**HSBC Fund Rebalancing Project**

Internal Design Document

4/1/2019

Version 1.3

**Team REST**

Aurora (Yutian) Qiu

Cathy Leung

Leo Cheng

Leo (Jixing) Li

Shaw (Xiao) Lu

Stanley Ye

Tanya (Yi) Tan

## **Table of Contents**

[**Table of Contents**](#_y36sphi1asmk) **2**

[**Revision History**](#_rcm0w0hpldll) **4**

[**Stakeholders**](#_d4zcu9kv655n) **4**

[**Introduction**](#_8aklkefdxewj) **5**

[System Overview](#_yud39ffewtj7) 5

[System Goals](#_qcwzail3z0n1) 5

[System Assumptions](#_anprxbls03vh) 5

[**Programming Environment**](#_3bmw1stfhrt9) **6**

[Programming Tools / Framework / Environment](#_9j9my29a6q1r) 6

[Programming Languages](#_buqca4oh1suw) 6

[Source Control System](#_1a1wr1z7tb16) 6

[Infrastructure](#_q768eobdqqku) 6

[**Production and Test Environments**](#_ncybh44rd5ni) **6**

[Production environment](#_plb28vkas4zb) 6

[Test environment](#_amw6t4ptpr8o) 6

[**Data Design**](#_wgh8keb7mjqy) **7**

[Entity Relationship (ER) Diagram](#_b4ygcqo447a) 7

[**Software Architecture Design**](#_2uu95s4ge233) **7**

[**API Design**](#_fqdaaweapozn) **8**

[POST: /portfolio/{id}](#_d8n2aaxzjr89) 8

[GET: /portfolio/{id}](#_8q3dzn1delho) 8

[PUT: /portfolio/{id}/allocations](#_c59z8c6bfef6) 9

[PUT: /portfolio/{id}/deviation](#_rn4hx4v3xslg) 9

[POST: /portfolio/{id}/rebalance](#_wrwb6tvk5nm1) 10

[POST: /portfolio/{id}/recommendation/{recommendation\_id}/execute](#_pkvipe8d1xgd) 11

[PUT: /portfolio/{id}/recommendation/{recommendation\_id}/modify](#_mrx17bytsxbv) 11

[**Algorithms**](#_5k2lb13zww4b) **12**

[Fund Scoring Algorithm](#_hq1hbhik00t2) 12

[**Notable Trade-offs**](#_86f0mqvn0oe) **13**

[Load Testing](#_r4jk9yijwil8) 13

[**Notable Risks**](#_rczwt4fyqgjn) **13**

[Mitigation Strategies](#_1gwqzppsvkit) 14

[**UI Design**](#_tuqywol70j0) **14**

[**References**](#_sc3nak8jflfg) **19**

## 

## **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **DATE** | **Name** | **Description of Change** |
| 1.3 | 4/1/2019 | Team REST | Final version |
| 1.2 | 2/10/2019 | Team REST | Updated ERD and modified API design |
| 1.1 | 2/1/2019 | Team REST | Modified API design, redefined the fund scoring algorithms for the stretch goal, updated ERD and UI |
| 1.0 | 1/30/2019 | Team REST | Initial draft |

## **Stakeholders**

Categories:

1. Agree with contents
2. Agree, subject to incorporation of comments
3. Disagree, comments included

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approver Name/Title** | **Signature** | **Sign Date** | **Subject to Category** | **Comments** |
| Wilson Fung |  |  |  |  |
| Clifford Lee |  |  |  |  |
| Jerry Jim |  |  |  |  |

## 

## **Introduction**

### **System Overview**

The HSBC Fund Rebalancing Service is a collection of REST APIs that provide portfolio rebalancing to HSBC clients. The system allows clients to track their investments without going to a branch manager, manage their mix of funds, and protect clients from over-concentrating their risks due to a market shift.

### **System Goals**

The HSBC Fund Rebalancing Service is aimed at providing HSBC clients with an improved experience of micro-investment management by:

* Reducing time and inefficiency for HSBC clients to perform micro-investment management on their portfolios.
* Increasing intelligence, convenience, and visibility of fund management.
* Minimizing human resource investments from HSBC.
* Improving communication between HSBC and its clients by providing more transparent investment advisory.
* Revolutionizing HSBC’s RoboAdvisor product and thus increasing HSBC’s brand awareness.

### **System Assumptions**

* MySQL Database
  + A database instance will be created on GCP to store preference and recommendation as well as retrieve data stored in the Mock System.
  + The portfolio entries and size will not exceed the maximum storage size of 500 G based on GCP Cloud SQL Pricing.
* Hosting
  + The fund rebalancing service will be hosted on GCP with free credit provided.
* API
  + Reasonable concurrent server load that is able to scale based on the number of clients who will be using the service.
  + Our API methods will deal with CAD currency for all funds.

## 

## **Programming Environment**

### Programming Tools / Framework / Environment

* Spring Boot 2.0
* GitLab CI
* Postman
* JUnit 5
* MySQL 8.0
* Gson 2.8.4
* Intellij IDEA 2016.2.4

### Programming Languages

* Java v1.8
* JavaScript
* HTML5/CSS3

### Source Control System

* GitLab

### Infrastructure

* Google Cloud Platform

## 

## **Production and Test Environments**

### **Production environment**

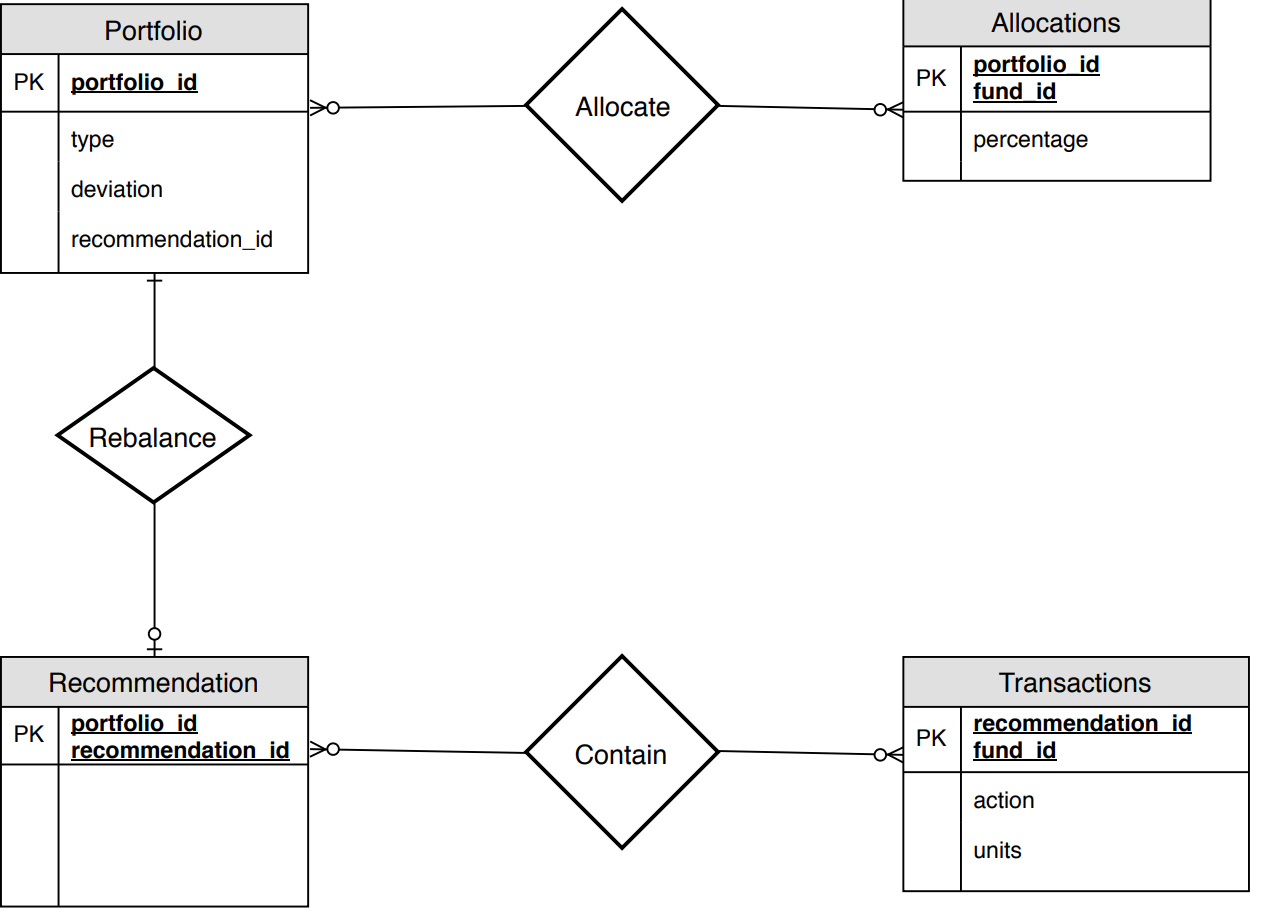
The REST APIs system and a database will be hosted on Google Cloud Platform (GCP).

### **Test environment**

* Test data: Our project uses a MySQL database to store the data. We will make use of the sample data provided by HSBC and we will also design sample data ourselves for testing purpose.
* JUnit will be used for unit testing.
* Postman will be utilized for API testing and debugging.

## **Data Design**

### **Entity Relationship (ER) Diagram**



* The Allocations table has a many-one relationship to the Portfolio table. Each portfolio can allocate many funds and an allocation has to be in a portfolio.
* The Transactions table has a many-one relationship to the Recommendation table. Each recommendation can contain many transactions and a transaction has to be in a recommendation.
* The Recommendation table has a one-optional relationship to the Portfolio table. Once a recommendation is executed, then it rebalances a portfolio. A portfolio can have either 1 recommendation to rebalance it or none recommendation. However, a recommendation can only rebalance one portfolio.

## 

## **Software Architecture Design**

Since the rebalance service system contains seven API endpoints, we have decided to put all the API endpoints into one class.

## **API Design**

All the API endpoints in the rebalance service system require a customer (client) id in the header and apply to “type: fund” portfolios only.

In the stretch goal, the system will extend these API endpoints to handle “type: category” portfolios and recommendation modification feature, thus “PUT: /portfolio/{id}/recommendation/{recommendation\_id}/modify” will be added. This endpoint applies to “type: category” portfolios only.

### POST: /portfolio/{id}

|  |  |
| --- | --- |
| HTTP Method | POST |
| Endpoints | /portfolio/{id} |
| Request Body | {  deviation: n,  type: fund,  allocations: [{fundId: n, percentage: nn}, { fundId: n, percentage: nn}, …]  } |
| Response Body | Empty |
| Action | This method records how clients want to balance their portfolios for the first time |
| Acceptance Criteria | * If the request succeeds, create an entry in the portfolio table of the database with the id as the primary key and return {200 OK} * If there are invalid inputs, return {400 Bad Request} * If internal server error, return {500 Internal Server Error} |
| Engineering Note | * The percentage from allocations should sum to 100, otherwise it is invalid input data. * Sample data about funds can be retrieved from HSBC API: GET /fund/{id} |

### GET: /portfolio/{id}

|  |  |
| --- | --- |
| HTTP Method | GET |
| Endpoints | /portfolio/{id} |
| Request Body | Empty |
| Response Body | {  id: n,  deviation: n,  type: fund,  allocations: [{fund\_id: n, percentage: nn}, { fund\_id: n, percentage: nn}, …]  } |
| Action | This method retrieves the asset mix preference for a portfolio whose id is {id} |
| Acceptance criteria | * If the request succeeds, return {200: response body} that contains the client’s asset mix preference * If there are invalid inputs or no data is found from the Portfolio table in the database, return {4XX: error message} |
| Engineering note | None |

### PUT: /portfolio/{id}/allocations

|  |  |
| --- | --- |
| HTTP Method | PUT |
| Endpoints | /portfolio/{id}/allocations |
| Request Body | [{fund\_id: n, percentage: nn}, { fund\_id: n, percentage: nn}, …] |
| Response Body | [{fund\_id: n, percentage: nn}, { fund\_id: n, percentage: nn}, …] |
| Action | This method updates the asset mix preference allocations against a portfolio whose id is {id} |
| Acceptance criteria | * If the request succeeds, return {200: response body} and update the data entry whose primary key is “/portfolio/{id}” in the database with new allocations * If there are invalid inputs, return {4XX: error message} |
| Engineering note | * The percentage from allocations should sum to to 100, otherwise, it is invalid input data. |

### PUT: /portfolio/{id}/deviation

|  |  |
| --- | --- |
| HTTP Method | PUT |
| Endpoints | /portfolio/{id}/deviation |
| Request Body | {deviation: n} |
| Response Body | {deviation: n} |
| Action | This method sets the allowed deviation percentage for a portfolio whose id is {id} |
| Acceptance criteria | * If the request succeeds, return {200: request body} and update the data entry whose primary key is “/portfolio/{id}” in the database with new deviation * If there are invalid inputs or no data is found from the Portfolio table in the database, return {4XX: error message} |
| Engineering note | * The valid deviation percent is from the asset mix: [0-5%], otherwise, it is invalid input data. |

### POST: /portfolio/{id}/rebalance

|  |  |
| --- | --- |
| HTTP Method | POST |
| Endpoints | /portfolio/{id}/rebalance |
| Request Body | Empty |
| Response Body | {  recommendation\_id: n  transactions:[  {  action: buy|sell,  fund\_id: n,  units: n,  },  {action: buy|sell, … }  ]  } |
| Action | This method returns a set of transactions that rebalances the client’s portfolio allocations back to the initially preferred percentages.  The business logic to generate the set of transactions for each fund under a specific portfolio is as follows:   * If a fund exceeds its initial preference percentage, then the method recommends selling certain units of this fund. * If a fund has lower percentage than initial preference percentage, then the method recommends to buy more units of this fund. |
| Acceptance criteria | * If the request succeeds, return {200: response body} that contains an unique recomendation\_id and a set of recommendations * If invalid inputs, return {4XX: error message} |
| Engineering note | * Sample data about funds can be retrieved from HSBC API: GET /fund/{id} |

### POST: /portfolio/{id}/category\_rebalance

|  |  |
| --- | --- |
| HTTP Method | POST |
| Endpoints | /portfolio/{id}/category\_rebalance |
| Request Body | Empty |
| Response Body | [{  recommendation\_id: n  transactions:[  {  action: buy|sell,  fund\_id: n,  units: n,  },  {action: buy|sell, … }  ]  }, ...] |
| Action | This method returns multiple possible recommendations that will rebalance the client’s portfolio allocations back to the initially preferred category percentages.  The business logic to generate the set of transactions for each category under a specific portfolio is as follows:   * If a category exceeds its initial preference percentage, then the method recommends selling certain units of funds within this category starting with the lowest scored funds. * If a category has lower percentage than initial preference percentage, then the method generates three recommendations that buy different funds within the specified category.   + Firstly the method will suggest buying the fund with the highest score   + Secondly the method will suggest buying a currently owned fund within the category with the highest score   + Lastly the method will suggest buying a fund with the lowest unit price to achieve the closest allocation to the target |
| Acceptance criteria | * If the request succeeds, return {200: response body} that contains an unique recomendation\_id and a set of recommendations * If invalid inputs, return {4XX: error message} |
| Engineering note | * Sample data about funds can be retrieved from HSBC API: GET /fund/{id} |

### POST: /portfolio/{id}/recommendation/{recommendation\_id}/execute

|  |  |
| --- | --- |
| HTTP Method | POST |
| Endpoints | /portfolio/{id}/recommendation/{recommendation\_id}/execute |
| Request Body | {recommendationId: n} |
| Response Body | empty |
| Action | This method executes the list of recommended rebalance transactions with {recommendation\_id} |
| Acceptance criteria | * If the request succeeds, execute the set of transactions in recomendation\_id and return a response code 200 * If there are invalid inputs or no data is found from the Recommendation table in the database, return {4XX: error message} |
| Engineering note | * This method will need to call HSBC API POST /transaction method. |

### PUT: /portfolio/{id}/recommendation/{recommendation\_id}/modify

|  |  |
| --- | --- |
| HTTP Method | PUT |
| Endpoints | /portfolio/{id}/recommendation/{recommendation\_id}/modify |
| Request Body | [  {  action: buy|sell,  fundId: n,  units: n,  },  {action: buy|sell, … }    ] |
| Response Body | {  recommendationId: n  transactions:[  {  action: buy|sell,  fundId: n,  units: n,  },  {action: buy|sell, … }    ]  } |
| Action | This method modifies the recommendation whose recommendation\_id is provided through the API parameter. It returns a new set of transaction actions based on the changes. |
| Acceptance Criteria | * If the request succeeds, modify the set of transactions in recomendation\_id based on the funds selected and return response code 200 with the response body * If there are invalid inputs or no data is found from the Recommendation table in the database, return {4XX: error message} |
| Engineering Note | * An example: If I want to buy 200 less unit of fund id: 255, then you would have an entry like this: {action: buy, fund\_id: 255, units: -200} |

## **Algorithms**

### Fund Scoring Algorithm

In the stretch goal, the POST “/rebalance” API method will be extended to recommend a set of transactions for “type: category” portfolios besides “type: fund” portfolios. There are multiple funds available under each category. To rebalance a “type: category” portfolio, the API method will recommend selling some units of fund for a category whose allocation exceeds its initial percentage and to buy some units of the top scored fund for a category whose allocation is lower than its initial percentage. The Fund Scoring algorithm will be used by this API method.

The algorithm will score the available funds under a category based on the following equation and return the top scored fund for that category.

Score for a fund = 5 years performance + 10 years performance + 3 years performance + 1 year performance + GICS Sectors Trend 1 year performance

* 5 years performance = 5 yrs perf weight x 5 yrs avg return
* 10 years performance = 10 yrs perf weight x 10 yrs avg return
* 3 years performance = 3 yrs perf weight x 3 yrs avg return
* 1 year performance = 1 yrs perf weight x 1 yrs avg return
* GICS[[1]](#footnote-0) Sectors Trend 1 yr performance = Sectors performance x sectors weight

The return rate will be retrieved from fund details through the HSBC API “/fund/{id}” method and the weights will be manually configured based on HSBC sponsors requirements.

## **Notable Trade-offs**

### Load Testing

In light of the nature of the application, we have determined that the likelihood of receiving “Error 429 Too Many Requests” is low. In addition, the application is designed to ensure scalability as HSBC can adopt horizontal scaling to handle increasing requests in the future. In consideration of this information and our constrained resources, we will not be doing extensive load testing. Instead, we will focus on manual user testing by sending API requests through Postman.

## **Notable Risks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk ID** | **Risk Description** | **Severity** | **Likelihood** |
| 1 | Google Cloud Platform (GCP) becomes unavailable | High | Low |
| 2 | The mock system not responding to our calls or returning invalid data | Medium | Medium |
| 3 | The source control system, GitLab, becomes unavailable | Medium | Low |
| 4 | API testing tool, Postman, becomes unavailable | Medium | Low |

### 

### Mitigation Strategies

|  |  |
| --- | --- |
| **Risk ID** | **Risk Mitigation** |
| 1 | Leverage our baseline contact with HSBC / Google to get an estimate on downtime although issue likelihood is rather low |
| 2 | Contact the mock system provider, HSBC, to get the system up and running correctly |
| 3 | Potentially switch to a different source control system temporarily to continue development |
| 4 | Switch to another API testing tool such as SoapUI temporarily to continue testing |

## 

## **UI Design**

A simple UI will be designed for demo purposes. Once the user logs in and selects a portfolio, they are able to view the current fund allocations with the option to create an initial preference as a button below the data. If the user chooses to set the initial preference, a pop-up window will appear for the user to input their values. Once the preference has been determined, the UI will be updated and the options for update deviation, preference, and rebalance will become available. If the user clicks the rebalance button, they will be brought to a new screen that shows the recommended transactions needed for rebalancing with the option to modify the given transactions or to execute the transactions. As the stretch goal has been implemented, there will be different dropdowns and input fields based on whether the portfolio is fund-type or category-type. Also rebalancing category type portfolios yield multiple recommendations that will help rebalance the user’s portfolio allocations.

## **Figure 1**

## **References**

1. HSBC RoboAdvisor Fund Rebalancing Project Document   
   <https://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fjo4zqfjte9v12v%2Fj7b3tb7ez6c2qi%2Fjr9sikfpig5q%2FHSBC__RoboAdvisor__Fund_rebalancing_Service_UBC_v_1.4.doc>
2. GCP Cloud SQL Pricing

https://cloud.google.com/sql/pricing#packages

1. GICS: There are 11 GICS sectors. More information can be reffered to https://www.barchart.com/ca/stocks/sectors/rankings?timeFrame=1y [↑](#footnote-ref-0)