.948463",
"116.566862,39.946693"
]

```

## 2.6.2 按地区查看机器位置分布

用例编号	019
接口名称	getMachineByArea
参数	machine_area
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 Map_MachineList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统, 查看某一地区的机器位置分布情况
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     A([getMachineByArea]) -- "&lt;&lt;uses&gt;&gt;" --&gt; B([getMachineList]) </pre>

## 2.6.1 按状态查看机器位置分布

用例编号	020
接口名称	getMachineByStatu
参数	machine_statu
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 Map_MachineList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统, 查看某一状态的机器位置分布情况
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     A([getMachineByStatu]) -- "&lt;&lt;uses&gt;&gt;" --&gt; B([getMachineList]) </pre>

## 2.7 员工管理模块

### 2.7.1 获取员工列表

用例编号	021
接口名称	getStaffList
参数	无
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 StaffList 接口,查看返回的数据信息是否正确。
用例目的	在后台管理系统获取服务器中的员工数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	无

返回结果:

```
"item_list": [  
  {  
    "id": "308638",  
    "username": "萧娜",  
    "sex": 2,  
    "state": 4,  
    "interest": 7,  
    "birthday": "2016-10-08",  
    "address": "北京市东城区",  
    "time": "09:00:00"  
  },  
  {  
    "id": "303826",  
    "username": "赵杰",  
    "sex": 2,  
    "state": 4,  
    "interest": 5,  
    "birthday": "1996-03-14",  
    "address": "北京市大兴区",  
    "time": "09:00:00"  
  },  
  {  
    "id": "300776",  
    "username": "汤明",  
    "sex": 2,  
    "state": 2,  
    "interest": 7,  
  }  
]
```

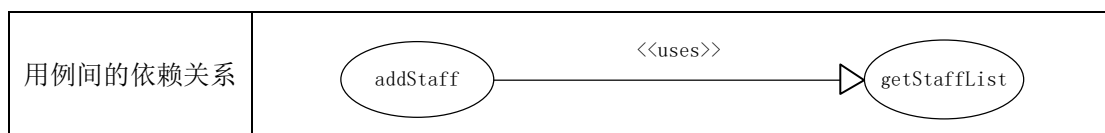


```
"birthday": "2017-01-03",
"address": "北京市延庆区",
"time": "09:00:00"
},
{
  "id": "304969",
  "username": "余芳",
  "sex": 1,
  "state": 3,
  "interest": 6,
  "birthday": "2001-07-29",
  "address": "北京市石景山区",
  "time": "09:00:00"
},
{
  "id": "306580",
  "username": "宋娜",
  "sex": 2,
  "state": 4,
  "interest": 1,
  "birthday": "1975-06-14",
  "address": "北京市房山区",
  "time": "09:00:00"
},
{
  "id": "302811",
  "username": "阎秀兰",
  "sex": 1,
  "state": 3,
  "interest": 3,
  "birthday": "1999-02-15",
  "address": "北京市通州区",
  "time": "09:00:00"
},
{
  "id": "303546",
  "username": "贺磊",
  "sex": 1,
  "state": 2,
  "interest": 6,
  "birthday": "1992-08-10",
  "address": "北京市房山区",
  "time": "09:00:00"
},
```

```
{
  "id": "306664",
  "username": "谢军",
  "sex": 1,
  "state": 4,
  "interest": 4,
  "birthday": "1976-12-14",
  "address": "北京市昌平区",
  "time": "09:00:00"
},
{
  "id": "306318",
  "username": "万秀兰",
  "sex": 2,
  "state": 2,
  "interest": 5,
  "birthday": "1982-09-08",
  "address": "北京市丰台区",
  "time": "09:00:00"
},
{
  "id": "306386",
  "username": "方平",
  "sex": 1,
  "state": 2,
  "interest": 2,
  "birthday": "1975-12-01",
  "address": "北京市大兴区",
  "time": "09:00:00"
}
]
```

### 2.7.2 添加员工信息

用例编号	022
接口名称	addStaff
参数	staff_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 StaffList 接口,查看已添加的员工数据信息是否正确。
用例目的	通过后台管理系统,向服务器添加的员工数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无



### 2.7.3 修改员工信息

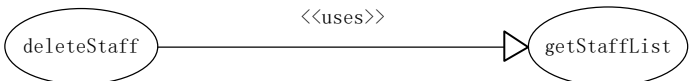
用例编号	023
接口名称	modifyStaff
参数	staff_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 StaffList 接口,查看已修改的员工数据信息是否正确。
用例目的	通过后台管理系统, 修改服务器中某一员工数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	<pre> graph LR     modifyStaff((modifyStaff)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getStaffList((getStaffList)) </pre>

### 2.7.4 查看员工详细信息

用例编号	024
接口名称	getStaffDetail
参数	staff_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 StaffDetail 接口,查看返回的员工详细信息是否正确。
用例目的	通过后台管理系统, 获取服务器中某一员工的详细数据信息
前提条件	管理人员登录后台管理系统
特殊的规程说明	无
用例间的依赖关系	<pre> graph LR     getStaffDetail((getStaffDetail)) -- "&lt;&lt;uses&gt;&gt;" --&gt; getStaffList((getStaffList)) </pre>

### 2.7.5 移除员工

用例编号	025
接口名称	deleteStaff
参数	staff_id
测试单元描述	通过 postman 的 GET 方法获取 easy-mock 上的 StaffList 接口,查看返回的员工信息列表是否正确。
用例目的	通过后台管理系统, 移除服务器中某一员工的数据信息
前提条件	管理人员登录后台管理系统

特殊的规程说明	无
用例间的依赖关系	 <pre> graph LR     deleteStaff([deleteStaff]) -- "&lt;&lt;uses&gt;&gt;" --&gt; getStaffList([getStaffList]) </pre>

## 2.8 测试结果综合分析及建议

从 postman 测试返回的数据基本无误，核心单元模块运行正常，依赖用例之间调配使用无碍，整体达到预期功能效果。

## 2.9 测试经验总结

要灵活运用各种辅助测试工具，针对某一功能，要划分好单元模块。

# 3 功能测试

## 3.1 \*\*\*\*功能

### 3.1.1 测试用例与执行分析

测试用例：

用例编号	001			
功能描述				
用例目的				
前提条件				
特殊的规程说明				
用例间的依赖关系				
步骤	动作	期望结果	实际结果	备注
1				
2				
...	...	...	...	...

分析结果：

### 3.1.2 测试结果综合分析及建议

### 3.1.3 测试经验总结

## 4 系统测试

### 4.1 模型性能测试

#### 4.1.1 测试用例与执行分析

预期性能指标测试用例：

用例编号	001			
性能描述				
用例目的				
前提条件				
特殊的规程说明				
用例间的依赖关系				
步骤	输入/动作	期望的性能 (平均值)	实际性能 (平均值)	备注
1				
2				
...	...	...	...	...

分析结果：

#### 4.1.2 测试结果综合分析及建议

#### 4.1.3 测试经验总结