

NBP Exchange Rate Analysis Tool

Technical Documentation

Jakub Bak 245762
Albert Brożyna 245779
Rafał Szkopik 245936
Daniel Szymczak 245941

July 2025

Contents

1	Introduction	2
2	System Overview	2
3	Supported Commands	2
4	Class: CurrencyManager	2
4.1	Key Methods	2
5	Helper Functions	3
6	System Architecture	3
6.1	Component Diagram	3
6.2	Sequence Diagram	4
6.3	Activity Diagram	5
7	Statistical Analysis	5
8	Histogram Generation	6
9	Example Use Case	6
10	Conclusion	7

1 Introduction

This application is a terminal-based tool designed for analyzing historical exchange rates retrieved from the official API of the National Bank of Poland (NBP). It supports interactive commands to download data, perform statistical analysis, visualize trends using histograms, and export results.

2 System Overview

The project consists of the following core components:

- **CurrencyManager** — A comprehensive class responsible for managing currency-related operations. Its main responsibilities include:
 - Fetching exchange rate data from the public NBP (Narodowy Bank Polski) API for selected currencies and time ranges.
 - Parsing and aggregating data from chunked API requests
 - Computing key statistical measures such as *median*, *mode*, *standard deviation*, and the *coefficient of variation* for a given dataset.
 - Analyzing exchange rate trends to identify the number of increasing, decreasing, or stable sessions over a selected period.
 - Exporting exchange rate data to a CSV file for further external processing or archival.
- **Main application** — Acts as the central controller for user interaction. It processes and validates commands, collects required inputs, and delegates specific tasks to the corresponding methods of the **CurrencyManager**.
- **Helper functions** — A collection of lightweight utility functions used to validate dates, sanitize input, verify currency codes, and parse command-line arguments. These ensure robustness and correctness of the overall application, reducing redundancy and improving maintainability.

3 Supported Commands

- **help** — Display the list of available commands.
- **list-currencies** — List all supported currencies.
- **fetch-data** — Fetch historical exchange rates from NBP.
- **session-analysis** — Analyze daily up/down/stable trends in the selected period.
- **statistics** — Compute key statistical metrics.
- **export** — Export to supported formats.
- **change-histogram** — Generate histogram for a currency pair.
- **exit** — Exit the application.

4 Class: CurrencyManager

4.1 Key Methods

`fetch_data(currency, start, end)` Fetches exchange rates for a currency within a specified date range. It handles API constraints (max 93 days per request).

`export_to_csv(data, filename)` Exports the data list to a CSV file.

`session_analysis(data)` Calculates how many sessions were rising, falling, or stable.

`compute_statistics(data)` Computes median, mode, standard deviation, and coefficient of variation (CV).

`generate_histogram(data, title)` Creates and saves a histogram using `matplotlib`.

`get_period_dates(period)` Converts short period strings (like `1m`) into date ranges.

`show_help()` & `show_available_currencies()` Print information for users.

5 Helper Functions

- `get_valid_date(prompt)` — Ensures the user provides a valid date string in YYYY-MM-DD format.
- `get_valid_currency(cm)` — Verifies if the input currency is supported.
- `get_valid_currency_pair(cm)` — Checks whether the currency pair is valid, including PLN-based pairs.
- `get_valid_currency_and_period(cm)` — Combines currency and time period selection with proper validation.

6 System Architecture

6.1 Component Diagram

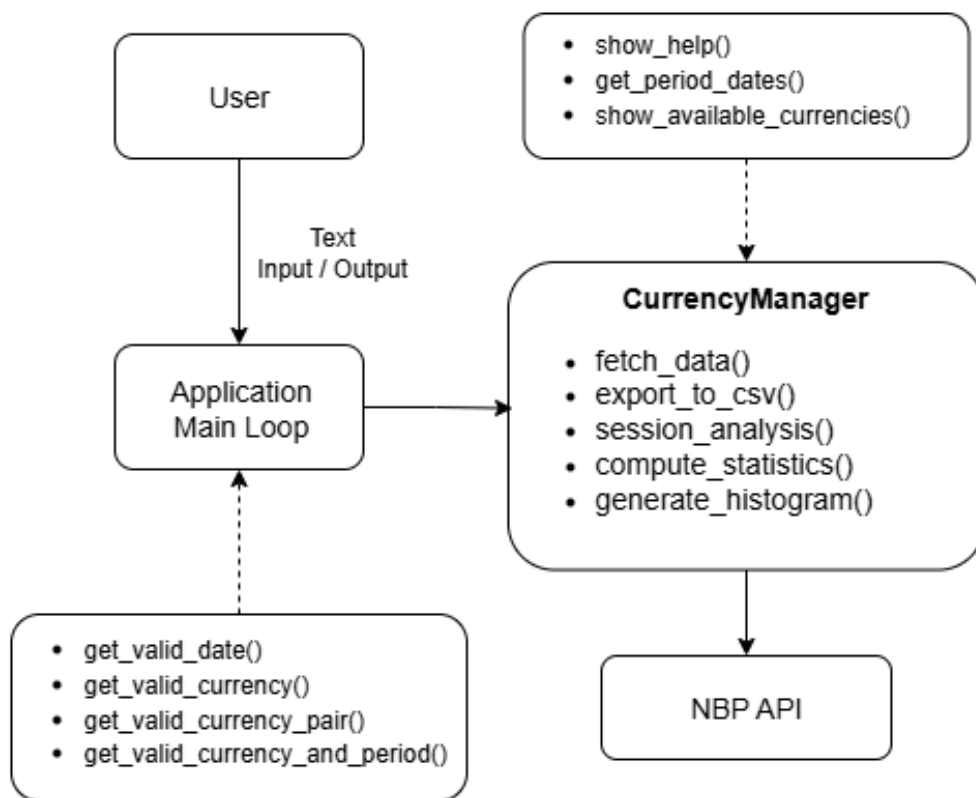


Figure 1: High-level component diagram of the application architecture.

6.2 Sequence Diagram

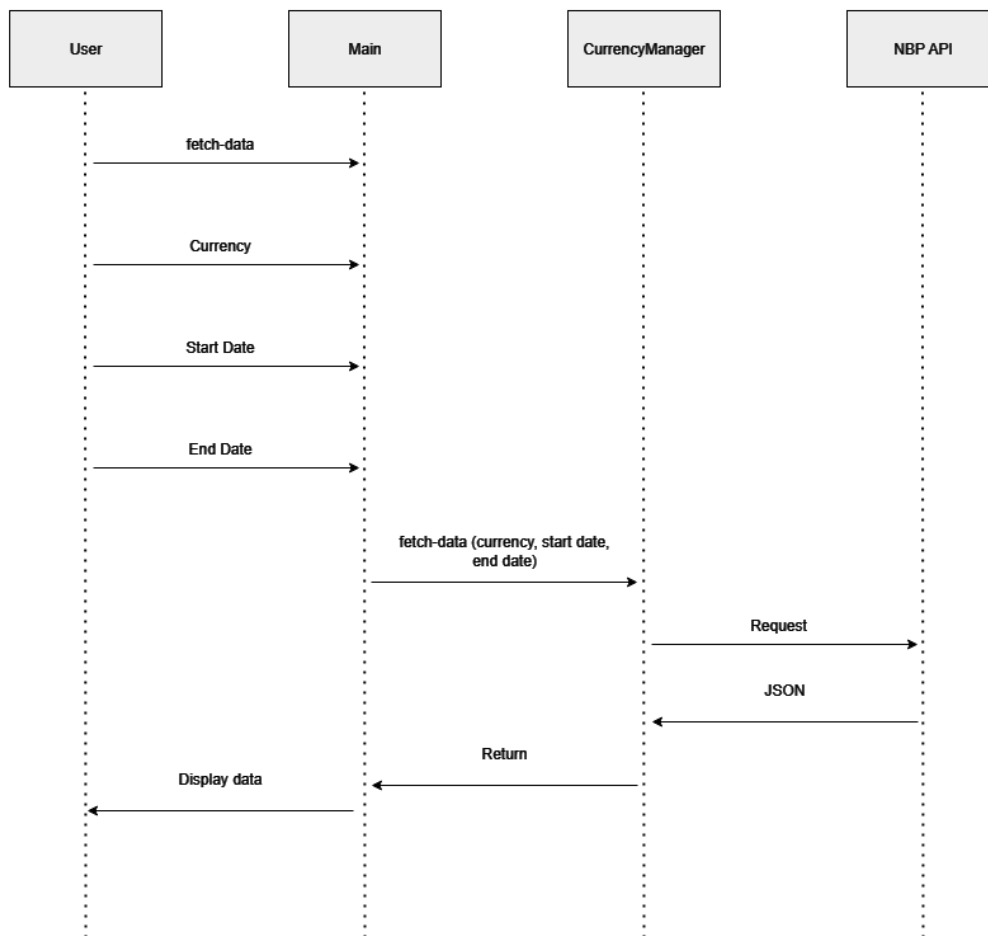


Figure 2: Sequence diagram depicting data fetching and analysis workflow.

6.3 Activity Diagram

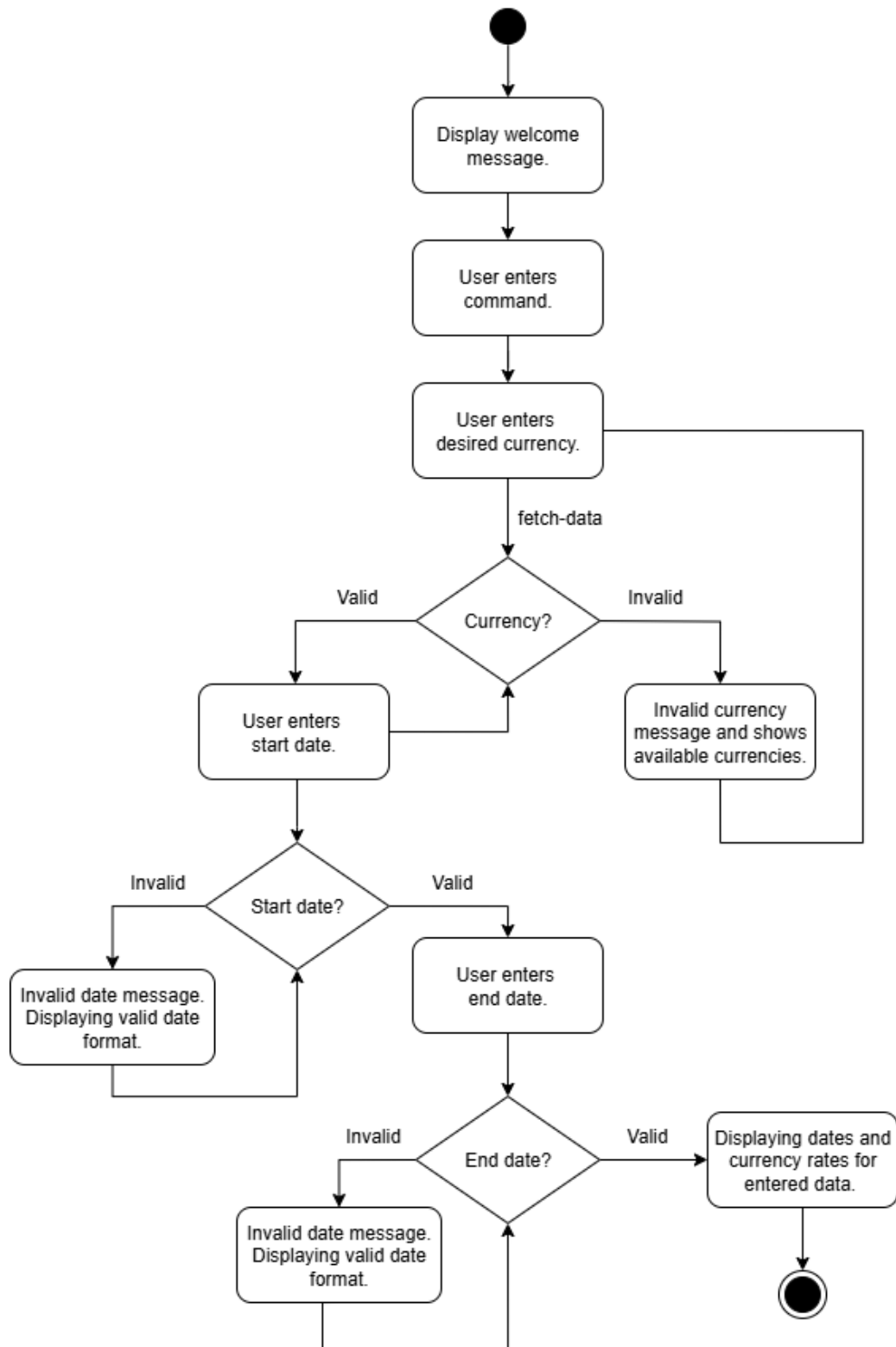


Figure 3: Activity diagram of user command processing and results display.

7 Statistical Analysis

When running the `statistics` command, the following values are calculated:

- **Median** — Middle exchange rate value.
- **Mode** — Most frequent rate.
- **Standard Deviation (σ)** — Dispersion of exchange rate values.
- **Coefficient of Variation (CV)** — Percentage-based measure:

$$CV = \frac{\sigma}{\mu} \cdot 100\%$$

where μ is the average rate.

8 Histogram Generation

Using `change-histogram`, users can:

- Choose a base and quote currency.
- Select a period (1 month or 1 quarter).
- Select start date.
- The app computes relative exchange rates.

Histograms are plotted using `matplotlib`, saved as `histogram.png`, and displayed.

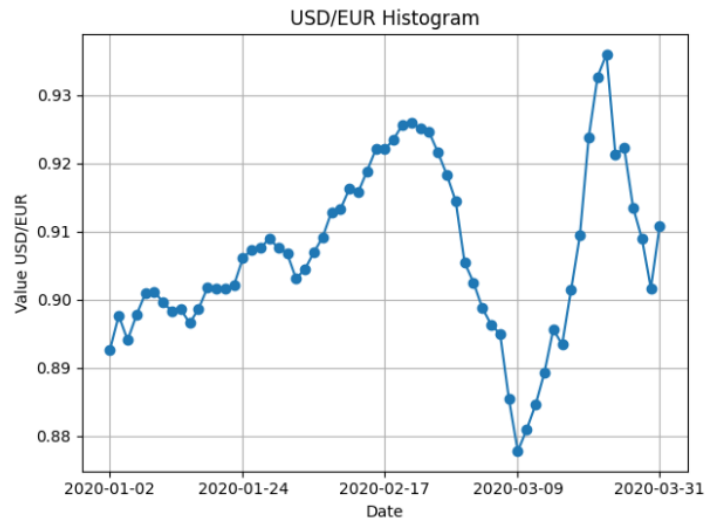


Figure 4: Example currency histogram

9 Example Use Case

Scenario

An economist wants to analyze the USD/EUR exchange rate behavior for the quarter starting on 2020-01-01.

1. Launch the application.
2. Run `change-histogram`.
3. Input USD/EUR, period: `1q`, start date: 2020-01-01.
4. The histogram is displayed and saved.

Command Output

```
Enter command: change-histogram
Currency pair (e.g. USD/EUR): USD/EUR
Period (1m or 1q): 1q
Start date (YYYY-MM-DD): 2020-01-01
[Histogram appears]
```

10 Conclusion

This project effectively demonstrates how to integrate real-world data from a public API and build a robust analysis tool with Python. The modular design ensures scalability, and the CLI interface provides a lightweight way to explore historical financial data.

Possible future enhancements:

- GUI version (e.g., using PyQt or Tkinter).
- Additional statistics (e.g., linear regression, Sharpe ratio).
- Multi-currency correlation analysis.
- Command history and saving user sessions.