Software Requirement Specifications (SRS)

For

WealthMitra(WM)

By: Star Catchers

- Aakanksha Sonawane
- Shubham Premi
- Sampada Pote
- Gayatri Jadhav
- Umang Jain
- Kanaad Rampurkar
- Pravesh Khandelwal
- Avip Patil
- Niteshkumar Singh

Table of contents

Introduction

1.1 Purpose:

WealthMitra (WM) is intended to help the user keep account of his/her moneyinvested in institutions such as Share Market.

This document is meant to delineate the features of WM, to serve as a guide to the developers on one hand and a software validation document forthe prospective client on the other.

1.2 Scope:

We describe what features are in the scope of the software and what are not inthe scope of the software to be developed.

In Scope:

- Managing investment of a single user, which would include maintainingbookkeeping information about entities like Portfolio, Security, and Transaction.
- Computation of Net-Worth and Return on Investment (ROI) of the Investor.
- Giving alerts to the user if he requests for one.
- Downloading the current prices of shares from the web.
- User Authentication.
- Authorization.

1.3 Out of Scope:

- Features for actual purchasing and selling of securities. That is, buying and selling of shares/securities is done outside WM.
- Tax computations for gains/losses.
- Any market related prediction.

1.4 Definitions, Acronyms, and Abbreviations:

Abbreviations:Acronyms and
Abbreviations:

- WM: WealthMitra.
- SRS: Software Requirements Specification.
- WWW: World Wide Web.
- GUI: Graphical User Interface.
- ROI: Return on Investment.

2. Product Perspective

WM is aimed toward a person who has considerable number of investments in stock market, and so needs software assistance for bookkeeping and computations regarding the investments. WM should be user-friendly, 'quick tolearn' and reliable software for the above purpose.

WM is intended to be a stand-alone product and should not depend on the availability of other software. It should run on both UNIX and Windows basedplatform.

3. Product Functions

WM should support the following use cases:

Class of use cases	Use cases	Description of use cases
Use case related to Installation	Installation	Creates and initializes working files.
Use cases related to system authorize	Login	Loin into WM
	Change Password	Change WM Password
Use cases related to portfolios	Create portfolio	Creates a new port folio
	Rename portfolio	Rename an existing portfolio
	Delete portfolio	Delete an existing portfolio
Use cases related to securities	Create security	Creates a new security in a given portfolio
	Rename security	Renames an existing security
	Delete security	Deletes an existing security
Use cases related to transactions	Add transaction	Add a transaction to a security
	Edit transaction	Edit an existing transaction
	Delete transaction	Delete an existing transaction
Use cases related to Informationdisplay	Display investment	Display information of the entire investment

	Display portfolio	Display information about a given portfolio
	Display security	Display information about a given security
Use cases related to computations	Compute net-worth	Compute net-worth of Investment/portfolio/security
	Compute ROI	Compute ROI of a given security
Use cases related to shareprices	Get current share price	Download the current share price from the net
	Edit share price	Edit the price of a share already present in the list
Use cases related to alerts	Set alerts	Set alert giving date and Details
	Show alerts	Show all the pending alerts
	Delete alerts	Delete an already set alert

4.User Characteristics:

- The user should be familiar with the Investment Management relatedterminology like Portfolio/Security/Transaction.
- The user should know the details of a transaction.

4.1Principal Actors:

- Principal actors in WM are Users, Admin, CEO, Advisor

4.2General Constraints:

- For full working WM requires Internet connection.
- WM is single-user software.

4.3Assumptions and Dependencies:

- Full working of WM is dependent on the availability of Internet connection.
- The downloading of share prices in WM is customized to
- www.indiainfoline.com. WM would not work for any other website.
- The company shares registered in the National Stock Exchange (NSE) andBombay Stock Exchange are considered by WM.

5. Non-functional requirements

Below are the product's non-functional requirements:

- Performance: The performance of the system should be fast and as per user requirement. From this system we will get expected outcome in less time and less space since efficiency is higher. Speed is totally depending on the responseof the database and connection type.

- Reliability: Applicants can access their resume 98% of the time without failure.
- Availability: The system shall be available during 24 hours of a day.
- Maintainability: Under maintenance for approximately three hours if any required.
- Recoverability: If a major incident happens on the website and, the businessmust take measures to go back to being fully operational within three days.
- Data integrity: The system shall maintain data integrity by keeping backups ofall updates to the database for every record transaction.
- Usability: The system is designed keeping in mind the usability issues considering the end-users who are developers/programmers. It provides detailed help which would lead to better and faster learning. Navigation ofsystem is easy.
- Consistency: Uniformity in layout, screens. Menus, colour scheme, format.
- Safety Requirements: Only administrators have access to the database of each individual user All data will be backed-up every day automatically. This makes iteasier to install and updates new functionality if required for the safety purpose backup of the database must be required.
- Security Requirements:

  Validate all data received via the HTTP Request. Not validating data can result in attacks such as Cross-Site Scripting, SQL Injection, HTTP Response Splitting, Log Injection, and Directory Traversal.

Validate the data on the server-side. All data (even hidden fields and data from pull-down lists) are subject to being modified by a malicious user and should be validated server-side.

6.Products Navigation

Wealth management website will cover the following pages:

Home Page:

- Home
- About us
- Contact us
- Login / Register
- Services / Products
- Feedback

About us:

- About Company
- Team
- Top Reviews

Contact us:

- Common contact us page with locations of the company.

Login page:

- login form / Register button which redirects to Register page

Services / Products:

- Show Product Details such as Stocks, Equity Mutual Funds, Debt Mutual Funds, Liquid Mutual Funds, Liquid Cash.
- Advice as per Subscription Plans

Feedback:

- Feedback Forms
- Feedbacks

Performance Requirements:

- Should run on 500 MHz, 64 MB machine.
- 90% of the responses should be within 2 sec, except for downloading currentprices for which more time is acceptable.

Design Constraints:

- Security: The files in which the information regarding securities and portfoliosshould be secured against malicious deformations.
- Fault Tolerance: Data should not become corrupted in case of system crash orpower failure.

External Interface Requirements:

- The user screen is split vertically into two panes. The left pane contains the Investment tree, which expands and contracts as per user action. The right partdisplays the information related to investment/portfolio/security that is specified on the left pane.

7.Functional Requirements

7.1 Functional Requirements for Customer:

- View investments .
- Overall profit and loss
- Current market value of the investment made.
- Able to insert, update or delete his investment details.
- Change password, upgrade subscription plan and delete his Account

- View advice page ,stocks and a mutual funds page.

7.2 Functional Requirements for Admin:

- Add/Update/Delete Employee
- Add/Update/Delete new subscription plan
- Able to see advisors and customers details and perform CRUD operations.
- View performance of company.
- View performance of advisors.
- Make Changes in the respective Databases.
- View Profit and loss data of his customers.

7.3 Functional requirements

Advisor:

- View performance of Customers linked.
- View Performance of stocks advised by theadvisor.
- View total profit and loss of all the customers linked with advisor in suitable graphs and format.

7.4 Functional requirements of CEO:

- View Company Performance graphs and values.
- View Customers Information and details with their portfolioand profit (Highest to lowest order)
- View Advisors Information with their details and performance.

8.Use Cases:

8.1 Use cases related to system authorization:

1.Login:

Primary Actor: User

Pre-Condition: Nil

Main Scenario:

- Start the application. User prompted for login and password.
- User gives the login and password.
- System does authentication.
- Main screen is displayed.

Alternate Scenario:

- Authorization fails
- Prompt the user that he typed the wrong password
- Allow him to re-enter the password. Give him 3 chances.

2.Change Password:

Primary Actor: User

Pre-Condition: User logged in

Main Scenario:

- User initiates the password change command.
- User is prompted for old password, new password and confirm new password
- User gives the old password, new password and confirm new password.
- System does authentication.
- New password is registered with the system.

Alternate Scenario:

- Authorization fails
- Prompt the user that he typed the wrong password
- Allow him to re-enter the password. Give him 3 chances.
- New password and confirm new password do not match.
- Allow him to re-enter the attributes. Give 3 chances.

8.2 Use cases related to portfolios:

3.Create Portfolio:

Primary Actor: User

Pre-Condition: User Logged in

Main Scenario:

- User initiates the “create portfolio” functionality.
- System asks the user for the portfolio name.
- User enters the portfolio name.
- An empty portfolio is created.

Alternate Scenario:

- Portfolio with the same name exists.
- System asks the user for a different name.
- User enters a different name.
- Empty portfolio gets created.

4.Rename Portfolio:

Primary Actor: User

Pre-Condition: User Logged in

Main Scenario:

- User initiates the “rename portfolio” functionality.
- System asks for the portfolio to be renamed and the new name.
- User enters the new name.
- Portfolio is renamed.

Alternate Scenario:

- The portfolio whose name is supposed to change does not exist.
- Renaming fails, the error message is displayed.
- Portfolio with the same new name exists.
- Renaming fails, the error message is displayed.

5.Delete Portfolio:

Primary Actor: User

Pre-Condition: User Logged in

Main Scenario:

- User initiates the" delete portfolio" functionality.
- System asks for the name of the portfolio.
- The portfolio is deleted.

Alternate Scenario:

- Portfolio does not exist.
- Deletion fails, error message is displayed.

8.3Use cases related to information display:

6.Display Investment:

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

- User selects the option of viewing the Investment
- System computes the net worth for each portfolio.
- System displays the name of all portfolios with their net worth. It also shows the net-worth of entire investment by adding up the net-worth of each portfolio.

7.Display Portfolio:

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

- User selects the option of viewing a particular portfolio
- System computes the net-worth and ROI of each security.

- System displays the securities in the portfolio with the net-worth and ROI computed. It also displays the net-worth of the portfolio by summing up the net-worth of each security.

8.4 Use cases related to computations:

8.Compute Net-Worth.
Primary Actor: System.
Pre-Condition: User logged in & investment/portfolio/security Specified.

Main Scenario:

- System computes net-worth for investment/portfolio/security specified.

9.Compute Return on Investment.
Primary Actor: System.
Pre-Condition: User logged in & security specified

Main Scenario:

- System computes ROI for the security specified

8.5 Use cases related to share prices:

10.Getting the current prices of shares.
Primary Actor: User.
Pre-Condition: User logged in and internet connection exists.

Main Scenario:

- User initiates the “download current prices” functionality.
- The system downloads the current share prices from a particular website.

Alternate Scenario:

- Not able to download (due to network failure, site down, …)
- Display current value saved from before; ask the user to change it

11.Editing the share prices.

Primary Actor: User.

Pre-Condition: User logged in.

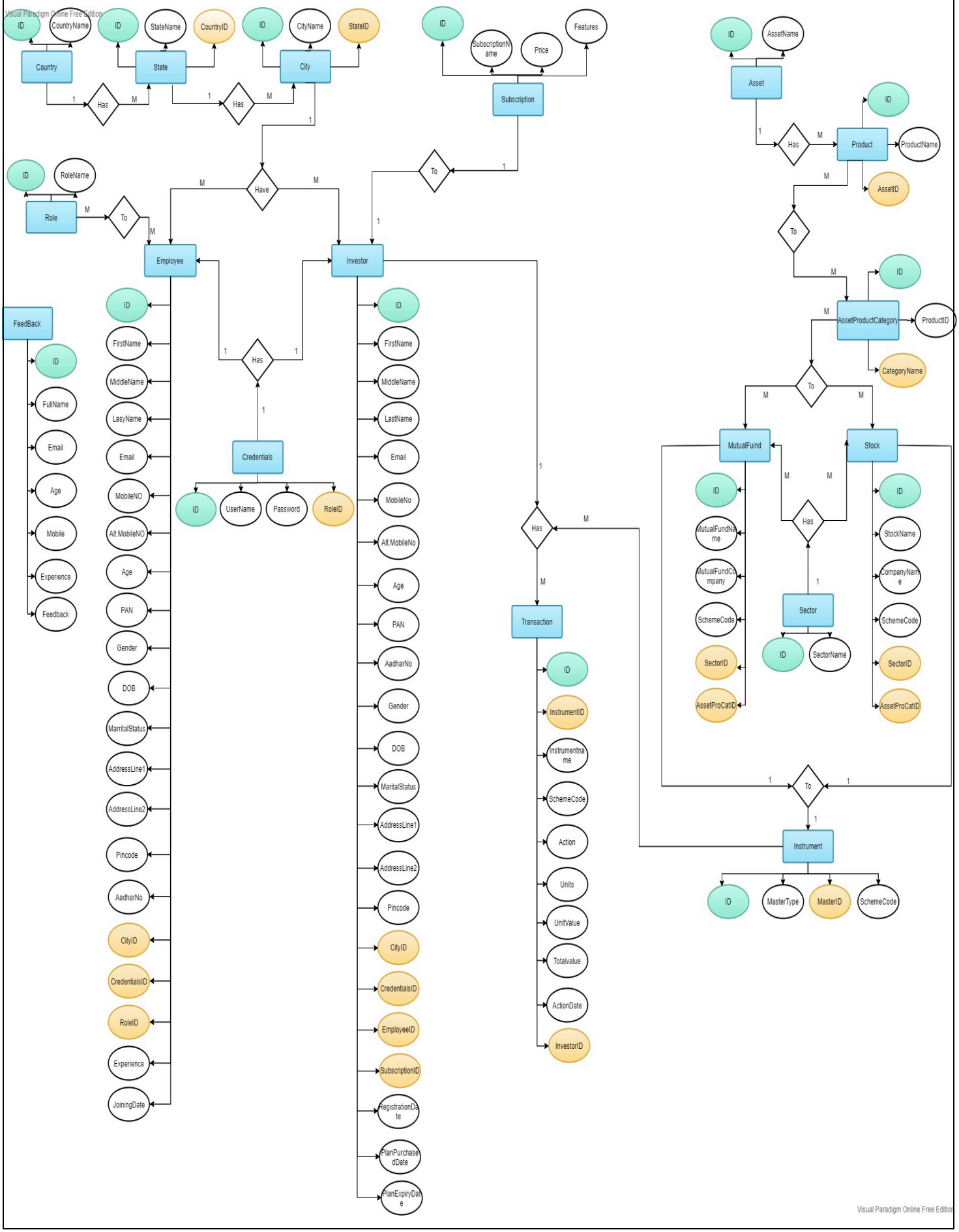
Main Scenario:

- User initiates the “Edit share prices” functionality.
- The system shows the list of all the companies and their current share prices (as known to the system).
- User edits the price of the company share he wants to edit.

Alternate Scenario:

- The entered price is invalid.
- User is shown the error displayed.
- The company name does not exist in the list of companies
- This is not handled by WM. WM is customized for the number of odd companies registered in the National stock exchange (NSE).

9.ER Diagram


Visual Paradigm Online Free Edition
ID
CountryName
Country
Has
StateName
State
CountryID
CityName
City
StateID
SubscriptionName
Price
Features
Subscription
To
Have
RoleName
Role
Employee
Investor
FeedBack
FullName
Email
Age
Mobile
Experience
Feedback
FirstName
MiddleName
LasyName
MobileNO
Alt.MobileNO
PAN
Gender
DOB
MarritalStatus
AddressLine1
AddressLine2
Pincode
AadharNo
CityID
CredentialsID
RoleID
JoiningDate
Credentials
UserName
Password
LastName
MobileNo
Alt.MobileNo
AadharNo
SubscriptionID
RegistrationDate
PlanPurchasedDate
PlanExpiryDate
Transaction
InstrumentID
Instrumentname
SchemeCode
Action
Units
UnitValue
TotalValue
ActionDate
InvestorID
Asset
AssetName
Product
ProductName
AssetID
AssetProductCategory
ProductID
CategoryName
MutualFuind
Stock
MutualFundName
MutualFundCompany
SectorID
AssetProCatID
StockName
CompanyName
Sector
SectorName
Instrument
MasterType
MasterID
Visual Paradigm Online Free Edition

10.Database Tables

Table: - Roles

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)		Not null	Unique
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Sector

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)		Not null	Unique
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Assets

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)		Not null	Unique
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Subscription Plans

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)		Not null	Unique
Price	Float		Not null	
Features	Varchar (100)		Not null	
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Country

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)		Not null	Unique
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Credentials

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		
UserName	Varchar (100)		Not null	
Password	Varchar (100)		Not null	
RoleId	Int	Foreign		
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - State

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar (100)			Unique
CountryId	Int	Foreign		
IsActive	Bit			Default 1
RecordCreatedBy	Varchar (100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar (100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - City

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar(100)			Unique
CountryId	Int	Foreign		
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Employees

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		
FName	Varchar(100)		Not null	
MName	Varchar(100)		Not null	
LName	Varchar(100)		Not null	
Email	Varchar(100)		Not null	
MobileNo	Varchar(100)		Not null	
Alternate MobileNo	Longint		Not null	
Age	Longint		Not null	
PAN	Varchar(100)		Not null	
Gender	Varchar(100)		Not null	
DOB	Datetime		Not null	
Marital Status	Varchar(100)		Not null	
AddressLine1	Varchar(100)		Null	
AddressLine1	Varchar(100)		Null	
PinCode	Int		Null	
AdharNo	Varchar(100)		Null	
CityId	Int	Foreign		
CredentialId	Int	Foreign		

RoleId	Int	Foreign		
Experience	Varchar(100)			
JoiningDate	Datetime			
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Products

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar(100)		Not null	Unique
AssetId	Int	Foreign		

Table: - AssetProductCategory

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
ProductId	Int	Foreign		
CategoryName	Varchar(100)		Not null	Unique

Table: - Investors

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		
FName	Varchar(100)		Not null	
MName	Varchar(100)		Not null	
LName	Varchar(100)		Not null	
Email	Varchar(100)		Not null	
MobileNo	Varchar(100)		Not null	

Alternate MobileNo	Longint		Not null	
Age	Longint		Not null	
PAN	Varchar(100)		Not null	
Gender	Varchar(100)		Not null	
DOB	Datetime		Not null	
Marital Status	Varchar(100)		Not null	
AddressLine1	Varchar(100)		Null	
AddressLine1	Varchar(100)		Null	
PinCode	Int		Null	
AdharNo	Varchar(100)		Null	
CityId	Int	Foreign		
CredentialId	Int	Foreign		
EmployeeId	Int	Foreign		
SubscriptionPlanId	Int	Foreign		
CurrentStatus	Varchar(100)			
RegistrationDate	Datetime		Not null	
PlanPurchasedDate	Datetime			
PlanExpiryDate	Datetime			
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Stocks

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar(100)		Not null	Unique
SchemeCode	Varchar(100)			Unique

AssetProductCategoryId	Int	Foreign		
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - MutualFunds

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		Identity
Name	Varchar(100)		Not null	Unique
SchemeCode	Varchar(100)			Unique
AssetProductCategoryId	Int	Foreign		
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Instruments

Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary	Not null	
SchemeCode	Varchar(100)			Unique
MasterType	Varchar(100)			
MasterTableId	Int	Foreign		

Table: - Transaction

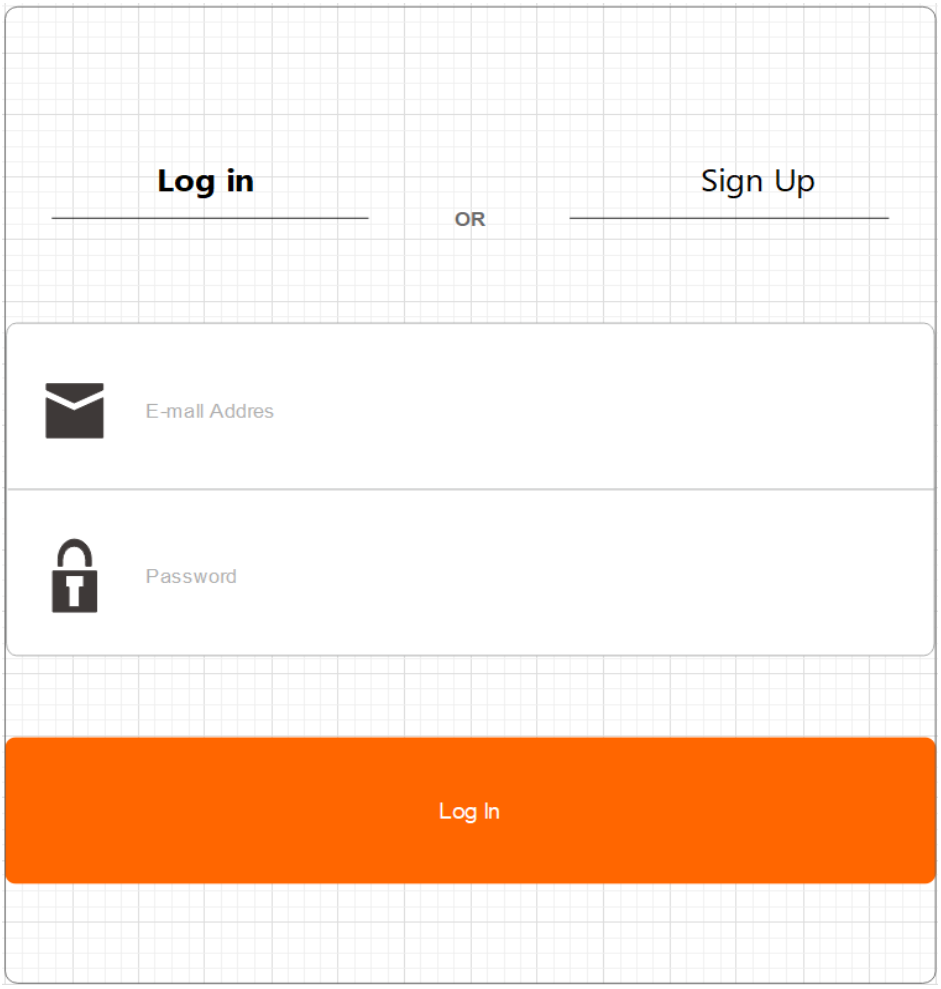
Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		
InvestorId	Int	Foreign		
InstrumentId	Int	Foreign		
NameOfInstrument	Varchar(100)		Not null	
SchemeCode	Varchar(100)		Not null	Unique
Action	Varchar(100)			
Unit	Int			
UnitValue	Int			
TotalValue	Int			
PurchasedDate	Datetime			
IsActive	Bit			Default 1
RecordCreatedBy	Varchar(100)		Not null	
RecordCreatedDate	Datetime		Not null	
RecordModifiedBy	Varchar(100)		Null	
RecordModifiedDate	Datetime		Null	

Table: - Feedback

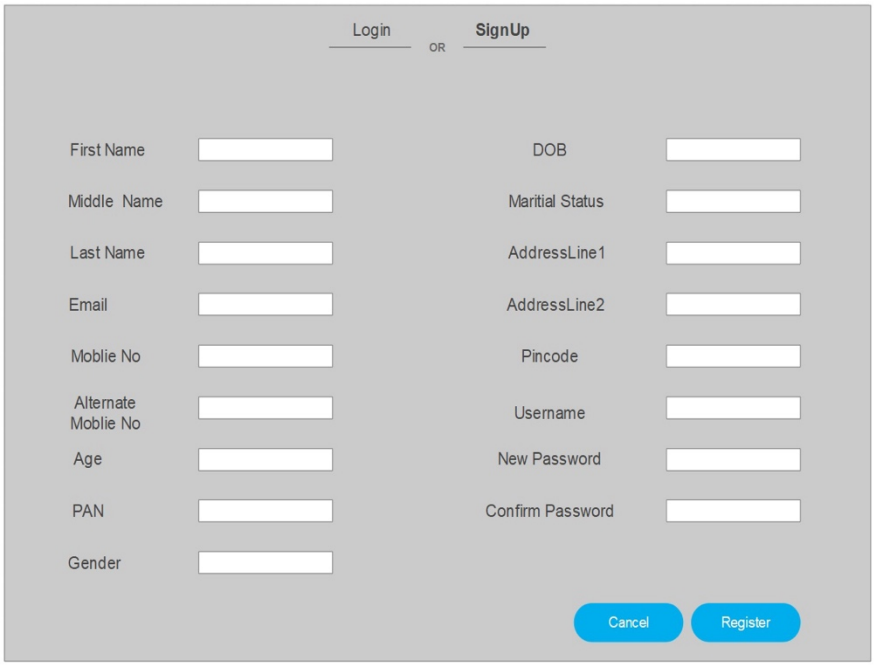
Fields	Datatype	Keys	Null/Not null	Identity
Id	Int	Primary		
FullName	Varchar(100)		Not null	
Email	Varchar(100)		Not null	
MobileNo	Longint			
Age	Int			
Message	Varchar(100)		Not null	
Experience	Varchar(100)		Not null	

11.Wireframes:

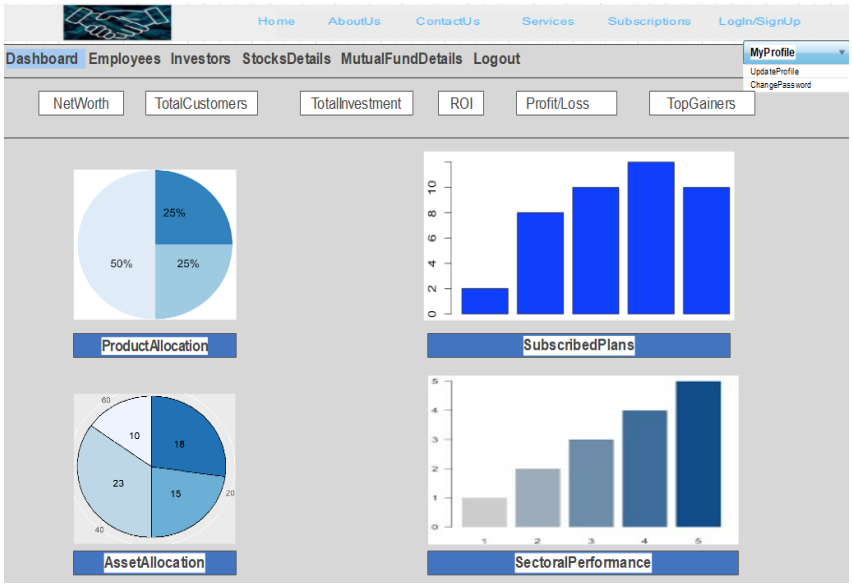
Login: -



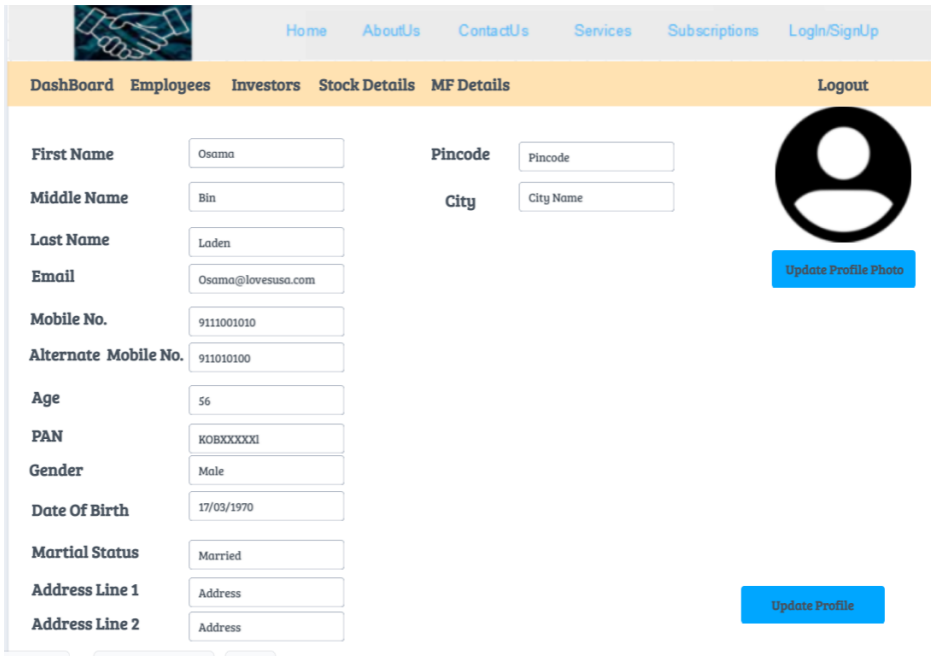
Sign Up: -



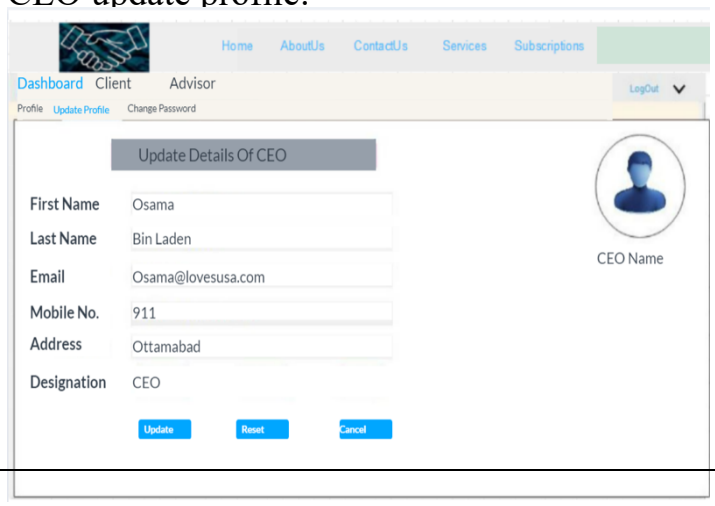
CEO Dashboard: -



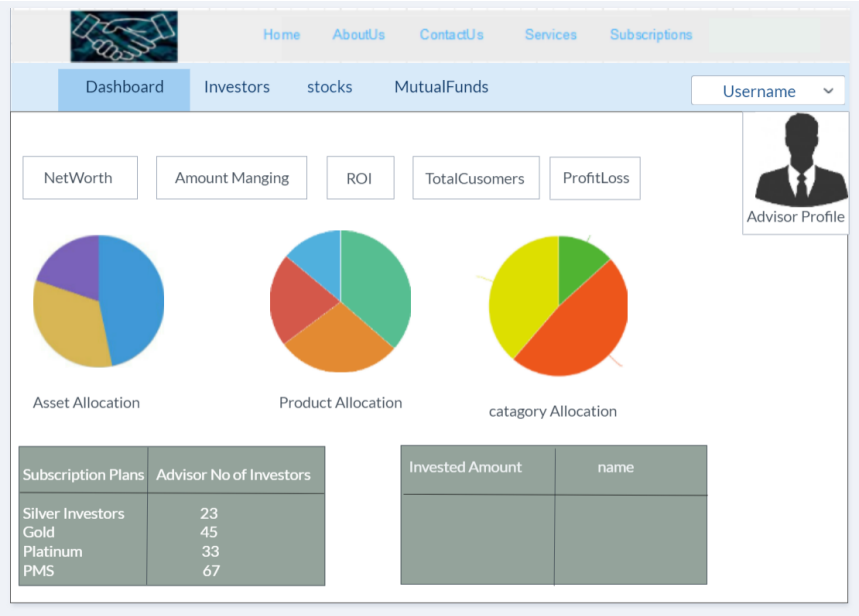
CEO Profile: -



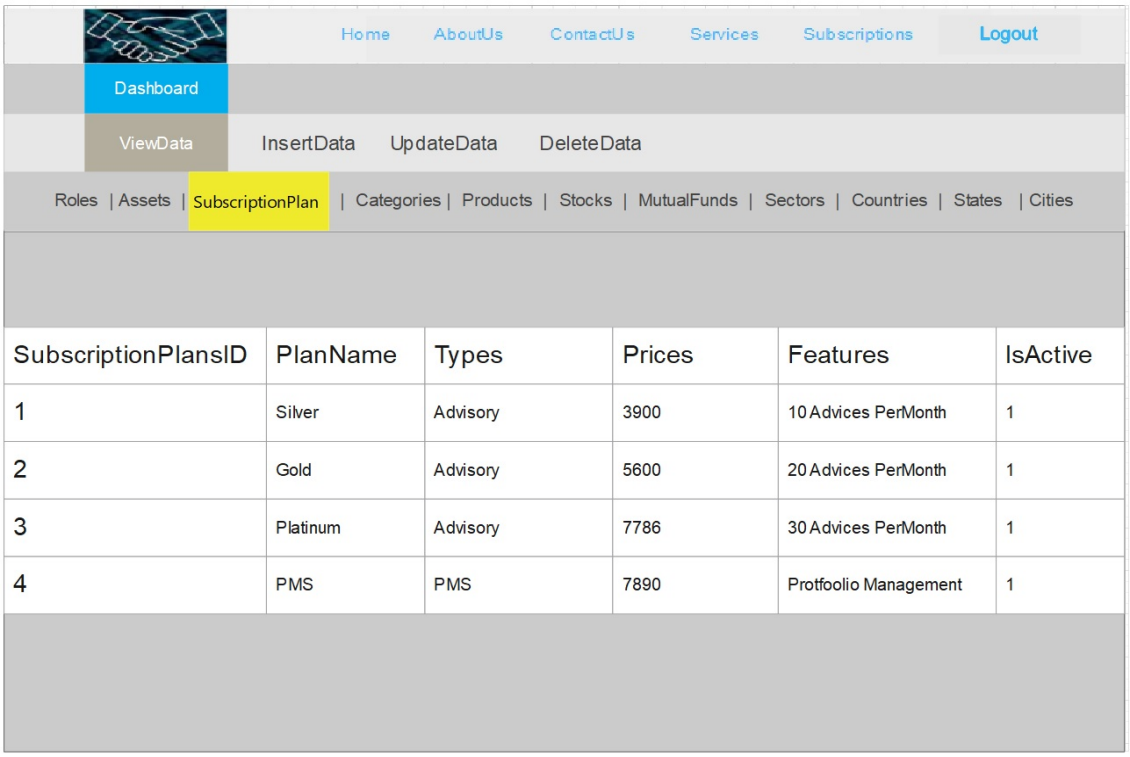
CEO update profile: -



Advisor Dashboard: -



Admin Dashboard: -



SubscriptionPlansID	PlanName	Types	Prices	Features	IsActive
1	Silver	Advisory	3900	10 Advices PerMonth	1
2	Gold	Advisory	5600	20 Advices PerMonth	1
3	Platinum	Advisory	7786	30 Advices PerMonth	1
4	PMS	PMS	7890	Protfoolio Management	1

Investor Dashboard: -