Project Phase 1

Team no - 10

Team Members: Prisha, Vanshika Dhingra, Aaryan Ajay Sharma

Introduction to Miniworld

The wave of finance in the youth of our country is rising rapidly as we witnessed a pandemic and a year-long lockdown. There are a number of websites which act as a good portfolio manager or a stock screener in the Indian finance arena, but there is yet to be an application/website which would provide the convenience of being both, a screener AND a portfolio manager.

Purpose Of Database

The purpose of our project ***Stock-base*** is to integrate the features of a typical portfolio management system and an Indian stock screener, like Google Finance and Screener for example. It would be a one-stop solution for keeping the records and updates of portfolios in check, while also acting as a good stock screener, saving people's time from fruitlessly juggling between different websites for researching a stock.

Users of the database

Users of the database will consist of investors and traders looking to find a convenient haven where they can easily & safely keep records, updates, and accounts of their portfolios, and have the convenience of a stock screener in the palm of their hands for personal research. All these users will have accounts named "User Account".

There will also be administrator accounts that can view the entirety of the database, using accounts called "Admin Account".

Applications of the database

- Users can view the current price of the stocks, and their percentage change.
- Users can screen and analyse stocks based on their preferred parameters.
- Users can maintain different portfolios which contain the stocks they have invested in.
- Users can also maintain a watchlist for the stocks that they want to constantly monitor.
- Users have the convenience of having a portfolio manager and screener in one place.

Database requirements

Assumptions/User Description

- An account has information such as -
    - Account ID
    - Username
    - Email
    - Password
- There are two kinds of accounts - User and Admin
    - User account has more information such as DOB, Age & Name
        - The user may not have the last name but should have a first name
- A user after using the app can give one or more than one feedback which is viewable to Admin
- Feedback has a title of not more than 60 characters and a text message which is at least 30 characters and at most 500 characters
- The user account will consist of watchlists and portfolio lists
    - The watchlist can have stocks enlisted with information such as -
        - Security Code like DMART for Avenue Supermarts LTD. or 532540 for Tata Consultancy Service LTD.
        - Name of the stock
        - Current Price of the stock
        - Price change, which is the difference between the Current Price and the Previous Closing price of the stock
        - Percentage change which is 100 × (Price change) /(Previous Closing)
        - Prices of stock are reported to 2 decimal places and in rupees
    - Portfolio lists will consist of the stocks owned by the user with information such as
        - Security Code like DMART for Avenue Supermarkets LTD. or 532540 for Tata Consultancy Service LTD.
        - Name of the stock
        - Current Price of the stock
        - Total quantity of the stock
        - Day change, which is total quantity times the difference between Current Price and the Previous Closing price of the stock
        - Percentage change which is 100 × (Price change) /(Previous Closing)
        - Market Value, which is Quantity times the current price of the stock
    - Both portfolio list and watchlist can have indices listed which will show
        - INDEX tag demarcating that the specific instrument is an index.
        - Name of the index
        - Points of the index
        - Point change, which is the difference between the current point and the previous close of the index
        - Percentage point change, which is 100 × (Point change) /(Previous Closing)
        - Points of the indices are reported to 2 decimal places
- The screen which will be displayed upon searching for a stock will consist of the following information about the stock
    - All the information that is displayed about a stock in portfolio list
    - Market cap reported to 2 decimal points in crores (Crs.)

- Yearly high and low of the stock (Both reported up to 2 decimal places)
- Stock P/E (reported up to 1 decimal places)
- Book Value (reported up to 2 decimal places)
- Dividend Yield (reported to 2 decimal places)
- ROCE (reported to 2 decimal places)
- ROE (reported to 2 decimal places)
- Face Value (reported up to 2 decimal places)

Strong Entity Types

Legends -

Underlined attribute → Primary Key
Italicised attributes → Derived
Boldened attributes → Composite

- **Account**
  - username: string → Not Null
  - email: string → Not Null
  - accountID: Int → 6 digit Int, Primary Key
  - password: string → Not Null

- **Transaction**
  - Purchase Price: float, Not Null
  - **Date of Purchase: string → Composite, Not Null**
    - Date of purchase
    - Month of purchase
    - Year of purchase
  - Current Price: float, Not Null
  - TransactionID: 6 Digit Int, Primary Key
  - Quantity: Int → NOT NULL
  - *marketValue : float→ NOT NULL, Derived*
    - Derived from current price and quantity
  - *TotalChange: float→ NOT NULL, Derived*
    - Derived from current price, purchase price and quantity
  - comment: string

- **Stock**
  - prevClose: float → Not Null
  - currentPrice: float → Not Null
  - dayHigh: float → Not Null
  - dayLow: float → Not Null
  - stockType: string → enum {BSE, NSE}
  - totalQuantity: int → Not Null
  - Security Code: string → Not Null
  - marketCap: float → Not Null
  - High: float → Not Null

    - Low: float → Not Null
    - Stock P/E: float → Not Null
    - BookValue: float → Not Null
    - dividendYield: float → Not Null
    - ROCE: float → Not Null
    - ROE: float → Not Null
    - faceValue: float → Not Null
    - *dayChange: float→ NOT NULL, Derived*
        - Derived from current price, purchase price and totalQuantity
    - *priceChange: float→ NOT NULL, Derived*
        - Derived from current price and prevClose

- **User**
    - Inherits all the attributes from accounts entity type

    - **Name: string → Composite**
        - First Name → Not Null
        - Middle Name
        - Last Name
    - *Age: Int → Not Null, Derived*
        - *Derived from Date of Birth*
    - **Date of Birth: string → Composite, Not Null**
        - Date of birth
        - Month of birth
        - Year of birth
    - **Phone Number: string → Not Null, Multivalued, Composite**
    - **Address: string → Composite**
        - Street name → Not Null
        - Address line 1 → Not Null
        - Address line 2
        - Pincode → Not Null

- **Admin**

    - Inherits all the attributes from accounts entity type

Weak Entity Types

- **Feedback**

    - Title: VARCHAR(60) → Not Null, Partial Key
    - Message: VARCHAR(500) → Not Null
    - Its identifying relationship is ***gives***
    - Its identifying entity is ***user***

- **watchList**
  - serialNo : int → Not Null, Partial Key
  - securityCode : string → Not Null
  - instrumentName : string → Not Null
  - currentPrice : float → Not Null
  - priceChange : float → Not Null
  - percentageChange : float → Not Null
  - indexTag (index or not) : boolean → int enum{0,1}
  - Its identifying relationship is ***has***
  - Its identifying entity is ***user***

- **portfolioList**
  - serialNo : int → Not Null, Partial Key
  - securityCode : string → Not Null
  - instrumentName : string → Not Null
  - currentPrice : float → Not Null
  - TotalQuantity : int → Not Null
  - dayChange : float → Not Null
  - percentageChange : float → Not Null
  - marketValue : float → Not Null
  - indexTag (index or not) : boolean → int enum {0,1}
  - Its identifying relationship is ***has***
  - Its identifying entity is ***user***

Relationship Types

- ***supervision***
  - Degree: 1 (unary)
  - recursive relationship
  - Entity Types: Supervisor **user** → Supervisee **user**
  - Min-max ratio
    - Supervisor user → (0, N): A user can have 0 supervisee at least, and N supervisee at most
    - Supervisee user → (0, 1): A user can be supervised by 0 supervisors at least and 1 supervisors at most
- ***gives***
  - Degree: 2 (Binary)
  - Entity Types: **User** → **Feedback**
  - Min-max ratio
    - User → (0, N): A user can give at least zero feedback, or may give more than one feedbacks
    - Feedback → (1, 1): A feedback can be given by at least one user and at most by one user
- ***contains***
  - Degree: 2 (Binary)
  - Entity Types: **portfolioList** → **Stock**

    - Min-max ratio
        - portfolioList → (0,N): A portfolio can have at least zero stocks in it and may have N stocks at most
        - Stock → (0, N): A stock may belong to no portfolio, and may belong to N portfolios at most
- ***Comprises of***
    - Degree: 2 (Binary)
    - Entity Types: **Stock → transaction**
    - Min-max ratio
        - stock → (0,N): A stock can have zero transactions at least, or N transactions at most
        - transaction → (1, 1): A transaction can belong to 1 stock at least and 1 at most
- ***performs***
    - Degree: 2 (Binary)
    - Entity Types: **user → transaction**
    - Min-max ratio
        - user → (0,N): A user can perform zero transactions at least, or N transactions at most
        - transaction → (1, 1): A transaction can be performed by one user at least, and one user at most
- ***has***
    - Degree: 3 (Binary)
    - Entity Types: **user ,portfolioList, watchList**
    - Min-max ratio
        - user → (0,N): A user can have at least zero portfolio list, and may at most have N portfolio lists
        - portfolioList → (1, 1): A portfolio list can belong to at least one user and at most one user
        - watchList → (1, 1): A watchlist can belong to at least one user and at most one user

- ***views***
    - Degree: 4 (Quaternary)
    - Entity Types: **Admin, Feedback, portfolioList, Transaction**
    - Min-max ratio
        - Admin → (0, N): Admin can view nothing, or can view N feedbacks, N portfolios and N transactions
        - Feedback → (0, N): A feedback can be viewed by 0 admin at least, or N admins at most
        - portfolioList→ (0, N): A portfolio can be viewed by 0 admin at least, or N admins at most
        - Transaction→ (0, N): A transaction can be viewed by 0 admin at least, or N admins at most

Subclass

- Admin and user are the subclasses of Account
- They inherit all the attributes of account

Functional Requirements

- **Modification**
    1. **INSERT:**
        - a feedback by a user
        - details of a user
    2. **UPDATE:**
        - current price of stocks
        - percentage gain or loss of a stock

- **Retrievals**
    1. **SELECTION:**
        - Retrieve the stock(s) whose stockType is BSE.
        - Retrieve the portfolioList(s) whose index tag is set to 1.
    2. **PROJECTION:**
        - Retrieve the marketValue of all the transactions whose purchase month is september
        - Retrieve the stocks whose price change is positive
    3. **SEARCH**:
        - List all the users whose name starts with ‘A’
    4. **AGGREGATE:**
        - Get the transaction with maximum purchase price
        - Get the total money invested in a particular stock
    5. **ANALYSIS:**
        - Analysing which stock is the better to invest in.
        - Analysing a user’s investment interests.

Summary

We are able to provide the benefits of both a portfolio manager and a scanner to the user and hence making their work easy and convenient.