**Stock Market Analysing System**

**测试文档**

****

**We are Scholar Of Four!**

|  |  |
| --- | --- |
| 姓名 | 学号 |
| 杨关 | 141250166 |
| 王泽霖 | 141250144 |
| 杨华安 | 141250167 |
| 熊凯奇 | 141250156 |

创建时间：2016/03/07

创建人：杨华安

更新历史：

更新时间：2016/06/9

更新人：熊凯奇

目录

[**1.单元测试计划** 2](#_Toc445312802)

[**2.单元测试用例** 4](#_Toc445312803)

[**Unit Test Case L1** 4](#_Toc445312804)

[**Unit Test Case L2** 5](#_Toc445312805)

[**Unit Test Case L3** 5](#_Toc445312806)

[**Unit Test Case L4** 6](#_Toc445312807)

[**3.单元测试报告** 6](#_Toc445312808)

**1.单元测试计划**

1. 单元测试计划概述

本文档为在“**Stock Market Analysing System**”进行单元测试时提供有关任务安排，方法，资源和进度方面的指导，目标是从代码中发现Bug，以备后续移除Bug参考，提高软件质量，最终达到用户需求。

1. 单元测试的范围

单元测试必须能够展示出“**Stock Market Analysing System**”中类的每个方法都能正常工作。单元测试的概念是将一个单元和该系统的其余部分隔离开来独立测试，因此可能需要创建测试驱动代码。

1. 单元测试的策略
2. 对方法的单元测试

* 检查对所有调用对象的使用
* 验证对所有数据结构的处理
* 验证对所有文件的处理
* 验证控制流的不变性
* 检查所有循环的正常终止
* 检查所有循环的异常终止
* 验证所有错误条件的处理
* 检查定时和同步

1. 对类的单元测试
2. 结合方法的执行，推荐对类的测试。

* 选择最常见的执行序列
* 需要包括可能导致缺陷的序列
* 手工计算预期的属性值

1. 集中对每个属性进行单元测试。

* 初始化，然后执行会影响属性的方法序列

1. 验证每个类的不变性。

* 验证默认初始值时的不变性。
* 执行方法序列。
* 验证不变性仍然正确。

**2.单元测试用例**

**Unit Test Case L1**

被测单元名称：getStockInfo

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

测试用例列表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| Test getStockInfo -001 | test getStockList1 | 市场代码“sha”,并选中其中第6个po的name | 输出”新疆天业“ | 基本符合预期 | Pass | 由于数据量较大且会自动更新只比较了任意选择的数据对象中认取的静态数据 |
| Test getStockInfo -002 | test getStockList2 | 市场代码“shb”，并选中其中第7个po的name | 输出“氯碱B股“ | 符合预期 | Pass | 由于数据量较大且会自动更新只比较了任意选择的数据对象中认取的静态数据，由于点击会限制输入，所以不考虑输入错误市场代码 |
| Test getStockInfo -003 | test getStockOne | 股票代码“000005”，并获得其中最后一个po的pre\_close | 输出大于0的浮点数 | 符合预期 | Pass | 由于数据量较大且会自动更新只比较了任意选择的数据对象中认取的数据且只要求大于0 |
| Test getStockInfo -004 | testgetNewStockList | 股票代码“000005”，并获得返回的po中的name | “世纪星源” | 符合预期 | Pass | 由于数据量较大且会自动更新只比较了任意选择的数据对象中认取的数据 |
| Test getStockInfo -005 | testgetCode1 | 股票代码“000009“ | ”000009“ | 符合预期 | Pass | 无 |
| Test getStockInfo -006 | testgetCode2 | 股票名称“保千里“ | ”600074“ | 符合预期 | Pass | 无 |
| Test getStockInfo -007 | testgetCode3 | 错误输入“589183“ | Null | 符合预期 | Pass | 无 |

测试类getStockInfoTest

**Unit Test Case L2**

被测单元名称：getMarketInfo

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

测试用例列表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestGetMarketInfo-001 | testGetMarketList | 选择获得的list中每一个的name | “上证指数”，”深证成指“，”沪深300“，”上证50“，”创业板“，”中小板“ | 符合预期 | Pass | 由于数据会自动更新只比较了其中的静态数据 |
| TestGetMarketInfo-002 | TestGetMarketOne | 市场代码“399001“，并选择其中第三个Po的open | 大于0 | 符合预期 | Pass | 数据会自动更新,只能判断大于0 |
| TestGetMarketInfo-003 | TestGetNewMarketList | 市场代码“399001“，并获得name | “深证成指“ | 符合预期 | Pass | 由于数据会自动更新只比较了其中的静态数据 |

测试类GetMarketInfoTest

**Unit Test Case L3**

被测单元名称：get\_p\_change

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

测试用例列表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestGetPchange-001 | testGetpchange1 | 6,9,0,3开头的股票代码各取一个,获得第三个po的market | 大于0 | 符合预期 | Pass | 因为是点击所以不存在股票代码不存在 |
| TestGetPchange-002 | testGetpchange2 | “490906” | Null | 符合预期 | Pass |  |

测试类GetPchangeTest

**Unit Test Case L4**

被测单元名称：getIndustry

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

测试用例列表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestGetIndustry-001 | testGetIndustry | 市场代码“sha”取获得List的长度 | 10 | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |

测试类GetIndustryTest

**Unit Test Case L5**

被测单元名称：getIndex

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestGetIndex-001 | testGetKDJ | 任意股票代码，取返回List的长度 | 30 | 基本符合预期 | Pass | 数据会随时间变化，所以只能c测试重要的静态数据，下同 |
| TestGetIndex-002 | testGetVOL | 任意股票代码，取返回List的长度 | 30 | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-003 | TestGetMACD | 任意股票代码，取返回List的长度 | 30 | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-004 | TestGetATR | 任意股票代码，取返回List的长度 | 30 | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |

测试类：GetIndexTest

**Unit Test Case L6**

被测单元名称：login

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestLogin-001 | testSignup1 | （保证数据库中用户表为空）两次调用signup | 第一次返回true，第二次返回false | 基本符合预期 | Pass | 无 |
| TestLogin-002 | testSignup2 | 输入长度大于12的string作为cid和password | Boolean值false | 基本符合预期 | Pass | 无 |
| TestLogin-003 | testLogin1 | 先调用signup注册一个正确格式的账户，而后登录 | Login返回true | 基本符合预期 | Pass | 无 |
| TestGetIndex-004 | Testlogin2 | 在Login中输入不符合格式的账号密码 | 返回false | 基本符合预期 | Pass | 无 |
| TestGetIndex-004 | TestGetOwnStock | 先注册正确用户名和密码，并调用raise方法存储，再输入注册的cid | 返回长度为1的List | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-005 | TestGetIndustry | 先注册正确用户名和密码，并调用raise方法存储，再输入注册的cid | 返回长度为10的浮点数数组 | 符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-006 | TestGetAll | 先注册正确用户名和密码，并调用raise方法存储至少三只自选股，再输入注册的cid | 返回的List里第一个PO的date为当天日期 | 符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-007 | TestGetOne | 先注册正确用户名和密码，并调用raise方法存储，再输入注册的cid | 返回的List里第一个PO的date为当天日期 | 符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-008 | TestDelete | 先注册正确用户名和密码，并调用raise方法存储2个po,在调用delete，再调用getOwnStock | 返回长度为1的List | 符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestGetIndex-009 | TestCheck1 | 先注册正确用户名和密码，并调用raise方法,check之前raise的cid和sid | 返回true | 符合预期 | Pass | 无 |
| TestGetIndex-010 | TestCheck2 | 先注册正确用户名和密码， check肯定不存在的Cid和sid | 返回false | 符合预期 | Pass | 无 |
| TestGetIndex-011 | TestGetOneStock | 先注册正确用户名和密码，然后raise若干po，使用注册的id | 返回长度为2且的字符串 | 符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |

测试类：logintest

**Unit Test Case L7**

被测单元名称：predict

用例设计人员：熊凯奇 设计日期：2016-06-09

执行测试人员: 熊凯奇 测试日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestPredict-001 | testPredict | 任意股票代码， | 不为null的字符串 | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestPredict-002 | testIncreaseMore | “sha”或“sza” | 长度为50的list | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestPredict-003 | TestIncrease | “sha”或“sza” | 长度为50的list | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |
| TestPredict-004 | TestChangeMax | “sha”或“sza” | 长度为50的list | 基本符合预期 | Pass | 数据会随时间变化，所以只能测试重要的静态数据，下同 |

测试类：predicttest

**3.单元测试报告**

1. 撰写单元测试报告目的
2. 整理和汇总单元测试结果，形成正式的文档
3. 为团队内部对单元的评审提供依据
4. 将该文档纳入软件产品配置管理库
5. 单元测试的范围和工具

单元测试使用工具Junit完成了对“**Stock Market Analysing System**”中主要类的方法的测试

1. 测试用例执行情况记录

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getStockInfo -001 | | |
| 测试单元 | | | getStockInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 测试代码 | | getStockInfoTest | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 市场代码“sha”,并选中其中第6个po的name | | | 显示新疆天业 | | 基本符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01人天 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getStockInfo -002 | | |
| 测试单元 | | | getStockInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 市场代码“shb”，并选中其中第7个po的name | | | 输出“氯碱B股“ | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01人天 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getStockInfo -003 | | |
| 测试单元 | | | getStockInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 测试代码 | | GetStocksListTest | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 输入股票代码“000005”，并获得其中最后一个po的pre\_close | | | 输出浮点数大于0 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01人天 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getStockInfo -004 | | |
| 测试单元 | | | getStockInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 股票代码“000005”，并获得返回的po中的name | | | 输出“世纪星源” | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01人天 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getStockInfo -005 | | |
| 测试单元 | | | getStockInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 股票代码“000009“ | | | 返回字符串“000009” | | 符合预期 | | Pass |  |
| 2 | 股票名称“保千里“ | | | 返回“保千里” | | 符合预期 | | Pass |  |
| 3 | 错误代码“323728” | | | 返回空字符串 | | 符合预期 | |  |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getMarketInfo -001 | | |
| 测试单元 | | | getMarketInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 选择获得的list中第一个的name | | | 输出“上证指数” | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getMarketInfo -002 | | |
| 测试单元 | | | getMarketInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 市场代码“399001“，并选择其中第三个Po的open | | | 输出大于0的浮点数 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getMarketInfo -003 | | |
| 测试单元 | | | getMarketInfo | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 市场代码“399001“，并获得name | | | “深证成指“ | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getPchange -001 | | |
| 测试单元 | | | getPchange | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 输入“600076”，并获得第三个po的market | | | 大于0 | | 符合预期 | | Pass |  |
| 2 | 输入”000009”  并获得第三个po的market | | | 大于0 | | 符合预期 | | Pass |  |
| 3 | “490906” | | | Null | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getIndustry -001 | | |
| 测试单元 | | | getIndustry | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 输入市场代码“sha”取获得List的长度 | | | 10 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getIndex-001 | | |
| 测试单元 | | | getIndex | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | “000005”，取返回List的长度 | | | 30 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getIndex-002 | | |
| 测试单元 | | | getIndex | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 在getVOL“000005”，取返回List的长度 | | | 30 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getIndex-003 | | |
| 测试单元 | | | getIndex | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | getMACD输入“000005”，取返回List的长度 | | | 30 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test getIndex-004 | | |
| 测试单元 | | | getIndex | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | “000005”，取返回List的长度 | | | 30 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-001 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 数据库中不存在输入的用户 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 两次调用signup | | | 第一次返回true，第二次返回false | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-002 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 输入长度大于12的string作为cid和password | | | Boolean值false | | 符合预期 | | Pass |  |
| 2 | 先调用signup注册一个正确格式的账户，而后登录 | | | Login返回true | | 符合预期 | | Pass |  |
| 3 | 在Login中输入不符合格式的账号密码 | | | 返回false | | 基本符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-004 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，并调用raise方法存储 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | getOwnStock输入注册的cid | | | 返回List的长度为1 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-005 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，并调用raise方法存储， | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | getIndustry输入注册的cid | | | 返回浮点数数组的长度为10 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-006 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，并调用raise方法存储至少三只自选股 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | getAll输入参数cid | | | 返回的List里第一个PO的date为当天日期 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-007 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，并调用raise方法存储 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | GetOne输入cid | | | 返回的List里第一个PO的date为当天日期 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-008 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，并调用raise方法存储2个po | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | 调用getOwnStock | | | 返回List长度为1 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-009 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | check之前raise的cid和sid | | | 返回true | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-010 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | check肯定不存在的Cid和sid | | | 返回字符串长度为2 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Login-011 | | |
| 测试单元 | | | login | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 先注册正确用户名和密码，然后raise3个po | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | GetOneStock输入注册的ID | | | 返回字符串长度为2 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Predict-001 | | |
| 测试单元 | | | predict | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | predict输入“000009” | | | 返回不为空字符串 | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Predict-002 | | |
| 测试单元 | | | predict | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | IncreaseMore输入“sha” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 2 | IncreaseMore输入“sza” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Predict-003 | | |
| 测试单元 | | | predict | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | IncreaseMore输入“sha” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 2 | IncreaseMore输入“sza” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例编号 | | | | | | | Test Predict-004 | | |
| 测试单元 | | | predict | | | | | | |
| 是否有测试代码 | | | 是 | | 是否有测试代码 | | 是 | | |
| 前提条件 | | | 无 | | | | | | |
| 序号 | 步骤及操作 | | | 预期结果 | | 实际执行结果 | | Pass/Fail | 备注 |
| 1 | IncreaseMore输入“sha” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 2 | IncreaseMore输入“sza” | | | 长度为50的list | | 符合预期 | | Pass |  |
| 测试人员 | | 熊凯奇 | | | | 测试日期 | | 2016-06-09 | |
| 测试用例实际执行成本 | | | | | | 0.01天人 | | | |

# 4集成测试计划

1. 集成测试计划概述

本文档主要描述**Stock Market Analysing System**的集成测试活动如何进行。包括如何控制集成测试活动，流程以及工作安排等。

本文档只计划针对软件的集成测试，不包括硬件，系统测试和单元测试内容。主要的读者是小组内部成员。集成测试的目标是验证系统和《项目设计文档》的吻合情况。

1. 集成测试的范围和资源
2. 集成测试的主要任务

在连接各个模块的时候，测试穿越模块接口的数据是否会丢失。

在组合各个模块的子功能后，测试是否能达到预期的父功能。

测试一个模块的功能是否会对另一个模块的功能产生不利影响。

测试单个模块的误差积累是否会放大到不可接受的程度。

1. 集成测试的资源

集成测试人员为4人，分工如下。

熊凯奇：软件质量保障员（集成测试负责人）

杨关：技术经理（架构的团队内部负责人）

杨华安，王泽霖：软件工程师

测试执行时间为1个工作日，在完成单元测试的第二天开始（预计为2016.6.9）。

1. 集成测试环境

搭建集成测试环境，1台PC。

操作系统环境：Windows8.1,10

输入习惯：中文。

测试地点在宿舍。

1. 集成测试计划资源分配

熊凯奇全面控制并负责集成测试的组织，书写测试用例，并负责测试环境搭建，同时决定提交上来的Bug是否需要修改。

杨关负责架构一致性和Bug的移除确认。

王泽霖和杨华安负责按照集成测试用例表进行测试活动，记录测试结果。

1. 待测试的清单和顺序
2. 子系统测试顺序

系统分为股票行情，行业分析，股票推荐，用户自选股四个部分。

1. 子系统的测试

集成子系统，测试相关的UI，业务逻辑和数据单元是否达到子系统的功能需求。

数据传递是否准确，对于传入值的控制范围是否一致。

类之间的调用是否正常。

执行集成测试的初步计划如下。

Day1

熊凯奇：构建测试环境，并组织测试的组织。

杨关：debug

杨华安：执行球员统计模块内部的集成和测试，并记录测试结果，反馈Bug。

王泽霖：执行球队统计模块内部的集成与测试，并记录测试结果，反馈Bug。

1. 集成测试结束交付产物
2. 集成测试用例文档的执行情况列表
3. 集成测试报告
4. 执行集成测试的入口和出口条件
5. 入口条件

单元测试必须完成。

开发出完整的系统

搭建了类似于实际应用/配置环境的测试环境

测试人员安排到位。

1. 出口条件（以下条件达到其一就可以进入系统测试）

所有的集成测试用例都被执行，测试结果100%通过

集成测试过程中发现的Bug已经被定位或者关闭。

1. 如何判断集成测试用例通过
2. 接口集成时，接口提供的功能或者数据正确。
3. 功能点集成时，验证与《项目设计文档》中的描述一致。
4. 如何判断集成测试用例失败以及失败后的行为
5. 接口集成时，接口提供的功能或者数据不准确。
6. 功能点集成时，验证与《项目设计文档》中对需求的描述不一致。
7. 集成测试用例未通过，提交测试负责人，由测试负责人熊凯奇判断是否需要转交技术经理杨关，定位和移除Bug。

**5集成测试用例**

**5.1 Integation Test Case L1**

测试用例标题：股票行情的组件集成测试

设计人员：熊凯奇

测试用例表： 设计日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| Version-01 | 查看实时行情 | 点击实时行情 | 显示各市场实时行情，默认进入时为上证A股市场 |  |  |  |
| Version-02 | 切换市场 | 点击其它市场 | 显示所选市场的实时股票列表 |  |  |  |
| Version-03 | 查看大盘 | 点击大盘指数 | 显示6个大盘的实时数据 |  |  |  |
| Version-04 | 查看个股 | 任意选择股票点击 | 显示个股历史信息表格k线图，折线图，散点图表示及相关指数折线表示以及前景预测 |  |  |  |
| Version-05 | 查看指数 | 将相应指数切换为macd | 相关指数中ATR折线图切为MACD折线图 |  |  |  |
| Version-06 | 屏蔽参数 | 点击k线图上方指数MA5 | MA5对应折线消失 |  |  |  |
| Version-07 | 详细显示 | 鼠标移至K线图中某点停留一秒 | 显示对应详细数据 |  |  |  |
| Version-08 | 股票搜索 | 在右上角输入框中输入股票代码或股票名称 | 如不存在则有弹框提示，否则跳转至个股页面 |  |  |  |

**5.2 Integation Test Case L2**

测试用例标题：行业分析的组件集成测试

设计人员：熊凯奇

测试用例表： 设计日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| Version-01 | 查看行业分析 | 点击行业分析或左上角analyse | 显示行业实时分析，涨幅榜，成交量分布扇形图，柱状图 |  |  |  |
| Version-02 | 切换市场 | 涨幅榜中点击深市 | 表格由默认沪市涨幅切换为深市 |  |  |  |
| Version-03 | 扇形图查看成交量详细 | 鼠标移至沪市某行业扇形区 | 显示对应行业详细成交量占比 |  |  |  |
| Version-04 | 查看成交量折线图 | 点击柱状图右方细小折线图表 | 柱状图切换为折线图 |  |  |  |
| Version-05 | 刷新最新情况 | 点击柱状图右方细小刷新图表 | 所有图表重新刷新数据 |  |  |  |

**5.3 Integation Test Case L3**

测试用例标题：股票推荐的组件集成测试

设计人员：熊凯奇

测试用例表： 设计日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| Version-01 | 查看股票推荐 | 点击推荐或右上角recommmend选择市场 | 显示对应市场三个策略的推荐 |  |  |  |
| Version-02 | 详细信息 | 鼠标移至股票指标排名柱状图中 | 显示相应指标详细信息 |  |  |  |
| Version-03 | 数据视图 | 点击柱状图右上表格图标 | 显示数据视图 |  |  |  |
| Version-04 | 折线图 | 点击柱状图右上折线图标 | 柱状图切换为折线图 |  |  |  |
| Version-05 | 刷新最新情况 | 点击柱状图右方细小刷新图表 | 所有图表重新刷新数据 |  |  |  |

**5.4 Integation Test Case L4**

测试用例标题：用户自选股的组件集成测试

设计人员：熊凯奇

测试用例表： 设计日期：2016-06-09

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| Version-01 | 用户注册 | 点击右上角user选择注册，并在注册界面输入用户名，密码，确认密码 | 如果用户名与密码大于4位小于12位且用户名未被占用，则直接进入登陆状态，否则提示注册信息有误 |  |  |  |
| Version-02 | 用户登录 | 输入用户名和密码 | 如登录信息无误则提示登陆成功，否则提示输入信息有误 |  |  |  |
| Version-03 | 选择自选股 | 登陆状态下个股页面点击选择自选股 | 提示添加成功并可以在个人主页查询 |  |  |  |
| Version-04 | 选择个人主页 | 点击user中个人主页 | 如未登录进入快速登陆界面，否则显示个人主页界面，包含虚拟投资，虚拟收益折线图，虚拟行业注资，自选股实时信息表 |  |  |  |
| Version-05 | 删除自选股 | 点击自选股列表或个股页面取消自选股 | 网络正常状态提示取消成功 |  |  |  |
| Version-06 | 修改密码 | 选择修改密码，输入新密码 | 如果密码大于4字节小于12个字节提示修改成功，否则提示信息有误 |  |  |  |
| Version-07 | 修改投资 | 输入股票代码和注资 | 总资产和初始资产发生变化 |  |  |  |
| Version-08 | 收益详情详细信息 | 鼠标移至收益详情 | 显示收益情况对应日期详细数据 |  |  |  |
| Version-09 | 行业注资详细信息 | 鼠标移至行业注资折线图中 | 显示行业注资对应行业详细数据 |  |  |  |

**6集成测试报告**

1. 引言部分
2. 撰写集成测试报告的目的
3. 汇总和统计集成测试的测试结果，并形成最终文档
4. 给出对于测试结果的分析和总结，为团队的开发和测试评审提供依据
5. 术语定义
6. 集成测试：每个模块通过单元测试后，将所有功能模块集成在一块进行测试，以验证各模块的正确性和模块接口的正确性。
7. 相互审查：小组内成员对他人已经测试过的模块进行抽样测试，提高测试效率。
8. 参考资料
9. 《需求分析说明书》
10. 《项目设计文档》
11. 《单元测试计划》
12. 《单元测试报告》
13. 概述
14. 测试对象

本测试主要为NBA\_information\_inquiring\_system的集成测试。主要的功能包括各市场股票实时行情图表，单只股票详细信息图表，预测及各专业指数图表，股票搜索，大盘实时行情图表，大盘历史信息图表，行业分析涨跌幅及成交量排名图表，股票预测多个策略推荐图表，用户注册登录，修改密码，增删自选股，设置虚拟资产及虚拟资产总体与行业与个股变化情况显示。

1. 测试目的

开发人员开发完成及单元测试后，测试所有单元在按照项目设计文档组装成模块，子系统或系统的过程中，各部分是否能正常工作并达到相应要求，确保各单元组合成为模块或系统后能够按照意图运行，同时确保增量的行为正确。

1. 测试环境
2. 操作系统环境：Windows8.1,10
3. 测试地点

宿舍。