Pavan Kumar Ramadugu - 1225163134

Ira A. Fulton Schools of Engineering
Arizona State University

CSE 535: Mobile Computing

ZenTrade - Recommendation Engine Component

Alignment with Guardian angel

The recommendation engine within ZenTrade plays a pivotal role in aligning with the "Guardian Angel" philosophy of the app. The recommendation engine's features and functionalities contribute to the Guardian Angel ethos in the following ways:

- Personalized Investment Recommendations: By considering personal information, risk appetite, stress levels, and past trading behavior, the engine creates investment suggestions uniquely suited to each user. This personalized approach is akin to a guardian angel's bespoke guidance, ensuring the financial advice is tailored to the user's individual profile.

- Emotional Risk Mitigation: The integration of stress level monitoring in the recommendation process ensures that investment decisions are influenced by the user's emotional state. This mirrors a guardian angel's role in protecting against emotionally driven, potentially harmful financial decisions.

- Data-Driven Decision Making: The engine's reliance on data from various sources for decision-making mirrors a guardian angel's use of wisdom and knowledge to guide and protect. The engine uses data to calculate investment metrics like returns, volatility, and momentum, ensuring recommendations are grounded in solid financial analysis.

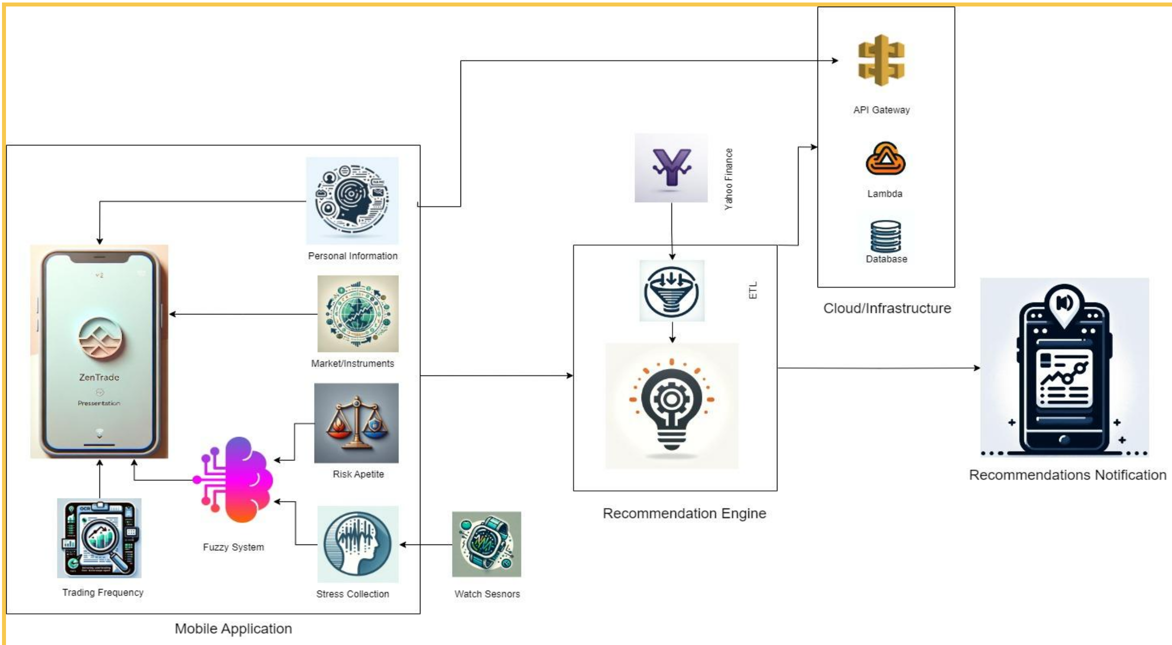
- Dynamic Adaptation: The recommendation engine's ability to adapt to changing market conditions and user preferences resembles a guardian angel's vigilance and flexibility, always seeking the best outcome for the user.

Design

TechStack Used: Python, Flask, ETL, Pandas, yfinance, niftystocks, pytickersymbols

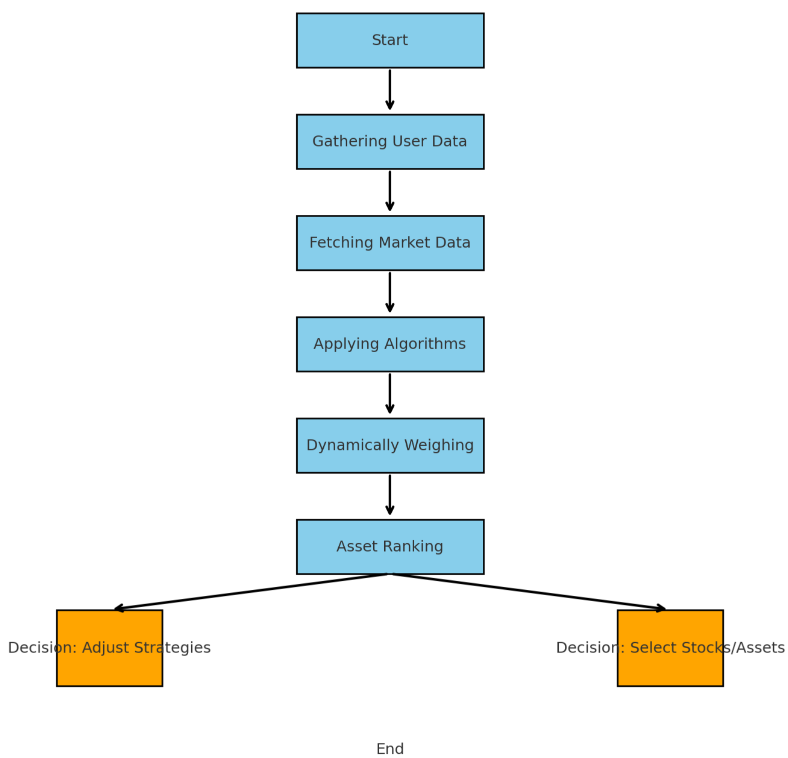
Overview: The process involves fetching adjusted close prices for specific tickers from Yahoo Finance, then calculating key investment metrics such as Returns, Volatility, Momentum, and Sharpe Ratio based on the user's trading activity. Weights are assigned to these metrics according to the user's portfolio factor (Conservative, Balanced, or Aggressive). Stocks are then ranked based on these weighted metrics, with top performers selected. Finally, stock recommendations are generated, tailored to the user's inputs regarding countries, frequency, and portfolio factor.

Pavan Kumar Ramadugu - 1225163134



Specifications

Creating a Control Flow Diagram (CFD) for the Recommendation Engine in the context of ZenTrade



Pavan Kumar Ramadugu - 1225163134

Testing strategies

As the developer of ZenTrade's recommendation engine, my testing strategy encompasses a comprehensive approach to ensure accuracy, efficiency, and alignment with the app's stress-smart investment philosophy. Initially, I focused on Unit Testing, validating individual functions like data fetching, metric calculations, and stock selection for correctness and efficiency. This is followed by Integration Testing, where I ensure seamless data flow and logical consistency throughout the recommendation process, from user input to final stock suggestions.

Further, I engaged in System Testing to validate the engine's integration within the larger ZenTrade system, including API functionality and its adaptation to user profiles and stress levels. Performance Testing is also crucial, where I assess the engine's response under high load scenarios and measure its computational efficiency. Lastly, Security Testing ensures user data privacy and protection against data breaches. This multi-layered testing strategy is designed to ensure that ZenTrade's recommendation engine is not only technically robust but also delivers personalized, stress-aware investment advice to our users.

Navigating Challenges

I encountered several challenges that tested my adaptability and problem-solving skills. One of the foremost challenges was ensuring the accuracy and relevancy of the recommendations in the face of dynamic and often unpredictable market data and varied user stress levels. This complexity made the algorithm's performance somewhat inconsistent initially. To navigate this, I focused on implementing robust data validation and error-handling mechanisms. I also made it a point to continually learn from real-world data and user feedback, which was instrumental in refining the algorithm to be more responsive to changing market conditions and user preferences.

Another significant hurdle was integrating diverse data sources like financial data, user risk profiles, and stress level readings. Ensuring a seamless blend of this information was critical for precise and personalized recommendations. However, I faced issues with data discrepancies in terms of quality, format, and timeliness, which initially hindered the integration process. To overcome this, I established strict data quality checks and developed a flexible data processing framework that could adeptly handle different data types. Regular updates and rigorous testing of the integration protocols became a part of my routine, helping me maintain a high level of accuracy and efficiency in the recommendation engine. These challenges, though daunting, proved to be invaluable learning experiences. They pushed me to enhance my skills and approach, leading to the development of a more resilient and user-centric investment advisory tool.