Asset Allocation

March 2023

Table of Contents

1 Asset Allocation

Asset Allocation is a significant activity of a Fund manager. Purpose of this exercise is as follows:

- Familiarize trainees with basic Securities domain, introducing to Asset classes; guardrails used in funds; NAVs;
- Trainees will create rich User interfaces (using Angular or React); Services in (.NET or Spring).

1.1 Objective

Developing an Asset Allocation module as part of a larger Portfolio Management product. The Asset Allocation module will consider investment themes, factors portfolio risk tolerance, investment goals, and time horizon to recommend a portfolio mix across different asset classes calculate the portfolio value and alerts for rebalancing.

1.2 High level Specifications

High-level specification is provided as a reference. We have identified a Minimum Viable Product (MVP) under the 'Scope of work' section in this document.

A high-level specification for the Portfolio Management will involve the following.

- **Determining the investment objectives:** The first step is to determine the investment objectives This includes defining the risk tolerance, investment horizon, and return expectations from different assets.
- **Selecting asset classes:** Based on the investment objectives and market analysis, the fund manager selects the appropriate asset classes for the portfolio. Popular asset classes include equities, fixed income (bonds and other debt securities), cash etc.
- **Setting asset allocation targets:** Once the asset classes are selected, the fund manager sets asset allocation targets for each asset class. The targets are typically based on historical performance, risk characteristics, and the investor's objectives. For example, a mutual fund manager might set a target of 60% equities, 30% fixed income, and 10% cash.
- **Implementing the portfolio:** The fund manager then implements the portfolio by selecting specific securities and investments within each asset class, within the equities asset class, the manager might choose to invest in large-cap, mid-cap, or small-cap stocks in different sectors such as technology, healthcare, or consumer goods.
- **Monitoring & Rebalancing:** Finally, the fund manager monitors the portfolio's performance and periodically rebalances the asset allocation to maintain the desired level of risk and return. For example, if the total allocation has differed by 5% than the original model, the manager needs to be alerted so suitable action can be taken

1.3 Scope of work:

The project focusses on developing an application which will be used by a Mutual Fund manager working for a large investment firm. The investment firm has collected a large sum of money (Rs.500 crores) from its investors and nominated a Fund manager. The Fund manager will use our Portfolio Management application to create the Portfolio, add securities to it and start monitoring it. Based on price movements of the underlying securities in the Portfolio, the Fund manager will rebalance his portfolio as per the Rebalancing Frequency defined.

Our application will use the Security Master created by the Firm and choose Securities made available by the Firm. The Fund manager will keep the Investment Themes (eg. Equities 60%; Bond 40%) in mind while building the Portfolio. The fund manager will be able to compare the Portfolio against Investment theme goals defined for the Fund and rebalance.

Allocation may be made in terms of the size of the security (large-cap, mid-cap, or small-cap), industry (Oil, Energy, IT, Banks) etc

1.4 Use cases for MVP

1 Creation of Portfolio Header and Theme Selection

1.1. Portfolio Management (Home Page):

- List Portfolio - On the main page, the Fund manager will be able to see all portfolios created / accessible by them.
- Add Function – Should have an add button which will allow the user to create a new portfolio. The application will launch 'create portfolio' screen.

1.2. Creating the Portfolio –

To create a portfolio the following parameters should be added –

- Name of the portfolio
- Type of Portfolio (Mention if allocations will be done by either weightage or amount)
- Currency of the portfolio (INR)
- Selecting Default Benchmark (for comparison) – e.g.- S&P BSE Sensex India, IISL Nifty 100 PR INR, MSCI India SMID GR INR etc.
- Select Exchange – NSE or BSE
- Select the theme
- Selecting the rebalancing Frequency – Daily, Monthly, Quarterly, every 6 months, Yearly
- Amount to be invested
- Saving the portfolio – Once the details are filled it should be saved and displayed on the main dashboard of the user with the following parameters – Name, Type, Currency, Number of Holdings and Benchmark set.
- Backend Functionality: Should store the above portfolio details in a table named 'Portfolio Details' having unique Identification and column names should be same as that of the parameters

1.3. Setting the Theme

There should be an option to select a model for every portfolio to follow, on selecting this option the different models should be displayed with the following parameters –

- A brief description of the model
- Allowed allocation limits (e.g. equities = 85%, cash = 5%, Bond = 10% for very aggressive model)
- Risk level
- Investment horizon

The models to be included are –

Conservative, Moderately Conservative, Aggressive, Moderately Aggressive & Very Aggressive

The user can simply select the model that needs to be followed using a dropdown

- Backend Functionality: The name of the theme selected should get saved in the portfolio details table under 'Theme' Column
- Allocation % for a Theme has to add up to 100%

Example of Aggressive Allocation:

Name	Asset Class	Allocation%	Risk	Investment Duration
	Equities (Domestic & International / Value & Growth Companies)	70.0%		
Aggressive	Alternative Investments	20.0%	High	Long Term
	Commodities	10.0%		

2 Portfolio Composition

2.1 Selection of Securities

- From a drop-down menu which will have a search option enabled to select different securities to add to the portfolio
- Once the security is selected its Name, Asset class, Sub-asset class, Price, Equity category (Large-cap, Mid-cap or Small-cap) should be displayed and have enterable fields of – Quantity from which the calculated price should be displayed
- The count of all the stocks and value of portfolio (sum of stocks) should be displayed at the top as new securities are being added.
- UI function: To carry out this operation Grid component should be used for adding and maintaining the securities.
- This will be a CRUD operation – add, delete, and edit option should be enabled for each security that is being added.
- Backend Functionality: Once each security is added along with the quantity and calculated price it will get saved to a new table named 'Assets & Holdings Details' including all the parameters mentioned above.
- The CRUD operation on the table done from the UI (add, edit, delete) should simultaneously reflect in the database.
- For linking the tables - This table (Assets & Holdings Details) will have a foreign key referring to the primary key of the 'Portfolio Details' table

2.2 Setting Target Allocation

- Display the Asset classes, equity category and assign the weightage/amount to the holdings w.r.t the model selected so the total weightage of the portfolio can come up to 100%

3 Assessing the Portfolio

- Comparison - The target allocation done of different asset classes needs to be compared with the theme defined initially for the portfolio. For this the theme selected while creating the portfolio as well as the current portfolio's asset classes needs to be displayed for comparison
- Rebalancing the Portfolio - When the portfolio's asset classes allocation deviates from the defined theme's asset classes allocations by more than 5% in total the application sends an alert to the fund manager informing the same.

3.1 Team Deliverables

- Data Model for Portfolios, Investment Themes
- UI wireframes
- Data Flow diagram
- High level architecture diagram
- API Contracts (exposed for other applications)
- User interface to Create / View/ Modify Portfolio.

3.2 Tech Stack

- NET or Java for Backend services
- Angular or React for UI
- Postgres or MySQL (H2 for development)
- Code Quality - Sonarlint / Sonarqube
- Code Repository – TBD (will be provided)
- Testing – Junit / Nunit;
- UI Testing - Selenium

3.3 Assumptions

- We will assume Application users are all internal users and have been authenticated through an existing AD.
- Teams will follow Agile Scrum and iterate in 2-week sprints.
- Demo to Business stakeholders (ASDMs / Project managers) after every Sprint.
- Demo's will be presented by 2 members. It will be recorded by the team.

3.4 Wireframes

Some sample wireframes is shown below. Please refer to the Excel 'Portfolio – UI' for additional information on the wireframes and datamodel.

Portfolio Header									
Portfolio Name	<<Enter Unique Name>>		Choose Theme of Investment	<<Choose values from dropdown of Themes>>					
			Fund Manager Name	<<Enter Name of Fund manager>>					
Base Currency	<<INR / USD / GBP>>		Initial Investment	<<500 Crores>>					
Exchange	<<Nasdaq>>								
Benchmark	<<Default Benchmark for compari		Rebalancing Frequency	<<daily / monthly / quarterly>>					
							<<Create>>	<<Reset>>	
							Buttons	Buttons	
Portfolio Details (Composition)									
Security Name	Asset Class	SubAsset Class	Equity Category (not for Bonds	Security Price	Qty	Value			
<<Dropdown>>	<<displayed>>	<<displayed	<<displayed>>	<<displayed>>	<<entered>>	<<displayed>>		Add Edit	Delete
Stock2			<<MediumCap>>	50	5	3		Add Edit	Delete
Stock3			<<LowCap>>	1000	2	2.5		Add Edit	Delete

3.5 Appendix - List of User Interfaces:

Consultants are encouraged to come up their own feature list based on requirements provided above. The below list of user interfaces is given to jump start. Please add / change this list based on the features you plan to build.

- Screen - Asset Classes - CRUD screen
- Screen - Security Master - Existing functionality; Will have a drop down for Securities.
- Screen - Investment Themes - CRUD screen
- Screen – Portfolio
    - Portfolio Header (Name, Portfolio manager, Initial Investment value, Benchmark etc)
    - Assign Themes (Guardrail - validate if allocation does not exceed Theme limit)
    - Add Securities (accept #shares, lookup price from Security master; calculate Value = #Shares * Price)
    - Initially "Available Balance" will be same as "Initial Investment Value"; For every Security added, reduce value of "Available Balance".
    - If Securities are 'sold' then the Value is added back to the "Available Balance".
    - Validate if the sum of all Securities do not exceed Allocation% defined as per Theme mapped. For example if Theme 'Aggressive' which has 90% Equity allocation is mapped to Portfolio. If Portfolio Initial investment is 50Cr, the sum of all Securities added in Portfolio cannot exceed 45 Cr (90% of 50 Crores).
    - Please note the value of the Portfolio will change based on daily stock prices and hence rebalancing a Portfolio is a normal activity for Portfolio managers. The rebalancing will happen as per the Rebalancing Frequency defined. For MVP this is only a manual process.
- Backend - Portfolio Status
    - Portfolio Status will be maintained in the Portfolio Header.
    - Portfolio Status will be "Display" only.

- Portfolio Status can be "New", "Live", "Terminated"
- When new Portfolio is created 'Status' is 'New'.
- Initial Value is editable until Status is 'New'.
- Once Portfolio is created (all securities added), application will provide for Portfolio to "Go Live". THe status will change from "New" to "Live".
- Once it is "Live" the Portfolio is baselined. Subsequent changes to Portfolio value (when Security prices change) will be reflected as Performance of Portfolio.

- Screen - To compare Current Allocation Vs Theme. Compare Portfolio current allocation % and Asset Class % in Theme mapped to Portfolio.

- List of Values in Screens (Dropdowns). Build appropriate backend APIs to support these dropdowns.
  - List of Securities
  - List of Themes
  - List of Asset Classes

Appendix - Important Terminologies

- Portfolio Management - Investment portfolio management involves building and overseeing a selection of assets such as stocks, bonds, and cash that meet the long-term financial goals and risk tolerance of an investor
- Asset allocation – It is an investment strategy that aims to balance risk and reward by apportioning a portfolio's assets according to an individual's goals, risk tolerance, and investment horizon. The three main asset classes—equities, fixed-income, and cash and equivalents—have different levels of risk and return, so each will behave differently over time
- Asset Class - An asset class is a grouping of investments that exhibit similar characteristics and are subject to the same laws and regulations. Equities (e.g., stocks), fixed income (e.g., bonds), cash and cash equivalents, real estate, commodities, and currencies are common examples of asset classes
- Portfolio Rebalancing – It refers to the process of returning the values of a portfolio's asset allocations to the levels defined by an investment plan. Those levels are intended to match an investor's tolerance for risk and desire for reward. Rebalancing involves periodically buying or selling the assets in a portfolio to regain and maintain that original, desired level of asset allocation.

thank you

www.hexaware.com

HEXAWARE