Detailed ER Diagram Description

Overview

In the Stock Market Portfolio application, the design uses a single collection, Stock, for both storing stock data and managing user watchlists. This approach simplifies the database schema by avoiding the need for a separate collection for the watchlist. Here's a detailed explanation:

ER Diagram:

1. Collection Overview

Collection Name: Stock

Purpose:

- **Data Retrieval**: This collection stores all stock-related information which is retrieved and displayed to users.
- **Watchlist Management**: It also manages user watchlists by storing stocks that users want to keep track of. There is no separate Watchlist collection; instead, the watchlist functionality is integrated into the Stock collection.

2. Data Fields

- **company**: Represents the name of the company associated with the stock (e.g., "Apple Inc.").
- **description**: Provides a brief description of the company or the stock (e.g., "Technology company specializing in consumer electronics").
- **initial_price**: Records the initial price of the stock when it was first listed (e.g., $50).
- **price_2002**: Contains the price of the stock in the year 2002 (e.g., $25).
- **price_2007**: Contains the price of the stock in the year 2007 (e.g., $40).
- **symbol**: A unique identifier or ticker symbol for the stock (e.g., "AAPL" for Apple Inc.).

3. Functionality

Retrieval of Stock Data:

- **GET Requests**: The application retrieves stock data from the Stock collection using GET requests to display stock information on the frontend.

Adding to Watchlist:

- **POST Requests**: Users can add stocks to their watchlist by making POST requests to the backend. The same Stock collection is used to handle these requests.
    - **Adding New Stocks**: When a new stock is added to the watchlist, it is inserted as a new document into the Stock collection.
    - **Updating Existing Stocks**: If the stock already exists, it can be updated with new information or marked as part of the user's watchlist.

4. Design Rationale

Unified Collection:

- **Simplicity**: Using a single collection for both stock data and watchlist management reduces complexity in the schema and simplifies data access and manipulation.
- **Efficiency**: This design eliminates the need for joins or multiple queries to manage and retrieve watchlist items, improving performance and maintainability.

5. Example Document

Here's an example of how a stock document might look in the Stock collection:

json


```
{
  "_id": "ObjectId('...')",
  "company": "Apple Inc.",
  "description": "Technology company specializing in consumer electronics.",
  "initial_price": 50,
  "price_2002": 25,
  "price_2007": 40,
  "symbol": "AAPL"
}
```


6. Benefits

- **Ease of Use**: Users can view and manage their watchlist using the same interface as for viewing all stocks.

- **Streamlined Operations**: Backend operations are streamlined by handling all stock-related requests within a single collection.

Conclusion

The design choice of using a single Stock collection for both stock data retrieval and watchlist management is efficient and straightforward, making the application easier to develop and maintain.