Software Design Document

Project Name : Quantitative Trading Platform

Developer : 鄒咏霖、汪宸宇、陳熙、李辰暘、侯思岑、李紹銘

# 目錄

[**目錄**](#_cjoj7jblafvs) **1**

[**版本變更紀錄**](#_hrluv6mjnov) **2**

[**1.系統模型與架構(System Model/System Architecture)**](#_76zleklfwn7c) **3**

[1.1 設計動機](#_hm041yis36ye) 3

[1.2 架構圖](#_op2hdjty75qh) 4

[**2.流程設計(Process Design)**](#_e0qr35kq5mzq) **5**

[**3.使用者介面設計(User Interface Design)**](#_bq3jgmubqag2) **6**

[3.1 Home Page (File Manager)](#_bpyh2q9saye5) 6

[3.2 Report Reader](#_dfmzq4px9k9t) 7

[3.3 Evaluation and Parameter Setting for Single Test](#_u41r7wl3uzx0) 8

[3.4 Evaluation and Parameter Setting for Batch Test](#_w74e3125psog) 9

[**4.類別圖設計(Class Diagram)**](#_nnzsyvi4hgox) **10**

[4.1 系統分類](#_2j5ljlpp4l5x) 10

[4.2 類別圖](#_40xkvh6urlhz) 11

[**5.循序圖設計(Sequence Diagram)**](#_kv7vmvfm3jjp) **12**

[**6.資料細部設計(Data Design)**](#_f1zbzce0vqov) **13**

# 版本變更紀錄

|  |  |  |
| --- | --- | --- |
| 版次 | 變更項目 | 變更日期 |
| 0.1 | Class Diagram, System Architecture | 2021/5/10 |
| 0.2 | UI Design | 2021/5/15 |
| 0.3 | Process Design, Sequence Diagram, Data Design | 2021/5/17 |

# 1.系統模型與架構(System Model/System Architecture)

## 1.1 Motivation

When it comes to investing, quantitative trading is perhaps the most reliable method widely used in major institutions on Wall Streets; hence, this project aims at building a quantitative platform to help users testing their trading ideas and accumulating their precious findings. Many quantitative research platforms have built up and became popular on the Internet, like Quantopian, TradingView, RiceQuant and so on. However, some common drawbacks of these platforms are:

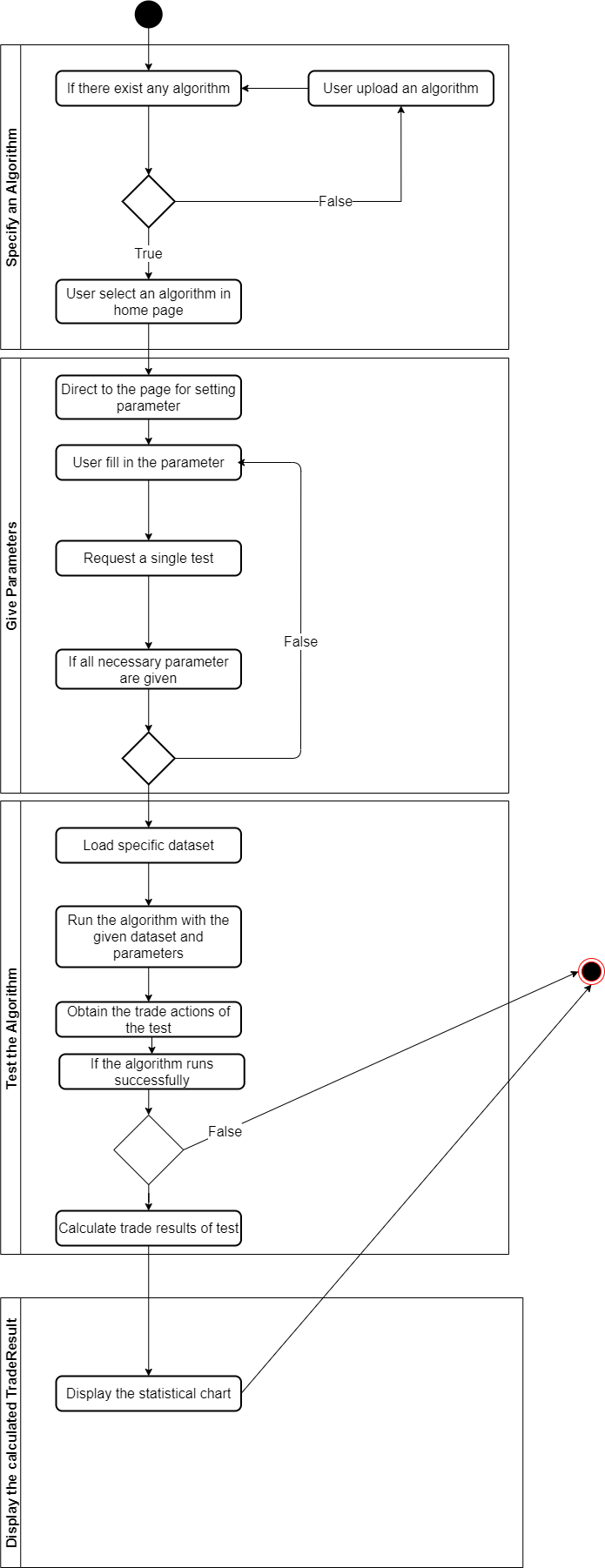
* many functionalities are costly
* visualization tools for performance analysis are inflexible or even lacking
* lack of a report-file management system; users have to organize their findings themselves, store them to their local hard drives and link them manually

We hope to build a system that can evaluate different trading methodologies and manage the outcomes gained from those testing results.

## 1.2 System Architecture

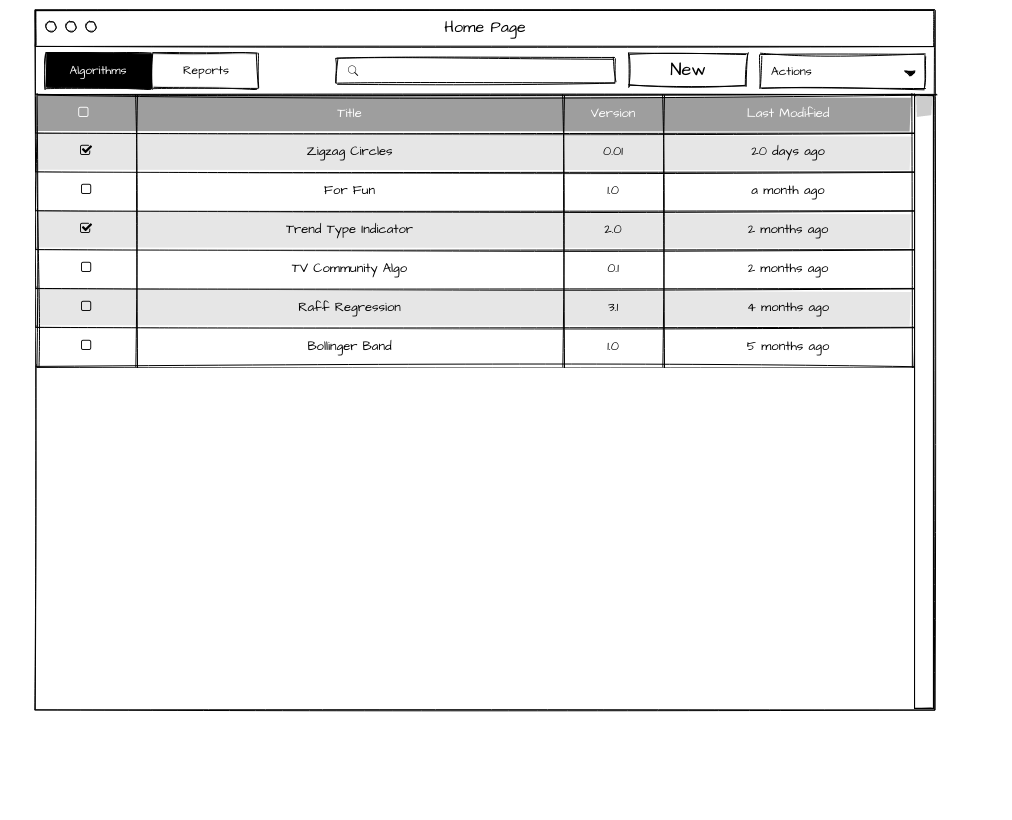
## 

# 2.流程設計(Process Design)



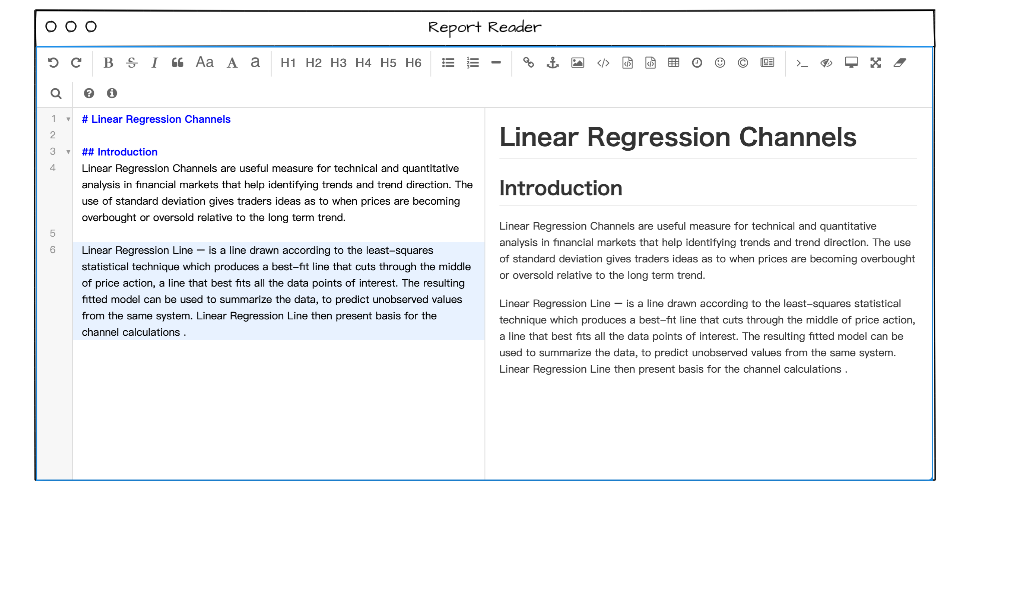
# 3.使用者介面設計(User Interface Design)

## 3.1 Home Page (File Manager)



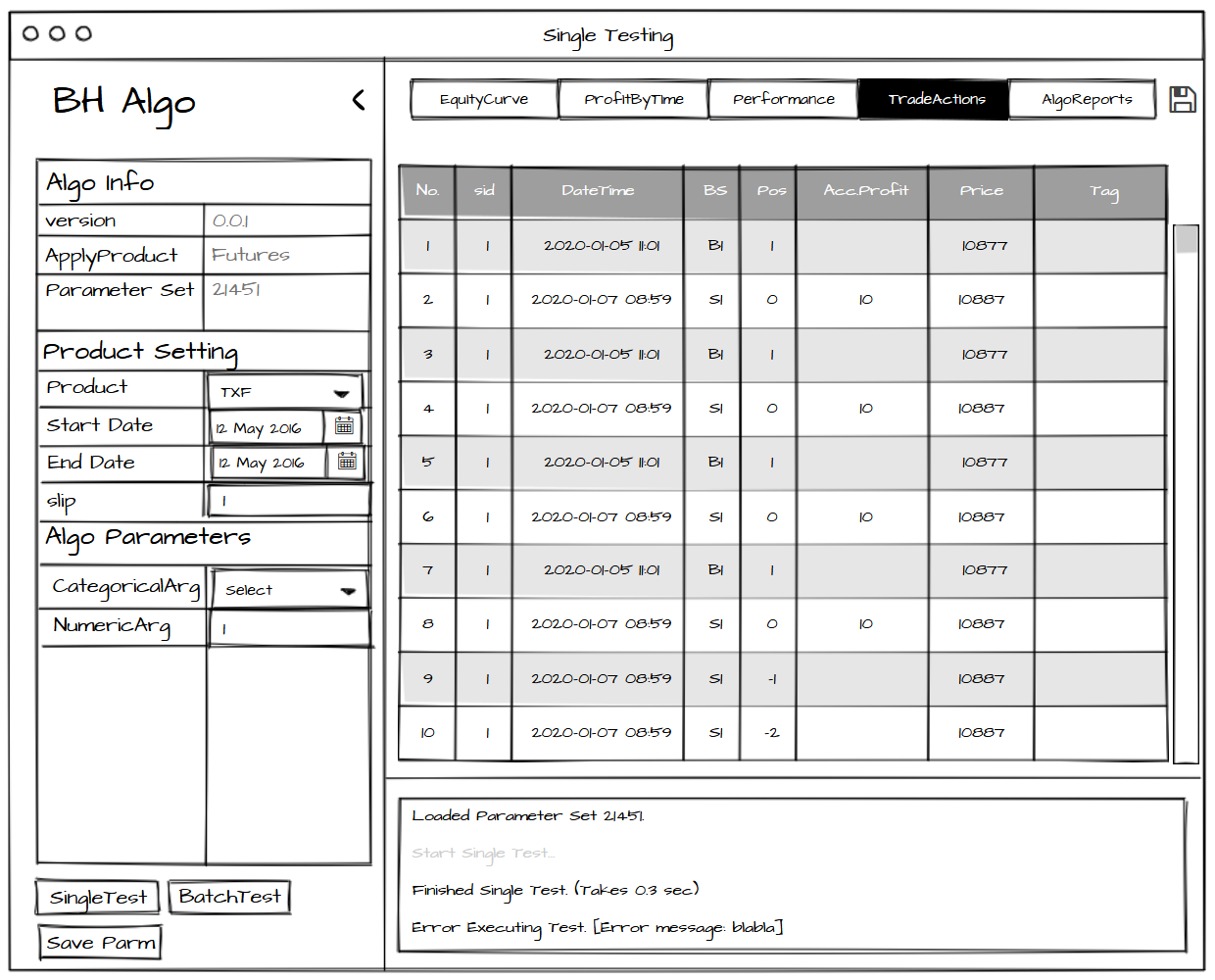
## 

## 3.2 Report Reader

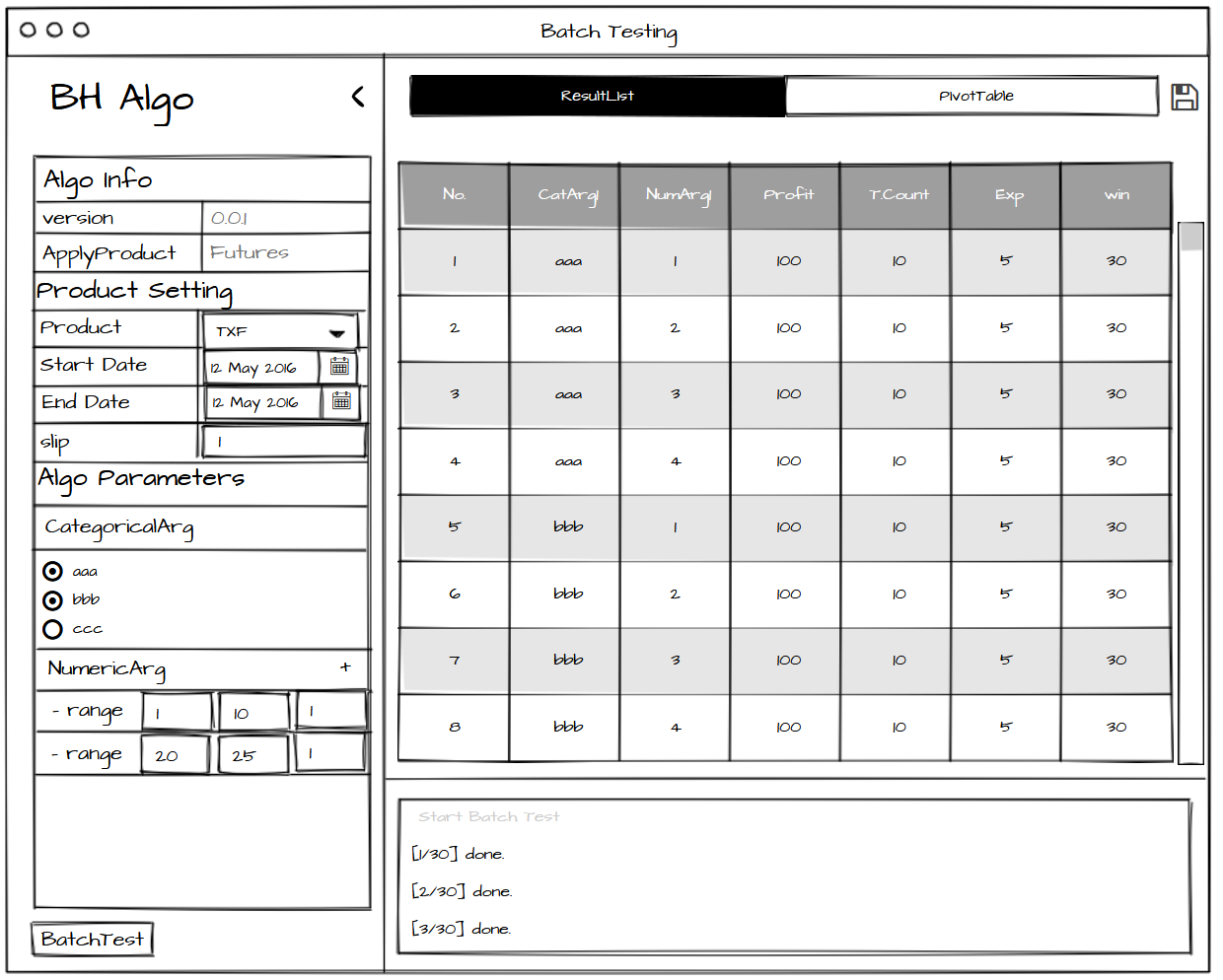


## 

## 3.3 Evaluation and Parameter Setting for Single Test



## 3.4 Evaluation and Parameter Setting for Batch Test



# 

# 

# 4.類別圖設計(Class Diagram)

## 4.1 系統分類

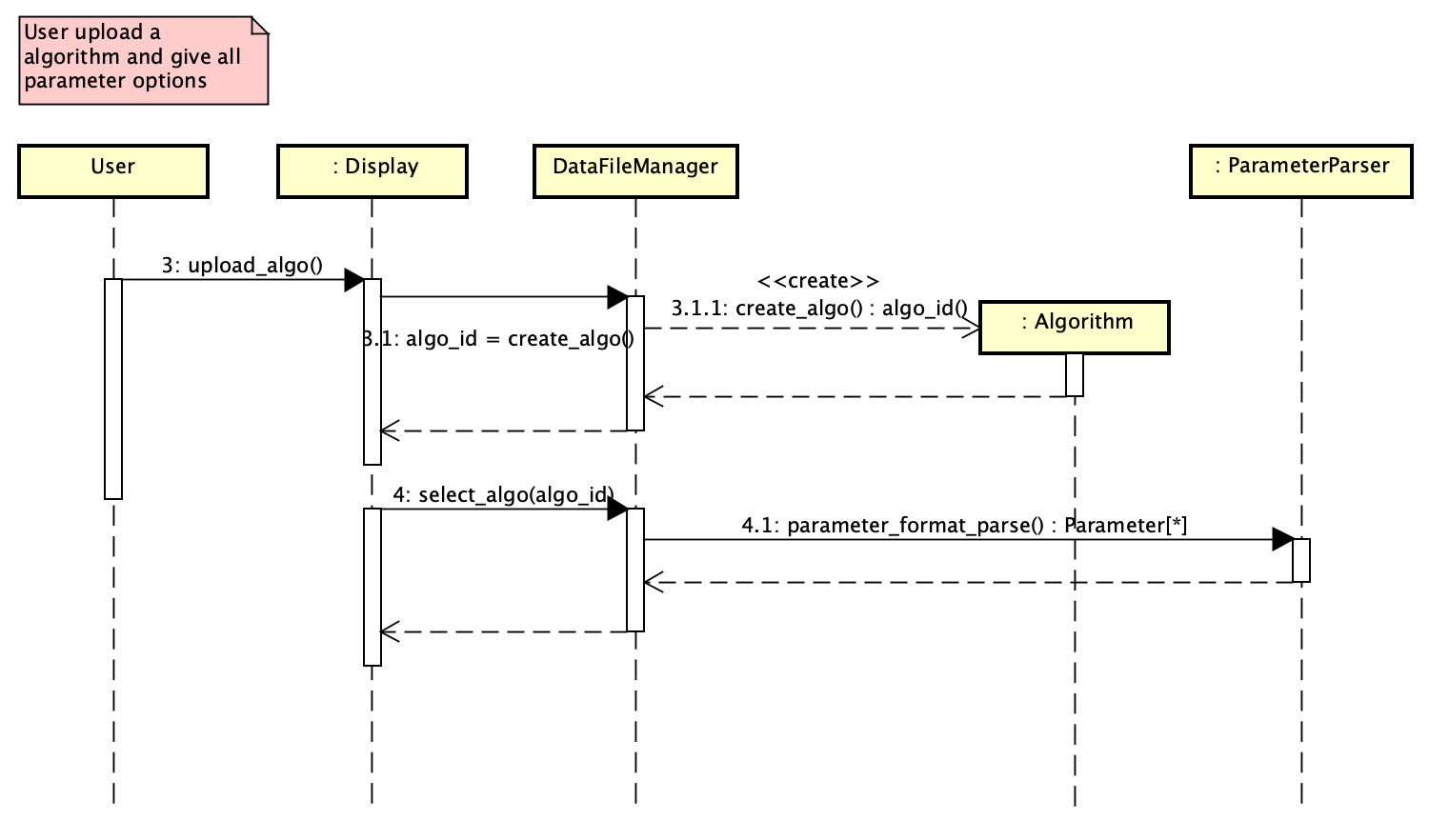
|  |  |  |
| --- | --- | --- |
| **Module** | **Description** | **顏色** |
| **算法測試系統 Algorithm Testing Module** | Building and testing trading algorithms |  |
| **評估系統 Evaluation Module** | Evaluating performance of an algorithm and display results on the screen |  |
| **文檔管理系統 File Management Module** | Providing the ability to manage algorithms and reports created by users and store them |  |
| **標的資料處理與供給系統 Underlying Data Management Module** | Forward the underlying assets’ data to other modules |  |
| **參數設定系統 Parameter Setting Module** | Rendering and setting parameters of an algorithm |  |
| **績效計算系統 Performance Calculator Module** | Calculating performance of an Algorithm |  |
| **資料類別Data Classes** | Representing the data classes we use in the system |  |

## 

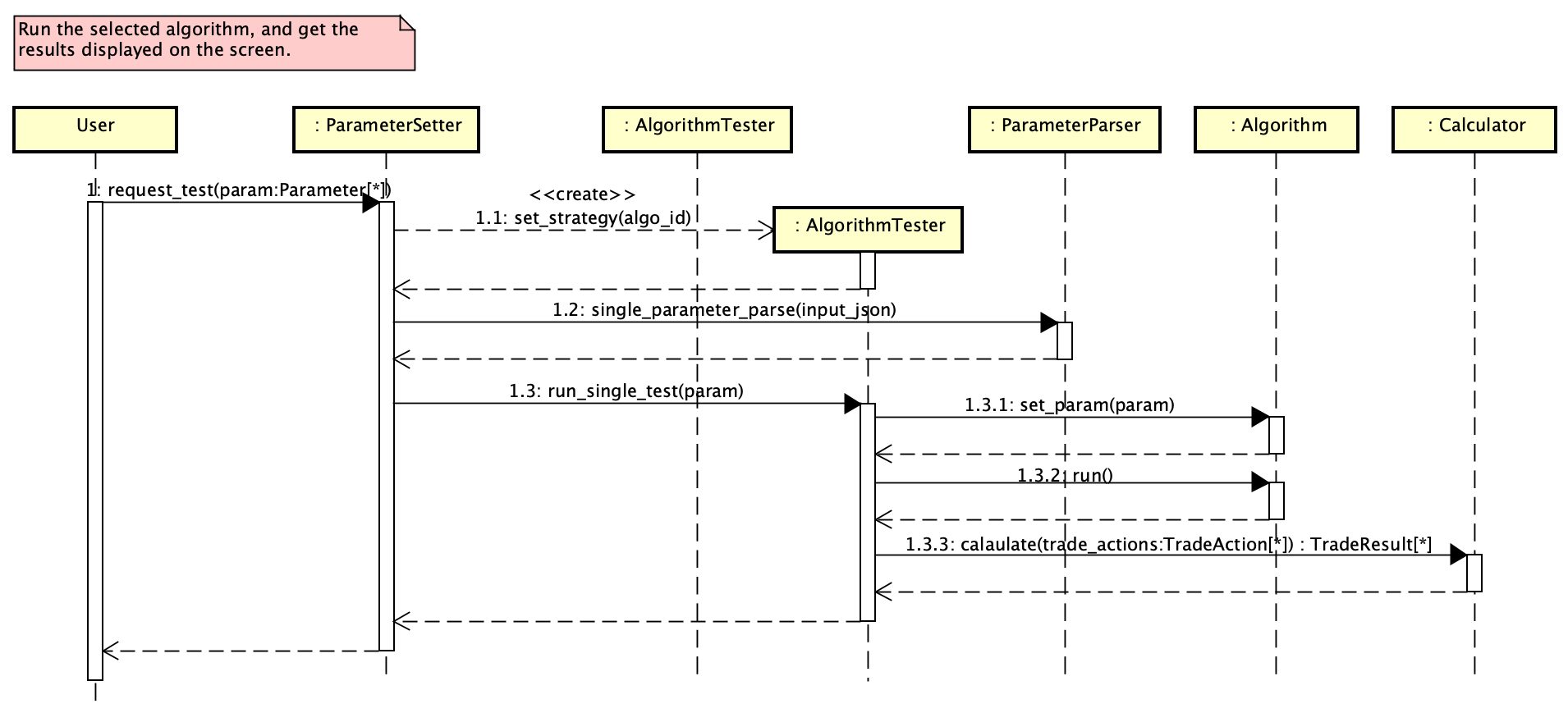
## 4.2 類別圖

# 5.循序圖設計(Sequence Diagram)

1. Upload and select an algorithm



1. Single test an algorithm



1. Batch test an algorithm

# 

# 

# 6.資料細部設計(Data Design)

* TradeAction

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ProductID | Int |  |
| DateTime | datetime |  |
| Pos | Int | +: buy; -: sell |
| Price | Float |  |
| Tag | String |  |

* TradeResult

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ProductID | Int |  |
| DateTime | datetime |  |
| IsLong | Bool | True: buy; False: sell |
| IsEnter | Bool | True:進場 False:出場 |
| Price | Float |  |
| Tag | String |  |
| Profit | Float |  |

* Product

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ID | Int |  |
| Name | String |  |
| TickSize | Float | 最小跳動點 |
| Unit | Float | 每點金額 |
| ExchangeRate | Float | 匯率 |

* TestResult

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ID | Int |  |
| TradeResults | TradeResult[] | TradeAction with Profit |
| TradeStats | TradeStat[] | 用在寫表 |

* TradeStat

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| Stat | String | 期望值 |
| Value | Float | 87.1 |

* ParameterSet

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ID | Int |  |
| Algo\_ID | Int |  |
| Parameters | ParameterValue[ ] |  |

* ParameterValue

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ID | Int |  |
| Type | String |  |
| Value | Object |  |

* AlgoInfo

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| ID | Int |  |
| Title | String |  |
| Version | String |  |
| Path | String | filename:Title+Version |

* ReportInfo

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| Algo\_ID | Int |  |
| Path | String |  |

* AssetData

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Note |
| Time | datetime |  |
| Contract | String |  |
| Open | Float |  |
| High | Float |  |
| Low | Float |  |
| Close | Float |  |
| Volume | Int |  |
| Delta | Float |  |