|  |  |
| --- | --- |
|  | **<Project Name>**  **Technical Design Document** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** |  |  |  | | **Role** |  |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc14171023)

[1.1 Purpose of this document 3](#_Toc14171024)

[1.2 Project overview 3](#_Toc14171025)

[2.0 Solution Summary 3](#_Toc14171026)

[2.1 Scope 3](#_Toc14171027)

[2.2 Assumptions 3](#_Toc14171028)

[2.3 Dependencies 3](#_Toc14171029)

[2.4 Risks 3](#_Toc14171030)

[3.0 Schematic Diagram 3](#_Toc14171031)

[4.0 System Design 4](#_Toc14171032)

[4.1 Proposed design 4](#_Toc14171033)

[4.2 Component inventory 4](#_Toc14171034)

[5.0 Database Design 4](#_Toc14171035)

[5.1 Data Model 4](#_Toc14171036)

[5.2 Tables Structure 4](#_Toc14171037)

[6.0 Appendices 4](#_Toc14171038)

[6.1 Glossary 4](#_Toc14171039)

[6.2 Other 5](#_Toc14171040)

[7.0 Terms & Conditions 5](#_Toc14171041)

[8.0 Change Log 5](#_Toc14171042)

# Introduction

## Purpose of this document

The purpose of this document is to document the technical design, component details and Database design. This will also capture the scope, assumptions, risk, dependencies of this project.

## Project overview

The Mutual Fund Management System facilitates portfolio management for a Portfolio Manager (PM) and enables investors to buy and redeem mutual fund units. It ensures efficient handling of funds, portfolio positions, and transactions.

# Solution Summary

## Scope

The system allows the PM to create and manage mutual funds, while investors can view available funds, invest, redeem units, and access their portfolio information.

**Portfolio Management by PM:**

* PMs can create and define mutual funds by specifying fund details such as name, stock composition, weightage, cash balance, entry/exit loads, and expense ratio.
* PMs have visibility into the total corpus of the fund, portfolio positions by each stock, and cash position in real-time.

**Investor Interaction:**

* Investors can access the system to search for available mutual funds for subscription or redemption.
* Investors have the capability to buy mutual funds for a specified investment amount and redeem mutual fund units from their portfolios.

**Transaction Management:**

* The system facilitates investment and redemption transactions based on the current day's closing NAV.
* Proper checks are implemented to ensure investors cannot sell more units than they own, thereby maintaining transaction integrity.

## Assumptions

* Assume an initial corpus of INR 100 crores.
* Only one PM exists.
* Dummy institutional investors participate in New Fund Offers (NFOs).
* Share buy/sell operations occur in the background.
* Cash transfers between investors and funds occur in the background.

## Dependencies

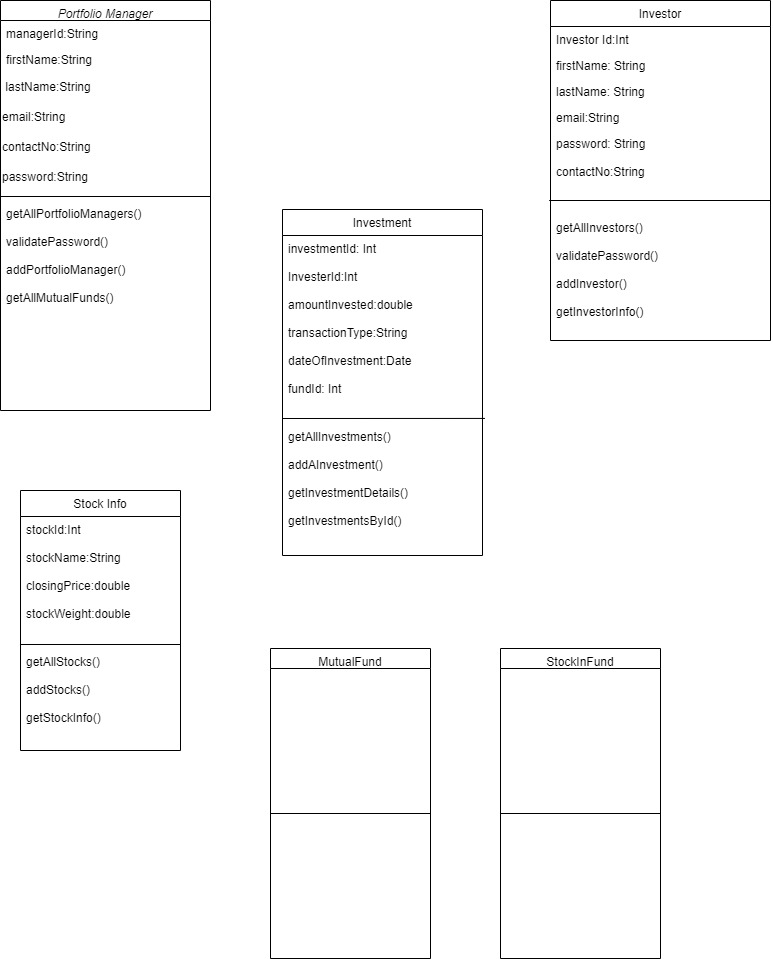
1. **JAVA Spring Boot:**
   * A powerful Java-based framework for building robust, scalable, and efficient web applications.
   * Spring Boot simplifies development, enhances productivity, and provides essential features out of the box.
   * **Used for:** Backend development, business logic, and RESTful APIs.
2. **Hibernate:**
   * An Object-Relational Mapping (ORM) framework that bridges the gap between Java objects and relational databases.
   * Simplifies database interactions, handles data persistence, and ensures efficient queries.
   * **Used for:** Data access layer, interacting with the MySQL database.
3. **MySQL:**
   * A popular open-source relational database management system.
   * Stores product information, user profiles, order history, and other essential data.
   * **Used for:** Storing and retrieving data efficiently.
4. **HTML, CSS, and Bootstrap:**
   * HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) create the structure and style of web pages.
   * Bootstrap, a front-end framework, provides responsive design components and pre-styled elements.
   * **Used for:** Creating user interfaces, responsive layouts, and consistent styling.
5. **JavaScript (JS):**
   * A versatile scripting language used for enhancing interactivity on web pages.
   * Enables dynamic features like real-time search, cart updates, and form validation.
   * **Used for:** Front-end interactions, client-side scripting.

## Risks

* Data security risks.
* Integration challenges with external systems.
* Regulatory compliance risks.

# Schematic Diagram

A schematic, or schematic diagram, is a representation of the elements of a [system](https://en.wikipedia.org/wiki/System) using abstract, graphic [symbols](https://en.wikipedia.org/wiki/Symbol) rather than realistic pictures. It gives an overview of overall system



# System Design

## Proposed design

The system comprises modules for PM management, investor management, fund subscription/redemption, NAV calculation, and transaction tracking.

**For Investors**

* + Intuitive Fund Exploration: Discover and analyze diverse mutual funds with there returns.
  + Transactions at the backend
  + Easy Registration & Login : Streamlined onboarding with secure access.Buy/sell with ease and instant confirmations.
  + Insightful Portfolio Dashboard : Real-time tracking and graphical insights.

**For Manager**

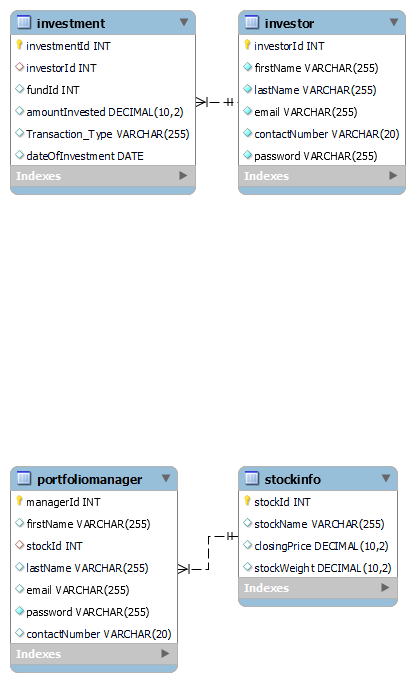
* + Secure Login : Streamlined onboarding with secure access.
  + Efficient Fund Management : Easily update fund details for accuracy.
  + Create Fund : Can create funds by selecting the shares and their weightage.
  + Customizable Parameters : Set expense ratio, entry load, and exit load.

## Component inventory

* User interfaces for PM and investors.
* Database for storing fund and investor information.
* Integration with external systems for NAV calculation and fund transfers.

# Database Design

## Data Model



## Tables Structure

*<This sub section will describe the table structure>*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Appendices

## Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| JS | Javascript |
| JAVA | A general-purpose, object oriented language |
| SPRING BOOT | A framework for building web apps and service |
| MYSQL | Microsoft’s Relational Database Management System |
| PM | Portfolio Manager |
| MF | Mutual Fund |
| NFO | New Fund Offer |
| NAV | Net Asset Value |

# Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of Cognizant Network, We have a Zero Tolerance Policy. Kindly adhere to 100% Compliance at all times.***

# Change Log

*Please note that this table needs to be maintained even if a Configuration Management tool is used.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Number | Changes made | | | |
| V<n.n> | *<If the change details are not explicitly documented in the table below, reference should be provided here>* | | | |
| Page no | Changed by | Effective date | Changes effected |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |