GollapudiJPhanMGuyonG

Gregory Guyon | Jai Gollapudi | Minh Phan |

Personal Cryptocurrency Portfolio Management System

README

Project Description

This system enables you to manage your cryptocurrency portfolio efficiently. It provides various features and functionalities, including:

- User Registration and Login: Register a new user or login to an existing account.
- Setting or Viewing Preferences: Set or view current preferences for an account (Preferred Currency, Notification Frequency)
- Creating or Viewing Portfolio: Create a new portfolio or view an existing one for an account.
- Setting or Viewing Alerts: Set alerts for specific market conditions or view existing alerts (Crypto Name, Price Increase or Decrease, Price Threshold).
- Viewing Wallet: View the wallet associated with the account (Wallet Address, Wallert Type)
- Inputting Transactions: Input new buy or sell transactions into the system.
- Viewing Transactions: View all the transactions associated with a portfolio in an account.
- View Profit/Loss: View the total profit or loss based on the transactions in a portfolio.
- Visualize Portfolio Distribution: Generate a pie chart of crypto distribution in a portfolio.
- Delete Profile: Delete an existing user profile along with all its data.

Software Requirements

- Python (https://www.python.org/downloads/)
- MySQL Database Management System (https://dev.mysql.com/downloads/)

Library Dependencies

The project requires the following Python libraries:

- mysql-connector-python
- pandas
- Datetime (datetime is a built-in Python library, so it doesn't need to be installed separately)
- matplotlib

Installation Directories

The main directory is called CS3200_GollapudiJPhanMGuyonG, and it contains the following files:

- main.py: The main script to run the application.
- functions.py: Contains all the function definitions to execute features of the system
- requirements.txt: Lists all the Python library dependencies.
- README.txt: The README requirements as listed in this document.

Installation Steps

To install and run this project:

- Ensure you have Python and MySQL installed on your machine.
- Install the Python libraries by running pip install -r requirements.txt

in your command line. (In the directory containing requirements.txt)
- Clone the project into your desired directory.
- Import the SQL dump file into your MySQL database to setup the initial database schema and populate it with necessary data. You can do this via a MySQL GUI tool like MySQL Workbench.

Running the Application

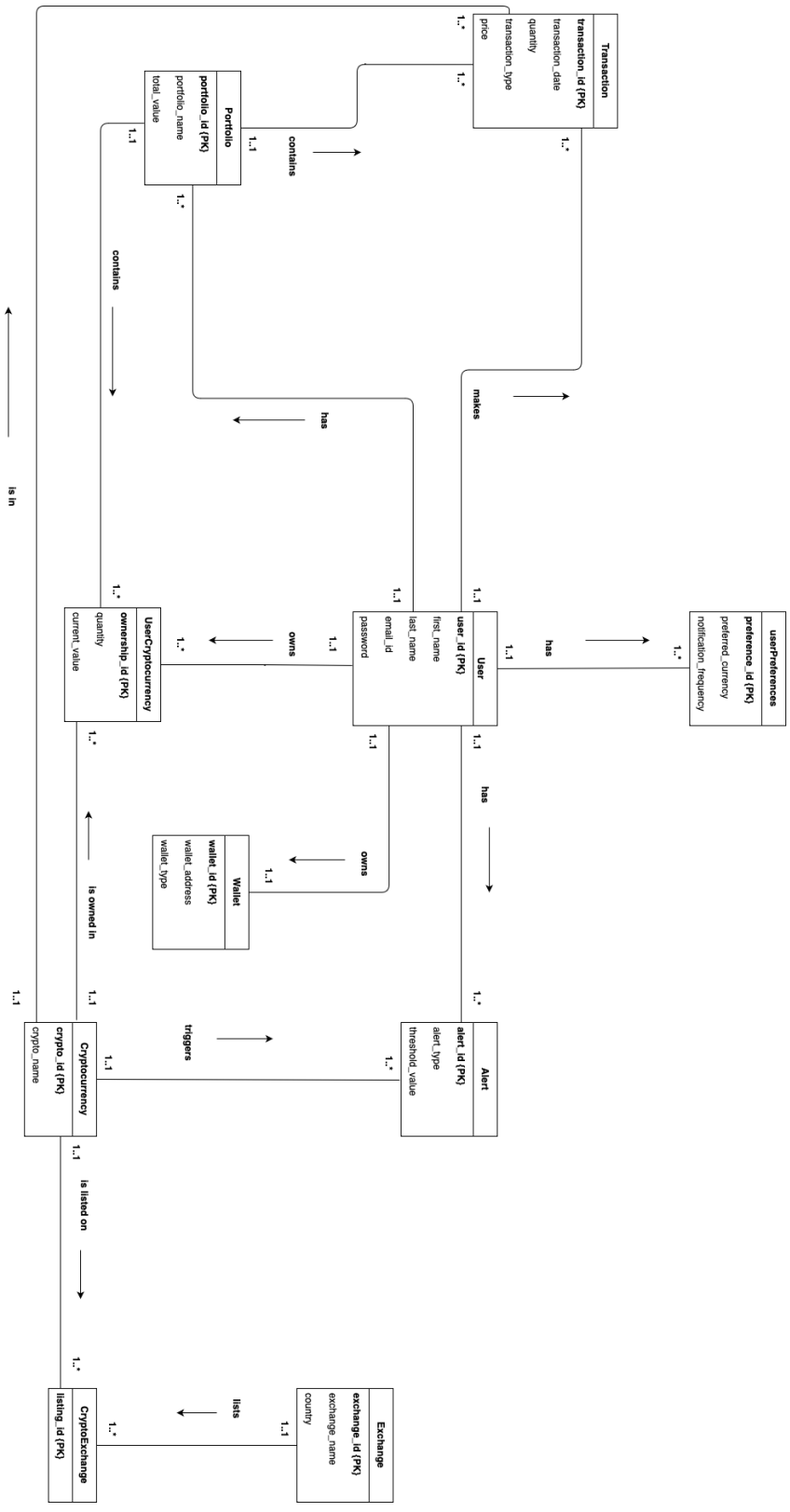
To run the application, navigate to the directory containing main.py in your command line, then run the command: "python main.py". Follow the prompts in your terminal to use the application.

NOTE: The data in the Cryptocurrency table and Exchange table have been prepopulated using INSERT statements in the SQL file to validate user data. The data in the rest of the tables have been prepopulated while interacting with our python application.

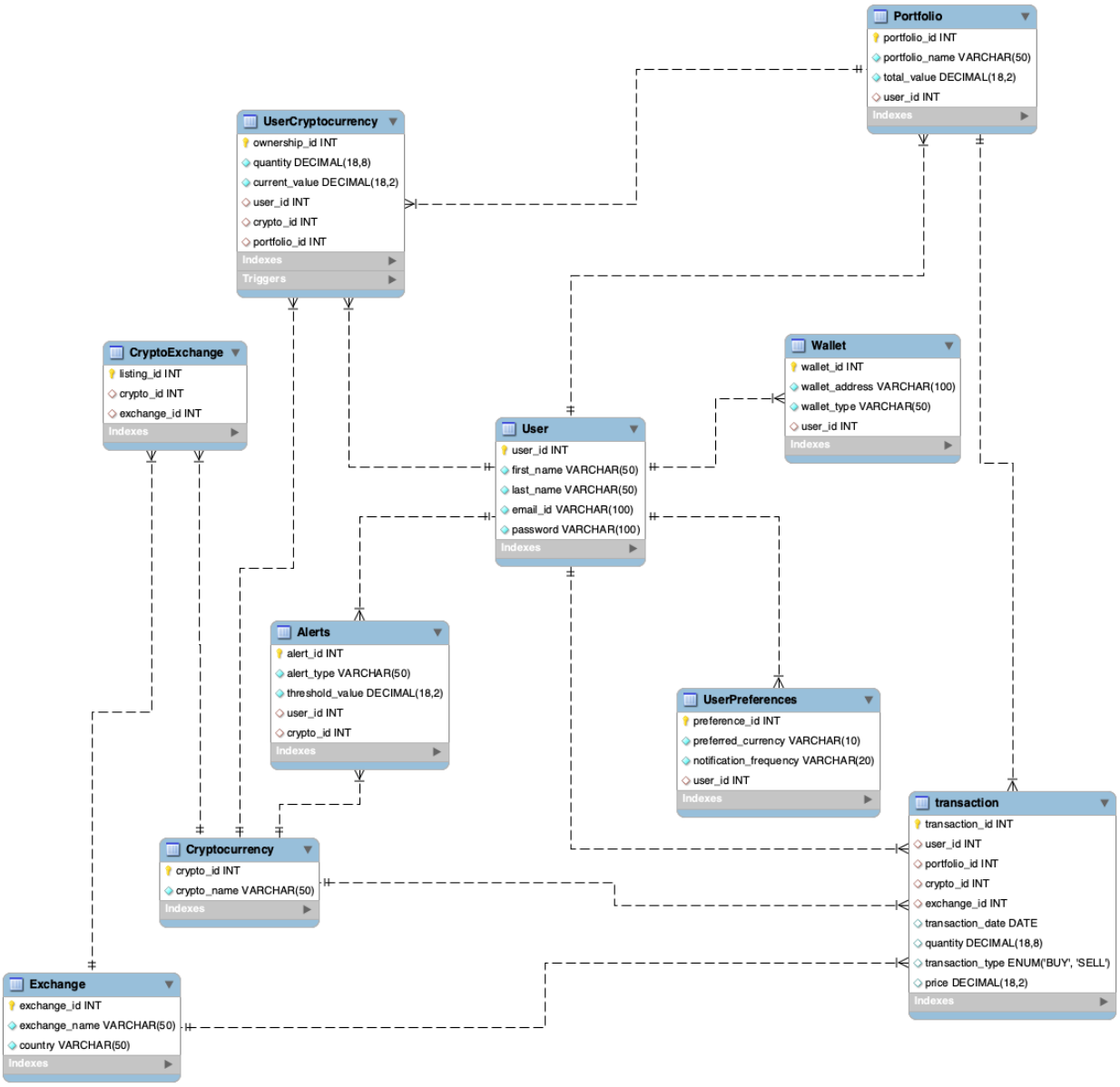
Assumptions

1. Immediate Transaction Input: We assume that a user inputs a buy or sell transaction into our system immediately after they execute a transaction on an exchange. This ensures that the data entered reflects the most recent transaction executed by the user.
2. Crypto Current Value: In light of our inability to use a real-time price API, we base the current value of the crypto on the price inputted by the user during a transaction. This assumption has implications for the current value of a crypto, total value of the portfolio, profit, and loss calculations, which rely on the user-provided value rather than a real-time market value.
3. Portfolio Total Value: Our total portfolio value calculations do not factor in profit or loss. Instead, they merely represent the aggregate of the current value of individual quantities of specified cryptos within a given portfolio. A user will be able to view the current profit/loss when interacting with our application.
4. Alerts, Preferences, and Wallet Functionalities: Certain functionalities such as Alerts, User Preferences, and Wallet do not have full operational capabilities due to the unavailability of real-time data. Without access to relevant APIs, these features serve primarily as placeholders for future development and enhancement.
5. Cryptocurrency and Exchange Validation: We have pre-populated our Cryptocurrency and Exchange tables with widely recognized cryptocurrencies and exchanges. This enables our system to validate user inputs against these tables, ensuring that users do not enter non-existent or obscure cryptos and exchanges. This further enhances data integrity and reliability within our application.
6. CryptoExchange Assumptions: When a user enters a buy or sell transaction, we make the assumption that the mentioned crypto is listed on the specified exchange. We do not validate this fact against any real-world data or API. This implies that if a user says they bought Bitcoin from Binance or sold Ethereum on FTX, our system assumes that these exchanges do indeed list these cryptos. This transaction information is then used to populate the CryptoExchange table, avoiding duplicates by not inserting already existing pairs of cryptos and exchanges.

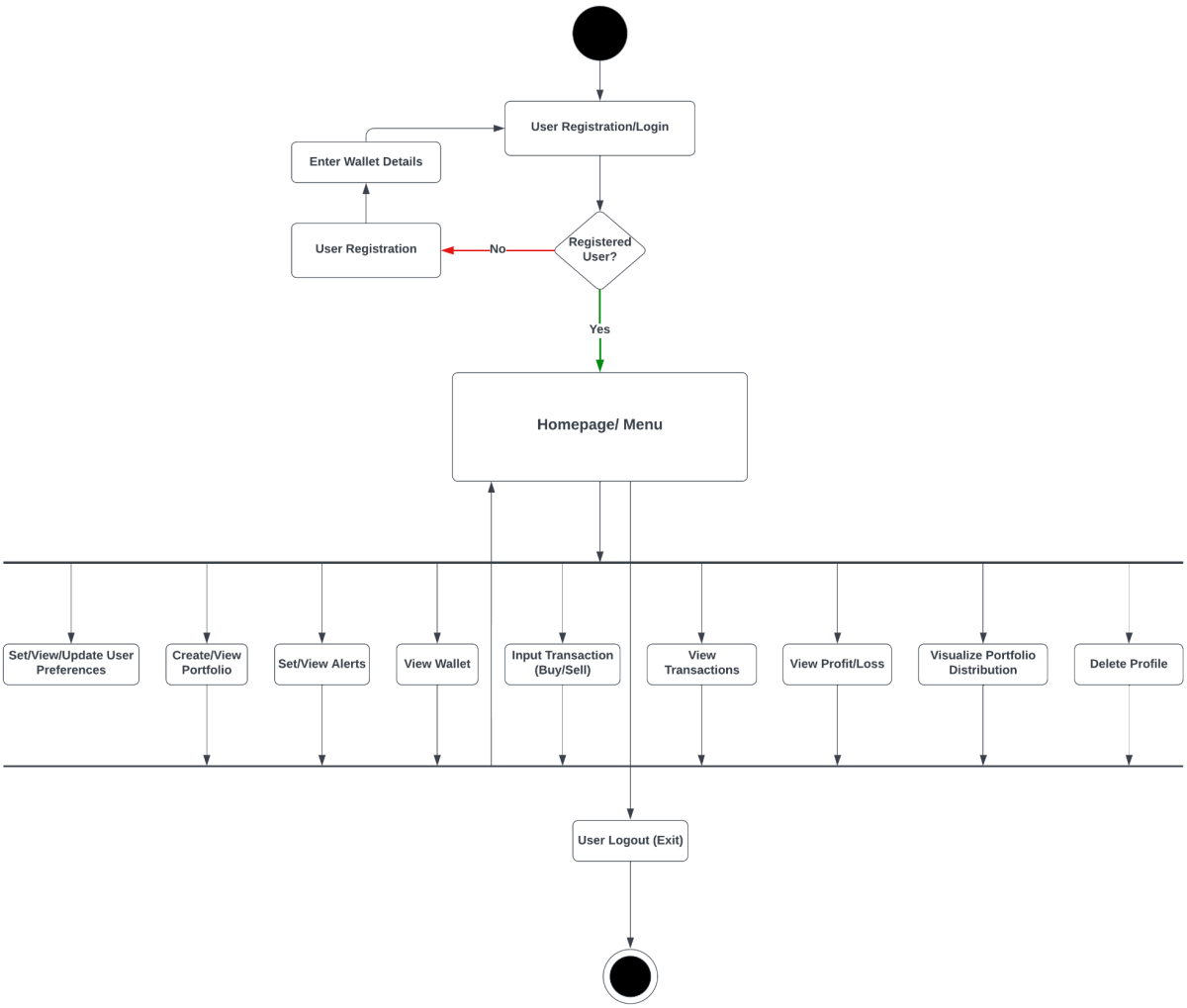
Conceptual Design



Logical Design



Activity Diagram - Final User Flow



Activity Diagram - Textual Description

1. User starts the program and inputs their MySQL credentials.
2. If the credentials are valid, the system connects to the MySQL database.
3. User is prompted to log in or register. After successful registration or login, first-time users must input their wallet details. Registered users won't be prompted for wallet details as we assume a user only has one wallet associated with all portfolios.
4. The user is presented with the Main Menu.
5. Depending on the user's selection from the menu, they are led through various processes such as setting preferences, managing portfolios, setting alerts, managing transactions, viewing profits/losses, visualizing portfolio distribution, or deleting their profile.
6. The user can repeatedly navigate the Main Menu until they choose to exit. On exit, the system disconnects from the database and the program ends.

Lessons Learned

1. **Technical Expertise Gained**

    a. Python: We've developed an in-depth understanding of Python programming, including its syntax, libraries such as pandas and matplotlib, and interacting with databases using mysql-connector-python.
    b. SQL: We've also learned how to work with MySQL, create complex SQL queries (CRUD), design database schema, and handle database transactions using functions, procedures, and triggers.
    c. Data Manipulation: Handling and manipulating data using pandas has given us data analysis and manipulation expertise and we've also learned how to visualize data easily.
    d. User Interaction: Designing a command-line interface for user interaction has given us insight into user experience and usability considerations in software design.

2. **Insights**

    a. Time Management: Breaking down the project into smaller tasks and working consistently over time proved to be more efficient than trying to do too much at once. Regular testing after each component was built helped to ensure that errors were caught and corrected early.
    b. Data Domain Insights: This project has also given us a deeper understanding of cryptocurrency trading, portfolio management, and the importance of real-time data in financial decision-making.

3. **Contemplated Alternative Approaches**

    a. Use of ORM: While using SQL for database interactions, we could have also considered using an ORM (Object-Relational Mapping) tool like SQLAlchemy for Python. ORMs can provide a higher level, more Pythonic interface to interact with databases.
    b. Real-Time Crypto Pricing: Another improvement could be the use of a cryptocurrency price API to fetch real-time prices of the cryptocurrencies. This would ensure that the user doesn't have to manually input the current price of a cryptocurrency when executing a buy or sell transaction, making the system more realistic and automated.
    c. GUI: While the project was completed with a command-line interface, we could consider creating a graphical user interface (GUI) for a better user experience in future iterations.

4. **Code**

    We would like to take the opportunity in this section to provide context regarding our code. For all major user functionalities and features, we have made use of stored procedures, functions, and triggers. However, to validate user input against data in the crypto_portfolio database, we have executed a few SQL queries within our Python application code.

Future Work

While our current cryptocurrency portfolio management system provides substantial functionalities, we recognize the potential for further development and enhancements. Some of the planned uses and potential areas for added functionality include:

1. Real-time Data Integration: As a primary area of future development, we plan to integrate real-time data sources into our application. This includes real-time cryptocurrency price data and exchange rates. The ability to capture real-time data can significantly improve our system's capabilities, making it possible to automate current value updates, calculate real-time profits and losses, and trigger alerts based on user-defined conditions.
2. Alerts System Enhancement: With real-time data, the alerts system can be enhanced to notify users immediately when specified market conditions are met. For example, an alert could be triggered when a particular cryptocurrency reaches a user-defined threshold. This feature will enhance the decision-making capabilities of our users.
3. Expanded Cryptocurrency and Exchange Support: We plan to broaden the range of cryptocurrencies and exchanges supported by our system. This will not only make our application relevant to a wider audience, but also improve our system's flexibility and adaptability to market changes.
4. User Interface Development: Currently, our application operates in a console-based environment. In the future, we plan to develop a user-friendly graphical interface. This would not only enhance the user experience but also provide visual representations of data, including charts and graphs for portfolio distribution, profit/loss trends, etc.

In conclusion, these enhancements would allow our system to provide a more comprehensive and accurate cryptocurrency portfolio management solution. We believe that this continued development will help our users make more informed and timely investment decisions.