Low Level Design (LLD)

Project Title: Cryptocurrency Liquidity Prediction for Market Stability

Data Ingestion

- Loaded two CSV files ('march16.csv' and 'march17.csv') using Pandas `read_csv()`.
- Merged the datasets for analysis.

Data Cleaning

- Dropped rows with missing values using `dropna()`.
- Removed duplicate records using `drop_duplicates()`.
- Converted relevant columns to appropriate data types when needed.

Feature Engineering

- Created moving average features:
    - `price_ma_2 = (price + price.shift(1)) / 2`
    - `mkt_cap_ma_2 = (market_cap + market_cap.shift(1)) / 2`
- Created volatility feature:
    - `volatility = (price.max() - price.min()) / price.mean()`
- Calculated liquidity ratio:
    - `liquidity_ratio = volume_24h / market_cap`

Exploratory Data Analysis (EDA)

- Plotted Bitcoin price trend over time using Matplotlib.
- Created a correlation heatmap using Seaborn `heatmap()`.
- Displayed summary statistics using `df.describe()`.

Model Building

- Splitted data into train and test sets using `train_test_split()`.
- Trained a basic Linear Regression model.
- Trained a Random Forest Regressor as the final model.
- Performed hyperparameter tuning if needed.

Model Evaluation

- Calculated performance metrics:
    - Root Mean Squared Error (RMSE)
    - Mean Absolute Error (MAE)
    - R^2 Score

Model Saving

- Saved the best model using Joblib:
    - `joblib.dump(model, 'liquidity_prediction_model.pkl')`

Local Deployment (Optional)

- A simple Streamlit or Flask app can be created to load the saved model and predict liquidity for new inputs.

LLD Flow Diagram: