**Stock Portfolio Management Application: Documentation**

**BY:**

**Team CompileCrew**

**Overview**

This program is focused on stock trading and it was written in Java. The people, who own stocks or want to invest in stocks, will be able to register or log in to the application to manage their stocks. The solution has graphical user interface and makes usage of SQLite as its data model for the application. Other recent changes include:

1. A welcome page appearing with both login and register applications.

2. Inclusion of stock names such as Google, Microsoft and Amazon to the system.

------------------------------------------------------------------------------------------------------------------------------

**How It Works**

**1. Welcome Page**

- Purpose: This page is characterized as the beginning of every user’s experience on the system

- Components:

- A login button on the screen that takes the user to the log-in page.

- A register button on the page that takes the user to the creating account page.

- Enhancement: This feature helps the users to navigate around the application with more ease than before.

**2. User Authentication**

- Login:

- The system asks the user to input their username and a corresponding password.

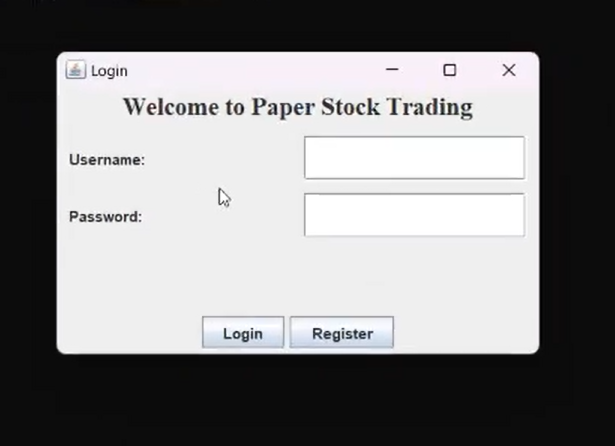
- The inputted data is checked for existence within the SQLite database through an interface, the `UserRepository` class.

- After these processes the user is logged in and the next screen is the main portfolio management interface.

- Register: Users can now create a new account to a site by inserting their username, password and other information.

- The new user is inserted by the `UserRepository` class into sqlite database.

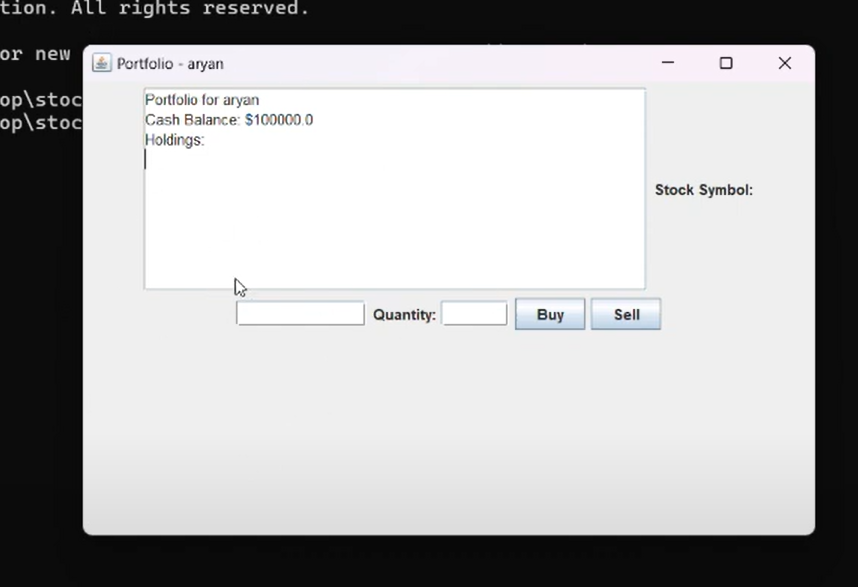
New credentials are sent to users after they register that they can use to log in.



**3. Portfolio Management:**

As per the login details users are able to:

* Add Stocks: To Buy Stock, write down the name of the stock and the amount required.
* Remove Stocks: To delete stocks that are in their portfolio.
* View Portfolio: provides an overview of all the shares they have including: stock name, quantity owned and total worth of their stock.

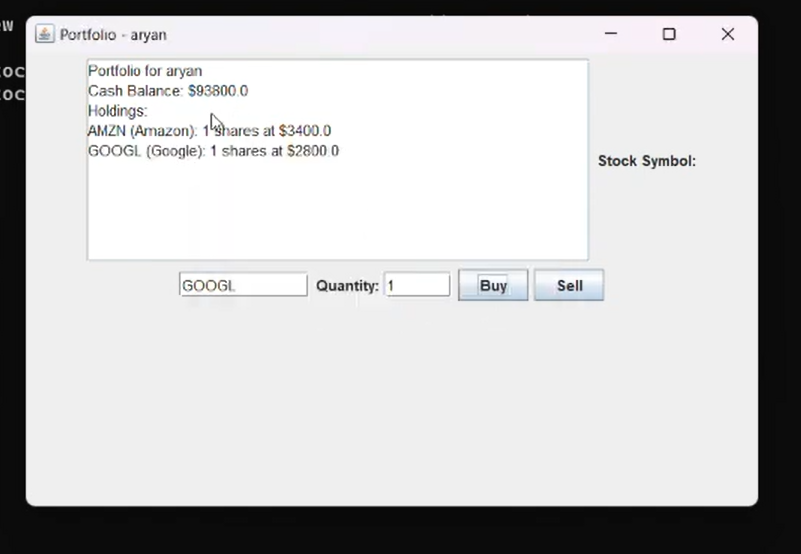


**4. Database Management:**

For long term storage of users and their portfolio details SQLite is used.

The essential three tables of a database include:

* Users: Contains data concerning the users such as (username, user password).
* Stocks: Contains information regarding stocks together with their names (for example, GOOGL, MSFT, AMZN and etc).
* Portfolio: User-to-stock ownership links and number of stocks per user.



5. Supported Stocks:

The following pre-coded stocks are only used for:

* GOOGL of Google Inc
* MSFT of Microsoft Inc
* AMZN for Amazon Inc

These stocks can be purchased by the users and added to their portfolio.

-------------------------------------------------------------------------------------------------

**Code Breakdown**

**1. MainApp.java:**

Purpose: The first window of the application.

Functionality:

Open the main interface of the application and display the welcome screen.

Perform all other initialization activities required (database connections, setup of other relevant connections).

**2. LoginFrame.java:**

Purpose: It has provisions for log-in as well as registration of users.

**3. Key Classes/Listeners:**

LoginButtonListener: Once the user has entered their credentials and logged in successfully, on press of this button the user is taken to the portfolio screen soon after where their stocks are displayed.

RegisterButtonListener: Takes new user credentials from the user and stores them in his/her account in the database.

**4. PortfolioFrame.java.:**

Purpose: A Core interface which deals with the management of portfolio.

Key Components:

Buy Button:

This button opens a dialog for the user to provide stock particulars like stock name and stock quantity. Uses the PortfolioRepository class to add the stock to the portfolio in the database.

Sell Button:

Allows the user to select which stock and how much to sell.

Selling of stock also comes with an update of the portfolio database.

**5. Portfolio Display:**

This function includes a table or a list that presents all the stocks owned by the user which clearly shows the quantity and value of those stocks.

**6. Database Management Classes:**

UserRepository.java:

This section is concerned with the three major types of users and their operations namely creation, reading, updating and deleting data about the user.

Does the login process of the users and signup by communicating with the Users table.

StockRepository.java:

Stock information which includes company symbols, stock name, price and other particulars are housed in the Stocks table.

This also includes stock name retrieval and details about the stock name.

PortfolioRepository.java:

Considers the stocks which belong to a particular user in the Portfolio table along with the number of such stocks.

Modifies the portfolio at the time of buying and selling of stocks.

**5. Database (`finance.db`)**

-  **Tables** :

  -  **Users** :

    - Columns: `id`, `username`, `password`

    - Example:

      | id  | username | password  |

      |-----|----------|-----------|

      | 1   | alice    | 1234abcd  |

  -  **Stocks** :

    - Columns: `id`, `stock\_name`

    - Example:

      | id  | stock\_name |

      |-----|------------|

      | 1   | GOOGL      |

      | 2   | MSFT       |

      | 3   | AMZN       |

  -  **Portfolio** :

    - Columns: `user\_id`, `stock\_id`, `quantity`

    - Example:

      | user\_id | stock\_id | quantity |

      |---------|----------|----------|

      | 1       | 1        | 10       |

**7. Upgrades To the Graphical User Interface:**

To better understand and interact with the application, a welcome page has been introduced many enhancements have been made.

Users can now better secure their accounts by having login and register buttons.

* Functions
* Authentication
* It is a secure login and registration process.

Portfolio Management:

It is a simple interface for graphics implementing buying, selling, and watching stocks.

Predefined Stocks:

Predefined stocks in the application are Google, Microsoft, and Amazon.

Persistence:

All user and portfolio information is stored in an SQLite Database for a durable solution.

--------------------------------------------------------------------------------------------------------------------

**Execution Steps**

1. **Compilation**:
   * Navigate to the directory containing the Java files.
   * Compile all .java files:

bash

Copy code

javac -cp sqlite-jdbc-3.47.1.0.jar .java

1. **Run the Application**:
   * Start the application with:

bash

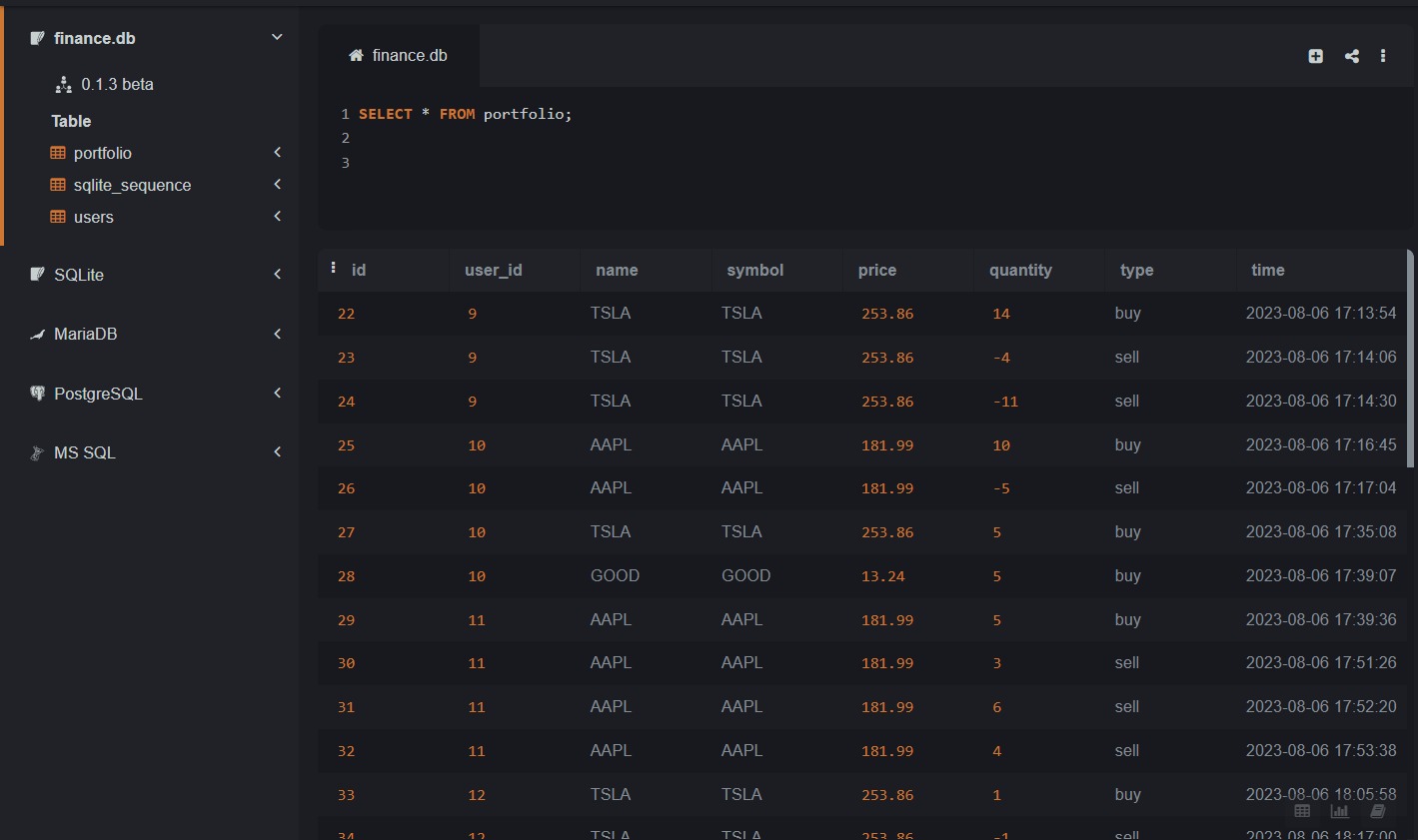
Copy code

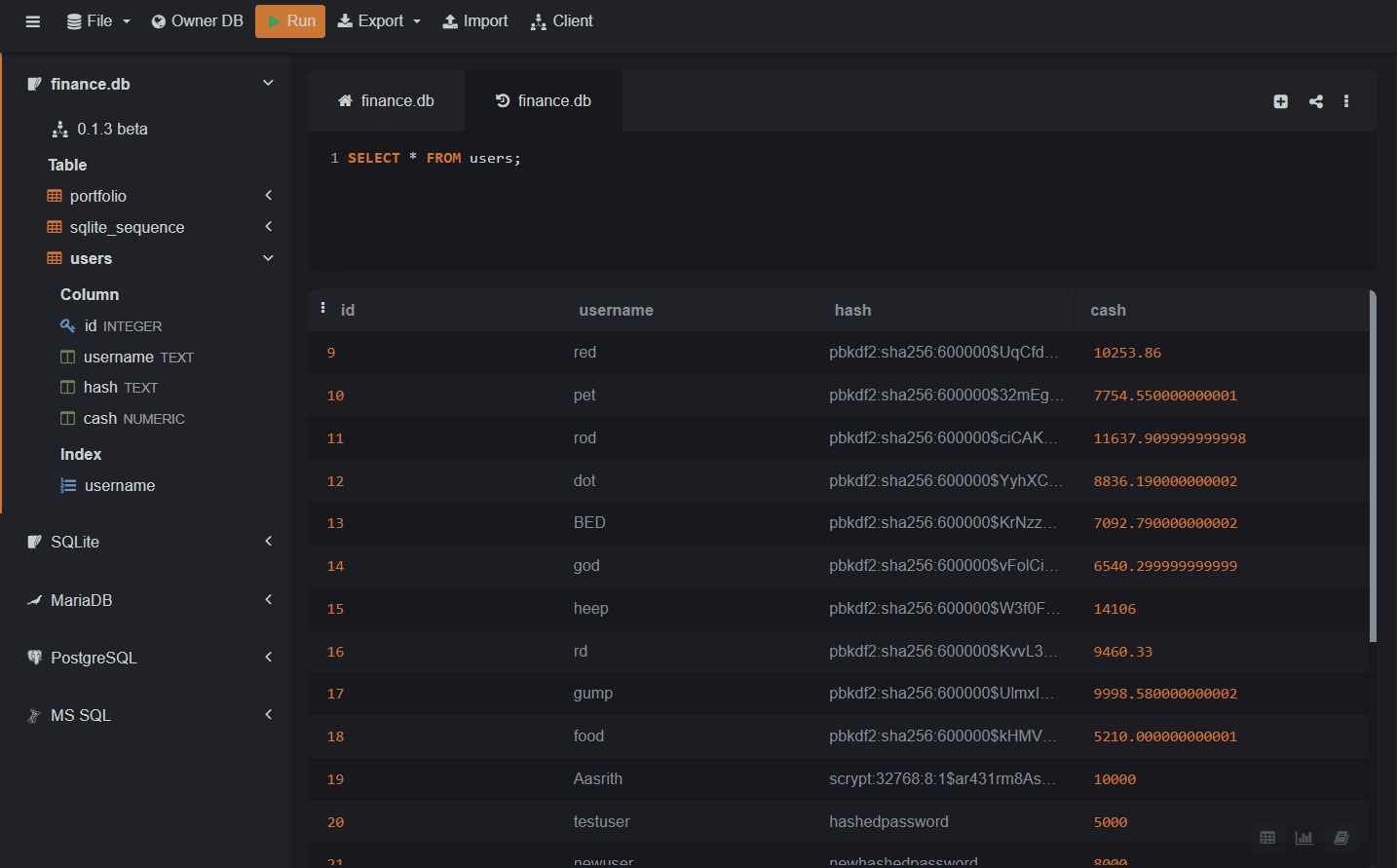
java -cp .:sqlite-jdbc-3.47.1.0.jar MainApp

1. **Navigate the Application**:
   * Welcome page: Choose to **Login** or **Register**.
   * After login: Manage your stock portfolio (add, remove, or view stocks).

Forthcoming Improvements:

* Incorporate real-time stock price updates via API.
* Add various other stocks aside from the provided ones.
* Most importantly, include functionality like performance tracking and advanced analytics.





**Contributors:**

**Aryan Karthik. P (SE22UARI022):**

1. GUI
2. MainApp
3. Representation of UML diagram

**Aasrith Reddy (SE22UARI092):**

1. Login Frame
2. Portfolio Class

**Sai Shivan Raj (SE22UARI106):**

1. User Repository Class
2. Transaction Class
3. Transaction Repository Class

**Nandan Reddy (SE22UARI001):**

1. Stock
2. Stock Repository Class

**Sandeep (SE22UARI151):**

1.User Class