1. **Users (250)**:
    - Represented as individual nodes.
    - Each user has an associated account with assets (stocks or mutual funds).
2. **Portfolio Service**:
    - Central component responsible for managing portfolios.
    - Receives real-time data updates from various sources.
    - Scales horizontally to handle increasing user load.
    - Includes the following subcomponents:
        - **User Authentication and Authorization**: Ensures secure access to user portfolios.
        - **Portfolio Database**: Stores user portfolio data securely.
        - **Portfolio Calculation Engine**: Calculates portfolio values based on asset prices.
        - **Real-time Update Handler**: Listens for data updates from external sources.
3. **Data Sources**:
    - Represented as external nodes.
    - Provide asset prices and other relevant data.
    - Data is fetched periodically (every 10 minutes).
    - Includes stock market APIs, mutual fund data providers, and other financial data sources.
4. **User Dashboard**:
    - Each user has access to their personalized dashboard.
    - Displays real-time portfolio information.
    - Communicates with the Portfolio Service.
    - Includes features like asset allocation charts, historical performance, and alerts.

5. **Reliability and Scalability**:
    - Use load balancers and redundant servers for reliability.
    - Implement caching mechanisms to reduce latency.
    - Distribute data processing across multiple nodes.
    - Monitor system health and auto-scale based on demand.
6. **Security Measures**:
    - **Encryption**: Encrypt data in transit (HTTPS) and at rest (database encryption).
    - **Access Controls**: Role-based access control (RBAC) for users and administrators.
    - **Audit Logs**: Log user actions and system events for auditing.
7. **Synchronization**:
    - **Data Consistency**: Ensure consistent data across distributed nodes.
    - **Event-Driven Updates**: Use message queues or event-driven architecture for real-time updates.
    - **Scheduled Jobs**: Regularly synchronize data with external sources.
8. **Error Handling**:
    - **Graceful Degradation**: Handle service failures gracefully.
    - **Retry Mechanisms**: Retry failed requests to external sources.
    - **Alerts and Monitoring**: Set up alerts for critical errors.
    - **Fallback Strategies**: Provide fallback data if external sources are unavailable.

Visual representation:


Users (250)
↓
Portfolio Service
↓
User Authentication
and Authorization
Portfolio Database
Portfolio Calculation Engine
(Real-time Updates)
Real-time Update Handler
↓
Data Sources
(Stock Prices, Mutual Fund Data)
↓
User Dashboard
↓
External Services