Daily Trade Reporting Engine

Release Notes

Version: **1.0**

**Date:** 03rd November 2017

**Author:** Adhilakshmi Gopalakrishnan

**Purpose**

The purpose of this document is to provide the user an understanding on the below information.

1. Problem Description
2. Expectation
3. Assumptions
4. Design and Implementation
5. Error Handling
6. JUnit Testing and Test Scenarios
7. Known Issues
8. Deliverables

**Problem Description**

Sample data represents the instructions sent by various clients to JP Morgan to execute in the international market.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Buy/Sell** | **AgreedFx** | **Currency** | **Instruction Date** | **Settlement Date** | **Units** | **Price Per Unit** |
| Foo | B | 0.50 | SGP | 01 Jan 2016 | 02 Jan 2016 | 200 | 100.25 |
| Bar | S | 0.22 | AED | 05 Jan 2016 | 07 Jan 2017 | 450 | 150 |

* A work week starts Monday and ends Friday, unless the currency of the trade is AED or SAR, where the work week starts Sunday and ends Thursday. No other holidays to be taken into account.
* A trade can only be settled on a working day.
* If an instructed settlement date falls on a weekend, then the settlement date should be changed to the next working day.
* USD amount of a trade = Price per unit \* Units \* Agreed Fx

**Expectation**

Create a report that shows

* Amount in USD settled incoming everyday
* Amount in USD settled outgoing everyday
* Ranking of entities based on incoming and outgoing amount. E.g.: If entity foo instructs the highest amount for a buy instruction, then foo is rank 1 for outgoing

**Assumptions**

The below are the assumption taken to proceed the design and implementation.

1. All the instruction sent by the various clients is available in the CSV file named “**TradingDetails.csv**” with the delimiter **“;”** and all the values should contain values but not null values.

**Example:**

foo;B;0.5;SGD;01-Jan-2016;02-Jan-2016;200;100.25

bar;S;0.22;AED;05-Jan-2016;07-Jan-2016;450;150.5

1. Date format in the CSV files taken as **dd-MMM-yyyy**.
2. If the Settlement date is in the weekend or holiday based on the currency then the validation of settlement date will be done and to be stored in the memory which is used for generating report. The file will not get updated with the next work day if the settlement date is in week end.
3. In the report, all the trading entities are to be displayed with rank based on the Entity type USD amount.
4. In the report ,ranked by descending order of USDAmt i.e. **Highest amount** in the first followed by **lowest amount** with the details of **Buying** trading entities and followed by **Selling trading entities** **Highest USD amount** to **Lowest USD amount.**
5. Since it is the report, so all the settlement dates are displayed in the report.
6. If there is trading details and trying to call the generate report method will generate the empty header and footer of the report and followed by the below message “**No Trading Entity Details Found**”.

**Design and Implementation**

The below JAVA programs are involved in the design of the solution to the problem specification.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Program Name** | **Description** |
| 1 | TradingSystemEntity.java | This java class is used to hold all the entity details are as member variables i.e.) entity, entity type, agreedfx, currency, instruction date, settlement date, no of units, price per unit and also having member functions to validate settlement date , calculate USD amount and generate the report. |
| 2 | TradingSystemException.java | This class handles the user defined exception of Trading system. |
| 3 | RetrieveTradingDetails.java | This java class is used to test the generation of report by creating the object of Trading SystemEntity class. |

The javadoc documentation for the above mentioned java files are done and available in the workspace “**DailyTradingSystem**” created path in the folder “**doc”.** Please refer for more details.

**Error Handling**

This design of the solution throws the below error code and error message if any error occurred during generation of the report.

|  |  |  |
| --- | --- | --- |
| **Error Code** | Error Message | **When** |
| 100 | Settlement Date cannot be null | Exception thrown when Settlement Date is null |
| 101 | Currency cannot be null | Exception thrown when Currency is null |
| 102 | Price per Unit cannot be null | Exception thrown when Price per unit is null |
| 103 | Price per Unit cannot Zero | Exception thrown when Price per unit is zero |
| 104 | NoOfUnit cannot be Zero | Exception thrown when no of unit is zero |
| 105 | agreedFx cannot be null | Exception thrown when agreedfx is null |
| 106 | agreedFx cannot Zero | Exception thrown when agreedfx is zero |
| 107 | No trading entities are found to be ranked | Exception thrown when there is no values in the list of TradingSystemEntity object. |

**JUnit Testing and Test Scenarios**

The below JUnit java classes created for testing the different test scenarios with the different test data.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Program Name** | **Description** |
| 1 | GenerateTradeEntityReportMethodTest.java | This JUnit java class is used to do unit test and check the function generateTradingEntityReport method is working fine with the valid values passed and null value passed as parameter. |
| 2 | CalculateUSDAmtTest.java | This JUnit Java class is used to do unit test the method calculateUSDAmtTest method in TradingEntitySystem class |
| 3 | RankEntityMethodTest.java | This JUnit Java class is used to do unit test the method rankEntity method in TradingEntitySystem class. |
| 4 | SettelmentDtTestWithParameter.java | This JUnit java class is used to do unit test the validateSettlementDt method in TradingSystemEntity class with different parameter passed by using the parameterized JUnit Test |
| 5 | AllTests | This JUnit java class is used as a Test Suite which calls the entire test in one run. |

**Test Scenarios**

1. Testing with the below Settlement dates and Currency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Currency** | **Settlement Date** | **Expected Output** | **Actual Output** | **Test Result** |
| AED | Saturday, 27 February 2016 | Sunday, 28 February 2016 | Sunday, 28 February 2016 | PASS |
| AED | Sunday, 28 February 2016 | Sunday, 28 February 2016 | Sunday, 28 February 2016 | PASS |
| AED | Monday, 29 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| AED | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | PASS |
| AED | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | PASS |
| AED | Thursday, 3 March 2016 | Thursday, 3 March 2016 | Thursday, 3 March 2016 | PASS |
| AED | Friday, 4 March 2016 | Sunday, 6 March 2016 | Sunday, 6 March 2016 | PASS |
| AED | Saturday, 5 March 2016 | Sunday, 6 March 2016 | Sunday, 6 March 2016 | PASS |
| SAR | Saturday, 27 February 2016 | Sunday, 28 February 2016 | Sunday, 28 February 2016 | PASS |
| SAR | Sunday, 28 February 2016 | Sunday, 28 February 2016 | Sunday, 28 February 2016 | PASS |
| SAR | Monday, 29 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| SAR | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | PASS |
| SAR | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | PASS |
| SAR | Thursday, 3 March 2016 | Thursday, 3 March 2016 | Thursday, 3 March 2016 | PASS |
| SAR | Friday, 4 March 2016 | Sunday, 6 March 2016 | Sunday, 6 March 2016 | PASS |
| SAR | Saturday, 5 March 2016 | Sunday, 6 March 2016 | Sunday, 6 March 2016 | PASS |
| INR | Saturday, 27 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| INR | Sunday, 28 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| INR | Monday, 29 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| INR | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | PASS |
| INR | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | PASS |
| INR | Thursday, 3 March 2016 | Thursday, 3 March 2016 | Thursday, 3 March 2016 | PASS |
| INR | Friday, 4 March 2016 | Friday, 4 March 2016 | Friday, 4 March 2016 | PASS |
| INR | Saturday, 5 March 2016 | Monday, 7 March 2016 | Monday, 7 March 2016 | PASS |
| INR | Sunday, 6 March 2016 | Monday, 7 March 2016 | Monday, 7 March 2016 | PASS |
| USD | Saturday, 27 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| USD | Sunday, 28 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| USD | Monday, 29 February 2016 | Monday, 29 February 2016 | Monday, 29 February 2016 | PASS |
| USD | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | Tuesday, 1 March 2016 | PASS |
| USD | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | Wednesday, 2 March 2016 | PASS |
| USD | Thursday, 3 March 2016 | Thursday, 3 March 2016 | Thursday, 3 March 2016 | PASS |
| USD | Friday, 4 March 2016 | Friday, 4 March 2016 | Friday, 4 March 2016 | PASS |
| USD | Saturday, 5 March 2016 | Monday, 7 March 2016 | Monday, 7 March 2016 | PASS |
| USD | Sunday, 6 March 2016 | Monday, 7 March 2016 | Monday, 7 March 2016 | PASS |

1. Testing with null values passed to the validateSettlementDt method and expect that TradeEntityException thrown with the error code and error message.
2. Testing with null values passed to the calculateUSDAmt method and expect that TradeEntityException thrown with the error code and error message.
3. Testing with the null values passed to the rankEntity method and expect that TradeEntityException thrown with the error code and error message.
4. Testing with the null values passed to the generateTradingEntityReport method and expects that no exception thrown if there is no values in the list object. Instead display the header and footer of the report are displayed with the message “**No Trading Entity Details Found**”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Test Case** | **Test Case Executed** | **Total Pass Test Case** | **Total Fail Test Case** |
| **47** | **47** | **47** | **0** |

**Known Issues**

There is no known issue.

**Deliverables**

The release of this project contains the below deliverables.

**DailyTradingSystem** as **Workspace name**

|  |  |  |
| --- | --- | --- |
| **No** | **File Name** | **File Path** |
| 1 | TradingSystemEntity.java | DailyTradingSystem\src\TradingSystemPack |
| 2 | RetrieveTradingDetails.java | DailyTradingSystem\src\TradingSystemPack |
| 3 | TradingSystemException.java | DailyTradingSystem\src\TradingSystemPack |
| 4 | AllTests.java | DailyTradingSystem\TradingSystemEntityTest\TradingSystemPack |
| 5 | CalculateUSDAmtTest.java | DailyTradingSystem\TradingSystemEntityTest\TradingSystemPack |
| 6 | GenerateTradeEntityReportMethodTest.java | DailyTradingSystem\TradingSystemEntityTest\TradingSystemPack |
| 7 | RankEntityMethodTest.java | DailyTradingSystem\TradingSystemEntityTest\TradingSystemPack |
| 8 | SettelmentDtTestWithParameter.java | DailyTradingSystem\TradingSystemEntityTest\TradingSystemPack |
| 9 | Javadoc | DailyTradingSystem\doc\ |
| 10 | TradingDetails.csv | DailyTradingSystem |