# 清结算系统-软件测试文档

目录

[清结算系统-软件测试文档 1](#_Toc516756687)

[**1.     简介** 2](#_Toc516756688)

[**1.1     目的** 2](#_Toc516756689)

[**1.2     背景** 2](#_Toc516756690)

[**1.3     范围** 2](#_Toc516756691)

[**2.     测试需求** 3](#_Toc516756692)

[**3.     测试策略** 3](#_Toc516756693)

[**3.1            测试** 3](#_Toc516756694)

[**3.1.1    功能测试** 3](#_Toc516756695)

[**3.1.2     数据和数据库完整性测试** 4](#_Toc516756696)

[**3.2            测试用例** 4](#_Toc516756697)

**1.     简介**

**1.1     目的**

清结算系统的这一“测试计划”文档有助于实现以下目标：

• 确定现有项目的功能能够正常运行。

• 保证项目对异常状态的处理。

**1.2     背景**

测试对象为清结算系统，主要进行单元测试，对其基本功能进行测试。

测试人员：刘梓雄

测试框架：IDEA Junit4

**1.3     范围**

本测试计划着重于单元测试，主要针对功能测试。将对充值（Recharge）、提现（Withdraw）、交易（Trade）、数据库记录（QueryRecord）、下载文件（DownloadFile）几个功能进行测试。

测试方法：黑盒测试。

**2.     测试需求**

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

|  |  |
| --- | --- |
| 待测用例（方法） | 名称 |
| 1 | 充值 |
| 2 | 提现 |
| 3 | 交易 |
| 4 | 数据库记录 |
| 5 | 下载文件 |

**3.     测试策略**

**3.1            测试**

**3.1.1    功能测试**

测试基于黑盒测试，即通过应用程序交互分析输出结果来验证应用程序及其内部进程。

|  |  |
| --- | --- |
| 测试目标： | 确保测试对象的功能正常，主要包括数据输入及处理。 |
| 方法： | 利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或异常警告消息。 |
| 完成标准： | • 所计划的测试已全部执行。  • 所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | 无 |

**3.1.2     数据和数据库完整性测试**

数据库和数据库进程应作为清结算系统中的子系统来进行测试。

|  |  |
| --- | --- |
| 测试目标： | 确保数据库访问方法和进程正常运行，数据不会遭到损坏。 |
| 方法： | • 调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据或对数据的请求。  • 检查数据库，确保数据已按预期的方式填充，并且所有数据库事件都按正常方式出现；或者检查所返回的数据，确保为正当的检索到了正确的数据 |
| 完成标准： | 所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。 |
| 需考虑的特殊事项： | 进程应该以手工方式调用。 |

**3.2            测试用例**

测试基于黑盒测试。

1. 充值、提现

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入及外部条件 | 有效等价类 | 等价类编号 | 无效等价类 | 等价类编号 |
| requestID | 不为空且为数字 | 1 | null | 13 |
|  |  |  | 不为空但不为数字 | 14 |
| userID | 不为空且为数字 | 2 | null | 15 |
|  |  |  | 不为空但不为数字 | 16 |
| amount | amount>=0 | 3 | <0 | 17 |
| 请求时间requestTime | 不为空且格式正确 | 4 | null | 18 |
|  |  |  | 格式错误 | 35 |
| 月MM | 1<=MM<=12 | 5 | MM<1 | 19 |
|  |  |  | MM>12 | 20 |
| 日DD |  |  |  |  |
| MM{1,3,5,7,8,10,12} | 1<=DD<=31 | 6 | DD<1 | 21 |
|  |  |  | DD>31 | 22 |
| MM{4,6,9,11} | 1<=DD<=30 | 7 | DD<1 | 23 |
|  |  |  | DD>30 | 24 |
| 非闰年2月 | 1<=DD<=28 | 8 | DD<1 | 25 |
|  |  |  | DD>28 | 26 |
| 闰年2月 | 1<=DD<=29 | 9 | DD<1 | 27 |
|  |  |  | DD>29 | 28 |
| 时h | 0<=h<24 | 10 | h<0 | 29 |
|  |  |  | h>=24 | 30 |
| 分m | 0<=m<60 | 11 | m<0 | 31 |
|  |  |  | m>=60 | 32 |
| 秒s | 0<=s<60 | 12 | s<0 | 33 |
|  |  |  | s>=60 | 34 |

测试用例：

|  |  |  |
| --- | --- | --- |
| 序号 | 输入 | 覆盖等价类 |
| Return “-1” | | |
| 1 | requestID=”” | 13 |
| 2 | requestID=”abc” | 14 |
| 3 | userID=”” | 15 |
| 4 | userID=”abc” | 16 |
| 5 | account = -100 | 17 |
| 6 | requestTime = “” | 18 |
| 7 | requestTime = “2018-00-01 03:03:03” | 4,10,12,19 |
| 8 | requestTime = “2018-13-01 03:03:03” | 4,10,12,20 |
| 9 | requestTime = “2018-03-00 03:03:03” | 4,5,10,12,21 |
| 10 | requestTime = “2018-03-33 03:03:03” | 4,5,10,12,22 |
| 11 | requestTime = “2018-04-00 03:03:03” | 4,5,10,12,23 |
| 12 | requestTime = “2018-04-31 03:03:03” | 4,5,10,12,24 |
| 13 | requestTime = “2018-02-00 03:03:03” | 4,5,10,12,25 |
| 14 | requestTime = “2018-02-29 03:03:03” | 4,5,10,12,26 |
| 15 | requestTime = “2008-02-00 03:03:03” | 4,5,10,12,27 |
| 16 | requestTime = “2008-02-30 03:03:03” | 4,5,10,12,28 |
| 17 | requestTime = “2018-06-01 -1:03:03” | 4,5,7,11,12,29 |
| 18 | requestTime = “2018-06-01 25:03:03” | 4,5,7,11,12,30 |
| 19 | requestTime = “2018-06-01 12:-3:03” | 4,5,7,10,12,31 |
| 20 | requestTime = “2018-06-01 12:61:03” | 4,5,7,10,12,32 |
| 21 | requestTime = “2018-06-01 12:03:-3” | 4,5,7,10,11,33 |
| 22 | requestTime = “2018-06-01 12:03:63” | 4,5,7,10,11,34 |
| 23 | requestTime = “asdfghjkl” | 35 |
| return requestID | | |
| 24 | requestID = “1001” userID = “1621” amount = 21 method = true requestTime = "2018-06-30 00:00:30" | 1,2,3,5,7,10,11,12 |
| 25 | requestID = “1001” userID = “1621” amount = 21 method = true requestTime = "2018-05-31 15:59:30" | 1,2,3,4,5,6,10,11,12 |
| 26 | requestID = “1001” userID = “1621” amount = 21 method = true requestTime = "2018-02-28 20:00:00" | 1,2,3,4,5,8,10,11,12 |
| 27 | requestID = “1001” userID = “1621” amount = 21 method = true requestTime = "2008-02-29 12:00:00" | 1,2,3,4,5,9,10,11,12 |

注：若数据库有相同requestID的记录，则抛出异常返回”-1”

1. 交易

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入及外部条件 | 有效等价类 | 等价类编号 | 无效等价类 | 等价类编号 |
| requestID | 不为空且为数字 | 1 | null | 13 |
|  |  |  | 不为空但不为数字 | 14 |
| userID | 不为空且为数字 | 2 | null | 15 |
|  |  |  | 不为空但不为数字 | 16 |
| amount | amount>=0 | 3 | <0 | 17 |
| 请求时间requestTime | 不为空且格式正确 | 4 | null | 18 |
|  |  |  | 格式错误 | 38 |
| 月MM | 1<=MM<=12 | 5 | MM<1 | 19 |
|  |  |  | MM>12 | 20 |
| 日DD |  |  |  |  |
| MM{1,3,5,7,8,10,12} | 1<=DD<=31 | 6 | DD<1 | 21 |
|  |  |  | DD>31 | 22 |
| MM{4,6,9,11} | 1<=DD<=30 | 7 | DD<1 | 23 |
|  |  |  | DD>30 | 24 |
| 非闰年2月 | 1<=DD<=28 | 8 | DD<1 | 25 |
|  |  |  | DD>28 | 26 |
| 闰年2月 | 1<=DD<=29 | 9 | h<1 | 27 |
|  |  |  | DD>29 | 28 |
| 时h | 0<=h<24 | 10 | h<0 | 29 |
|  |  |  | h>=24 | 30 |
| 分m | 0<=m<60 | 11 | m<0 | 31 |
|  |  |  | m>=60 | 32 |
| 秒s | 0<=s<60 | 12 | s<0 | 33 |
|  |  |  | s>=60 | 34 |
| merchantID | 不为空且为数字 | 35 | null | 36 |
|  |  |  | 不为空但不为数字 | 37 |

测试用例：

|  |  |  |
| --- | --- | --- |
| 序号 | 输入 | 覆盖等价类 |
| Return “-1” | | |
| 1 | requestID=”” | 13 |
| 2 | requestID=”abc” | 14 |
| 3 | userID=”” | 15 |
| 4 | userID=”abc” | 16 |
| 5 | account = -100 | 17 |
| 6 | requestTime = “” | 18 |
| 7 | requestTime = “2018-00-01 03:03:03” | 4,5,10,12,19 |
| 8 | requestTime = “2018-13-01 03:03:03” | 4,5,10,12,20 |
| 9 | requestTime = “2018-03-00 03:03:03” | 4,5,10,12,21 |
| 10 | requestTime = “2018-03-33 03:03:03” | 4,5,10,12,22 |
| 11 | requestTime = “2018-04-00 03:03:03” | 4,5,10,12,23 |
| 12 | requestTime = “2018-04-31 03:03:03” | 4,5,10,12,24 |
| 13 | requestTime = “2018-02-00 03:03:03” | 4,5,10,12,25 |
| 14 | requestTime = “2018-02-29 03:03:03” | 4,5,10,12,26 |
| 15 | requestTime = “2008-02-00 03:03:03” | 4,5,10,12,27 |
| 16 | requestTime = “2008-02-30 03:03:03” | 4,5,10,12,28 |
| 17 | requestTime = “2018-06-01 -1:03:03” | 4,5,7,11,12,29 |
| 18 | requestTime = “2018-06-01 25:03:03” | 4,5,7,11,12,30 |
| 19 | requestTime = “2018-06-01 12:-3:03” | 4,5,7,10,12,31 |
| 20 | requestTime = “2018-06-01 12:61:03” | 4,5,7,10,12,32 |
| 21 | requestTime = “2018-06-01 12:03:-3” | 4,5,7,10,11,33 |
| 22 | requestTime = “2018-06-01 12:03:63” | 4,5,7,10,11,34 |
| 23 | merchantID = “” | 36 |
| 24 | merchantID = “abc” | 37 |
| Return requestID | | |
| 25 | requestID = “1001” userID = “1621” merchantID = “2333” amount = 21 method = true requestTime = "2018-03-31 12:59:59" | 1,2,3,4,5,6,10,11,12,35 |
| 26 | requestID = “1001” userID = “1621” merchantID = “2333” amount = 99.8 method = true requestTime = "2018-06-30 00:00:00" | 1,2,3,4,5,7,10,11,12,35 |
| 27 | requestID = “1001” userID = “1621” merchantID = “2333” amount = 66.6 method = true requestTime = "2018-02-28 12:03:30" | 1,2,3,4,5,8,10,11,12,35 |
| 28 | requestID = “1001” userID = “1621” merchantID = “2333” amount = 666 method = true requestTime = "2008-02-29 16:00:00" | 1,2,3,4,5,9,10,11,12,35 |

注：若数据库有相同requestID的记录，则抛出异常返回”-1”

1. 记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入及外部条件 | 有效等价类 | 等价类编号 | 无效等价类 | 等价类编号 |
| 开始时间startTime | 不为空 | 1 | null | 11 |
| 结束时间startTime | 不为空 | 2 | null | 12 |
|  |  |  | 早于开始时间 | 29 |
| 时间格式 | | | | |
| 月MM | 1<=MM<=12 | 3 | MM<1 | 13 |
|  |  |  | MM>12 | 14 |
| 日DD |  |  |  |  |
| MM{1,3,5,7,8,10,12} | 1<=DD<=31 | 4 | DD<1 | 15 |
|  |  |  | DD>31 | 16 |
| MM{4,6,9,11} | 1<=DD<=30 | 5 | DD<1 | 17 |
|  |  |  | DD>30 | 18 |
| 非闰年2月 | 1<=DD<=28 | 6 | DD<1 | 19 |
|  |  |  | DD>28 | 20 |
| 闰年2月 | 1<=DD<=29 | 7 | h<1 | 21 |
|  |  |  | DD>29 | 22 |
| 时h | 0<=h<24 | 8 | h<0 | 23 |
|  |  |  | h>=24 | 24 |
| 分m | 0<=m<60 | 9 | m<0 | 25 |
|  |  |  | m>=60 | 26 |
| 秒s | 0<=s<60 | 10 | s<0 | 27 |
|  |  |  | s>=60 | 28 |
| operatorID | {0,1,2} | 30 | 其他 | 31 |

测试用例：(异常数据)

|  |  |  |
| --- | --- | --- |
| 序号 | 输入 | 覆盖等价类 |
| Return “-1” | | |
| 1 | startTime = “” | 11 |
| 2 | startTime = “2018-00-01 03:03:03” | 13 |
| 3 | startTime = “2018-13-01 03:03:03” | 14 |
| 4 | startTime = “2018-03-00 03:03:03” | 15 |
| 5 | startTime = “2018-03-33 03:03:03” | 16 |
| 6 | startTime = “2018-04-00 03:03:03” | 17 |
| 7 | startTime = “2018-04-31 03:03:03” | 18 |
| 8 | startTime = “2018-02-00 03:03:03” | 19 |
| 9 | startTime = “2018-02-29 03:03:03” | 20 |
| 10 | startTime = “2008-02-00 03:03:03” | 21 |
| 11 | startTime = “2008-02-30 03:03:03” | 22 |
| 12 | startTime = “2018-06-01 -1:03:03” | 23 |
| 13 | startTime = “2018-06-01 25:03:03” | 24 |
| 14 | startTime = “2018-06-01 12:-3:03” | 25 |
| 15 | startTime = “2018-06-01 12:61:03” | 26 |
| 16 | startTime = “2018-06-01 12:03:-3” | 27 |
| 17 | startTime = “2018-06-01 12:03:63” | 28 |
| 18 | startTime = “2018-06-01 12:00:00”  endTime = “” | 12 |
| 19 | startTime = “2018-06-01 12:00:00”  endTime = “2018-05-01 12:00:00” | 29 |
| 20 | operateID = 123 | 31 |
| return | | |
| 21 | startTime = “2008-01-01 00:00:00”  endTime = “2020-12-01 00:00:00”  operateID = 0 | 1,2,30 |

1. 下载

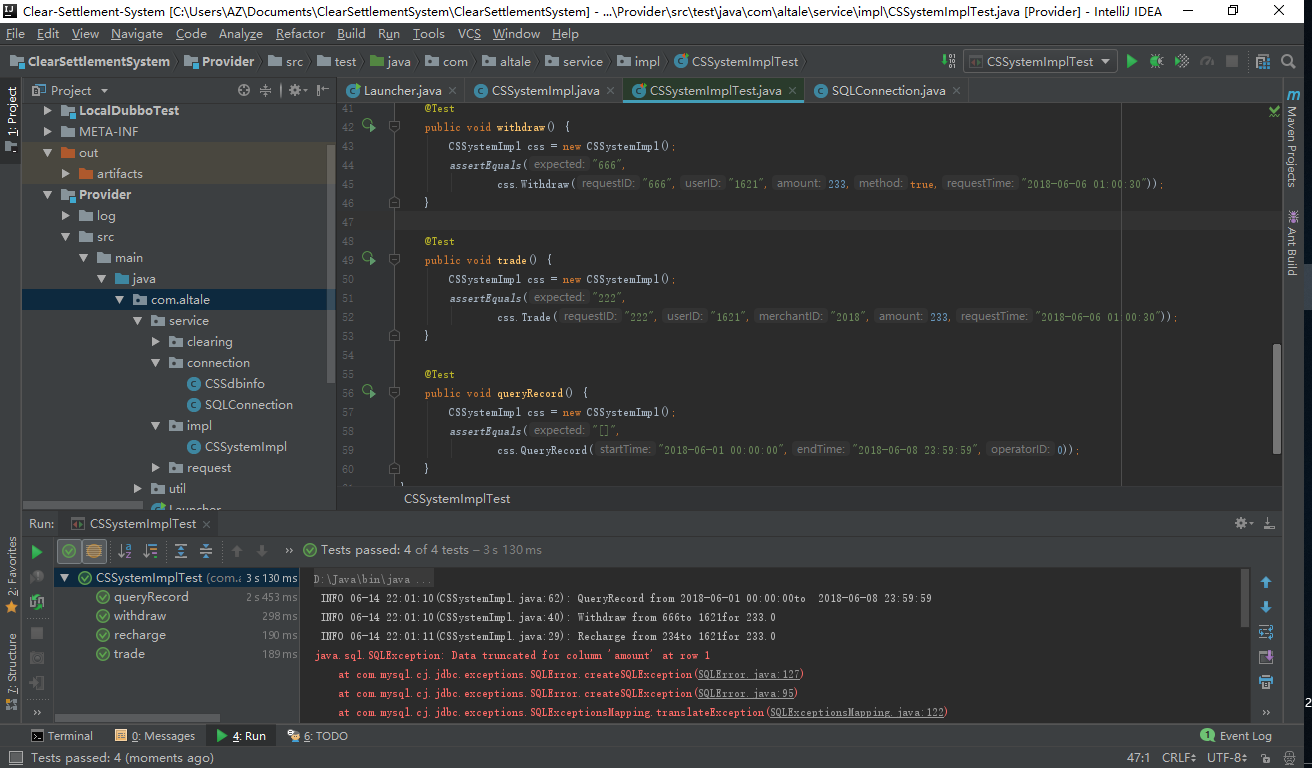
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. 输入及外部条件 | 有效等价类 | 等价类编号 | 无效等价类 | 等价类编号 |
| requestTime | 不为空 | 1 | null | 10 |
| 时间格式 | | | | |
| 月MM | 1<=MM<=12 | 2 | MM<1 | 11 |
|  |  |  | MM>12 | 12 |
| 日DD |  |  |  |  |
| MM{1,3,5,7,8,10,12} | 1<=DD<=31 | 3 | DD<1 | 13 |
|  |  |  | DD>31 | 14 |
| MM{4,6,9,11} | 1<=DD<=30 | 4 | DD<1 | 15 |
|  |  |  | DD>30 | 16 |
| 非闰年2月 | 1<=DD<=28 | 5 | DD<1 | 17 |
|  |  |  | DD>28 | 18 |
| 闰年2月 | 1<=DD<=29 | 6 | h<1 | 19 |
|  |  |  | DD>29 | 20 |
| 时h | 0<=h<24 | 7 | h<0 | 21 |
|  |  |  | h>=24 | 22 |
| 分m | 0<=m<60 | 8 | m<0 | 23 |
|  |  |  | m>=60 | 24 |
| 秒s | 0<=s<60 | 9 | s<0 | 25 |
|  |  |  | s>=60 | 26 |

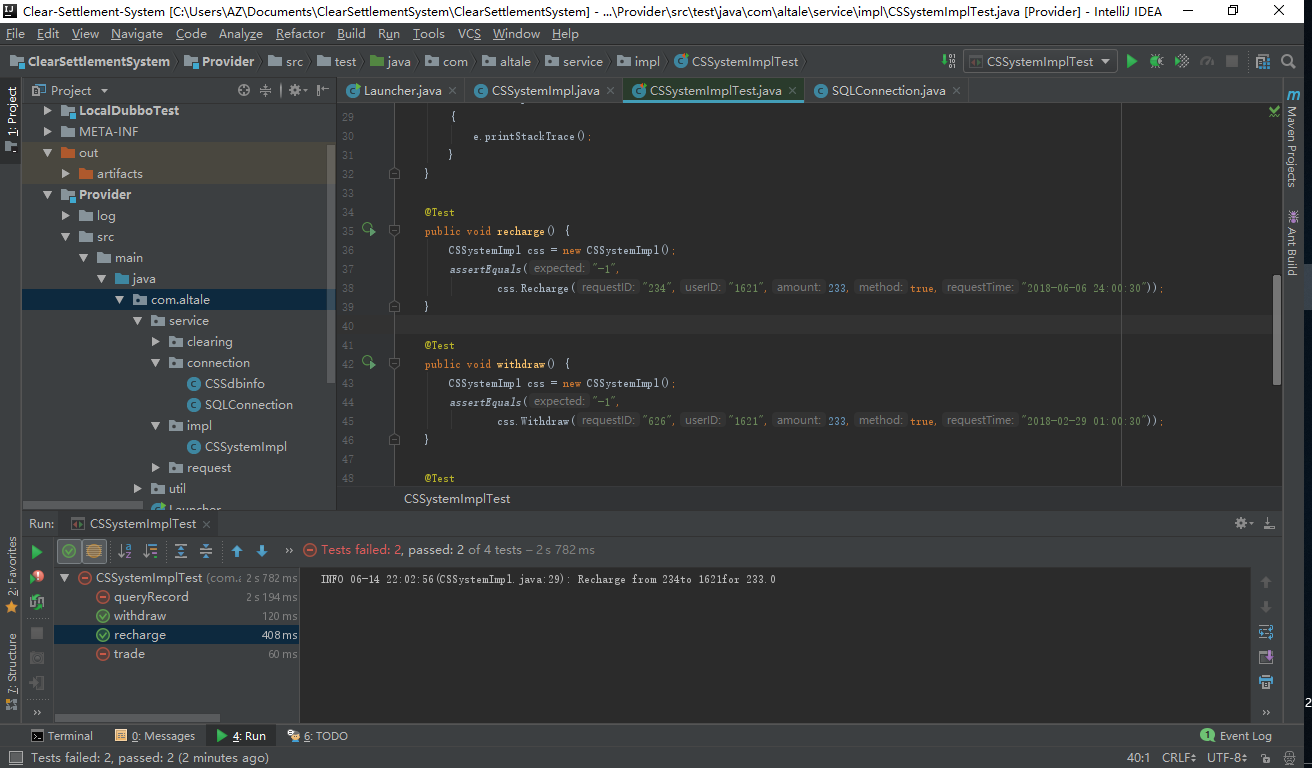
测试用例：

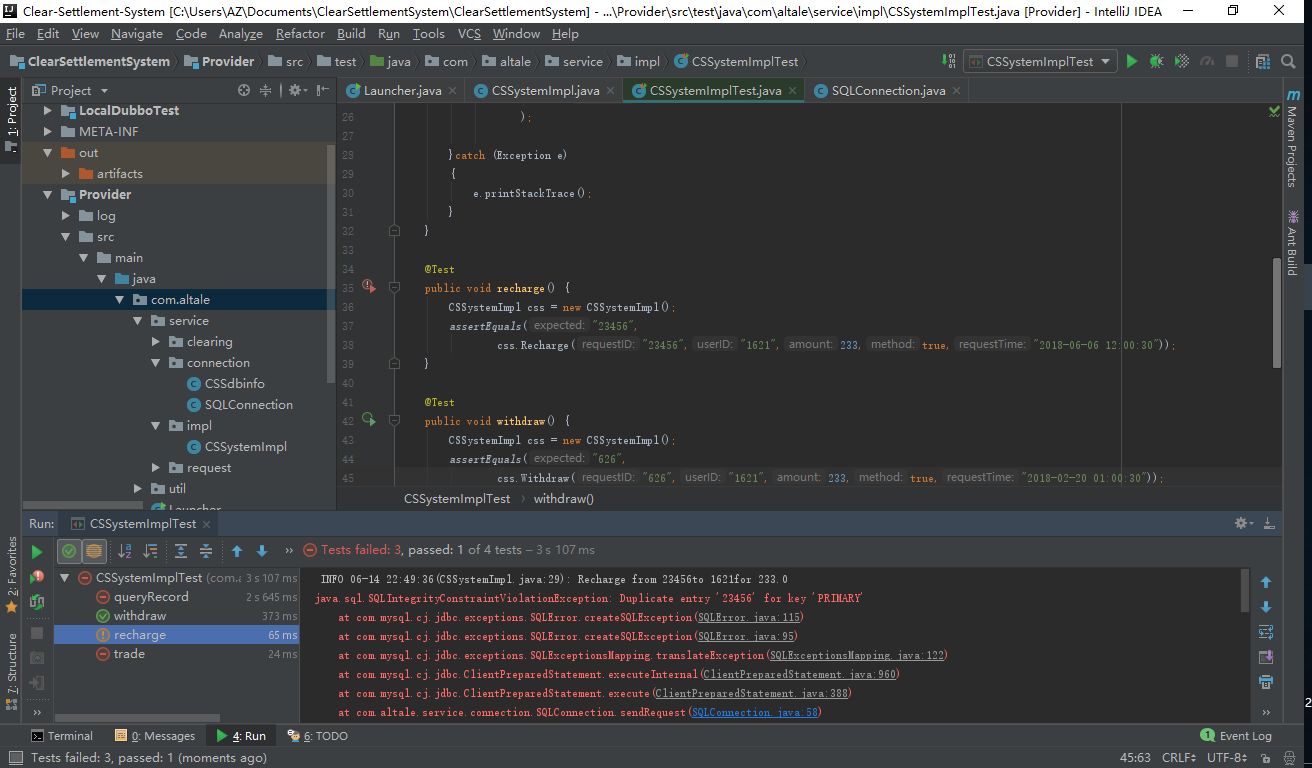
|  |  |  |
| --- | --- | --- |
| 序号 | 输入 | 覆盖等价类 |
| Return “-1” | | |
| 1 | requestTime = “” | 10 |
| 2 | requestTime = “2018-00-01 03:03:03” | 11 |
| 3 | requestTime = “2018-13-01 03:03:03” | 12 |
| 4 | requestTime = “2018-03-00 03:03:03” | 13 |
| 5 | requestTime = “2018-03-33 03:03:03” | 14 |
| 6 | requestTime = “2018-04-00 03:03:03” | 15 |
| 7 | requestTime = “2018-04-31 03:03:03” | 16 |
| 8 | requestTime = “2018-02-00 03:03:03” | 17 |
| 9 | requestTime = “2018-02-29 03:03:03” | 18 |
| 10 | requestTime = “2008-02-00 03:03:03” | 19 |
| 11 | requestTime = “2008-02-30 03:03:03” | 20 |
| 12 | requestTime = “2018-06-01 -1:03:03” | 21 |
| 13 | requestTime = “2018-06-01 25:03:03” | 22 |
| 14 | requestTime = “2018-06-01 12:-3:03” | 23 |
| 15 | requestTime = “2018-06-01 12:61:03” | 24 |
| 16 | requestTime = “2018-06-01 12:03:-3” | 25 |
| 17 | requestTime = “2018-06-01 12:03:63” | 26 |
| return | | |
| 18 | requestTime = “2018-05-30 20:00:30” | 1,2,3,7,8,9 |
| 19 | requestTime = “2018-06-06 20:00:30” | 1,2,4,7,8,9 |

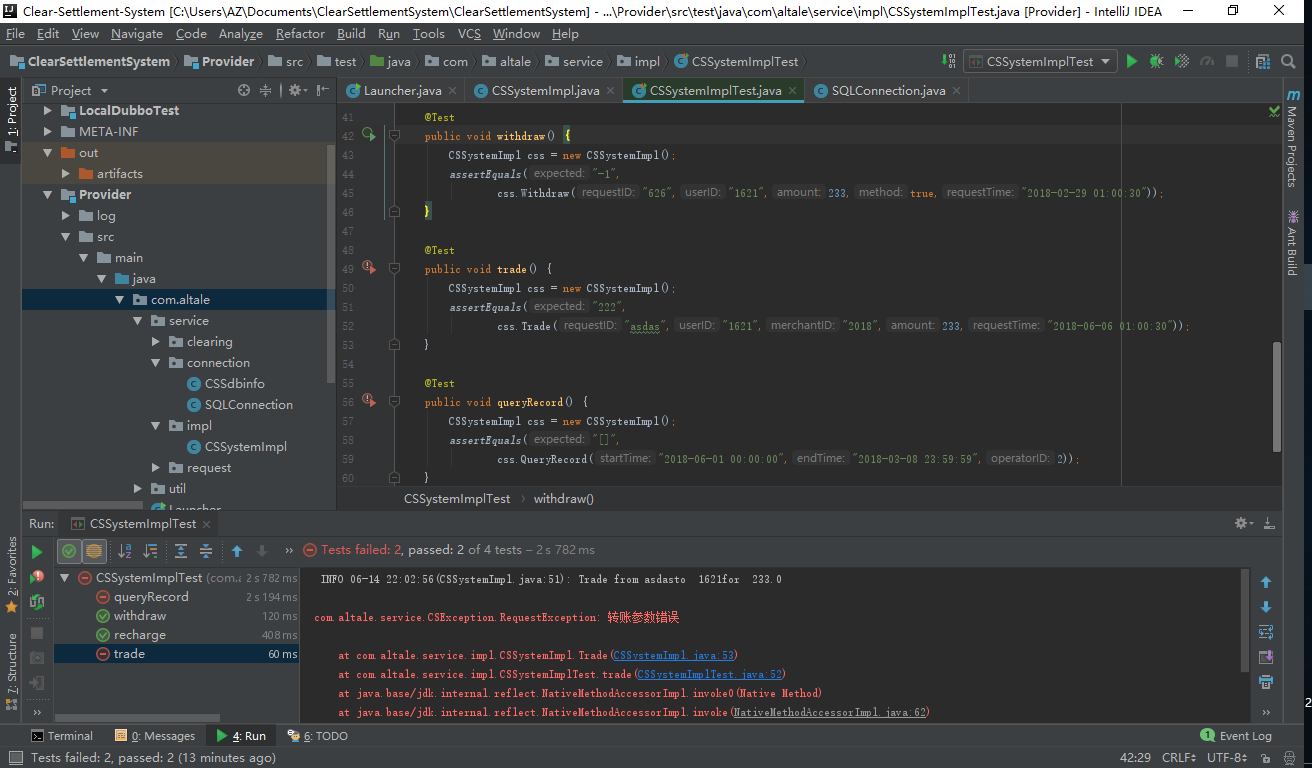
注：若数据库内这段时间没有记录，或者查询时间超出15天内，则抛出异常

**3.     测试结果举例**

Recharge,Withdraw,Trade,queryRecord模块测试，能够正确运行的用例（Recharge和Withdraw调用方法相同），测试通过。

Recharge,Withdraw，不能正常运行的无效用例，测试通过。

Recharge主键冲突，抛出异常，测试通过。

 Trade,queryRecord，不能正常运行的用例，测试通过。（抛出异常无返回值）