

HIGH LEVEL DESIGN **(HLD)**

DELOITTE CASE STUDY

Revision Number: 1.0

Last date of revision: 21/12/2024

Hitesh Gupta

Contents

| | | |
|----|---|---|
| 1. | Document Version Control | 3 |
| 2. | Abstract | 4 |
| 3. | 1. Introduction | 4 |
| | 3.1 Why this High-Level Design Document? | 4 |
| | 3.2 Scope | 4 |
| 4. | 2. General Description | 5 |
| | 4.1 Product Perspective & Problem Statement | 5 |
| | 4.2 Tools Used | 5 |
| 5. | 3. Design Details | 6 |
| | 5.1 Functional Architecture | 6 |
| | 5.2 Optimization | 8 |
| 6. | 4. KPIs | 8 |
| | 6.1 KPIs (Key Performance Indicators) | 8 |
| 7. | 5. Deployment | 9 |

Document Version Control

| Date | Version | Description | Author |
|-------------|---------|-------------------------------|--------------|
| 21 Dec 2024 | 1.0 | First Version of Complete HLD | Hitesh Gupta |
| | | | |

Abstract

The purpose of this project is to develop a financial analysis dashboard that uses time-series data from The World Bank to analyze economic indicators such as the Consumer Price Index (CPI), Exchange Rates, and Exports (in millions) for 200 countries. The dashboard will be designed to visualize trends and perform advanced analysis, including Year-over-Year (YoY) and Compound Annual Growth Rate (CAGR) calculations. The user interface will include dynamic graphical representations of the data that adjust according to user inputs via slicers for country and time period (monthly, quarterly, yearly). This project will not only help users interact with complex economic data but also provide key insights for financial decision-making.

1. Introduction

1.1 Why this High-Level Design Document?

This High-Level Design (HLD) document is created to provide a comprehensive overview of the architecture, design approach, and components involved in the development of the financial analysis dashboard. It serves as a blueprint for the development team and stakeholders, detailing the structure of the project, the tools involved, and the methods used to deliver the system. By documenting the design process, the team ensures that the final product meets the requirements and functions efficiently, providing users with valuable insights into financial data.

1.2 Scope

The scope of this project is to develop an interactive financial analysis dashboard using time-series data from The World Bank. The dashboard will focus on key economic indicators, including CPI, Exchange Rates, and Exports data for 200 countries. The system will allow users to:

1. Explore the relationship between economic indicators and trends over time.
2. Visualize data through dynamic graphs and charts that change based on user-selected filters such as country and time period (monthly, quarterly, yearly).
3. Perform advanced financial analysis, including YoY comparisons and CAGR calculations, to evaluate growth and trends in the dataset.
4. Improve data accessibility by structuring and cleaning the unorganized dataset, making it more useful for analysis.

This project will provide users with a powerful tool to assess financial indicators and make informed decisions based on economic trends.

2. General Description

2.1 Product Perspective & Problem Statement

This project addresses the need for better insights into the relationship between economic indicators and financial markets. The banking and financial services industry relies heavily on economic data to make informed decisions. This system will focus on key economic indicators such as the Consumer Price Index (CPI), Exchange Rates, and Exports data, sourced from The World Bank, to help users track and predict financial trends.

The dataset provided includes time-series data for 200 countries, covering variables such as CPI, Exchange Rate, and Exports (in millions). The data is unstructured and needs to be organized to extract meaningful insights. The dashboard will allow users to:

1. Structure the unorganized data and derive insights.
2. Apply slicers for countries and time periods (monthly, quarterly, and yearly) to analyze trends.
3. Perform financial analysis such as YoY and CAGR calculations to identify growth patterns.
4. Provide dynamic, interactive graphs and charts that adjust based on the slicer inputs.

The challenge is to create an interface that not only allows for these analyses but also enables users to understand complex relationships within the data through clear, dynamic visualizations.

2.2 Tools Used

This project will employ a range of tools and technologies for data processing, analysis, and visualization:

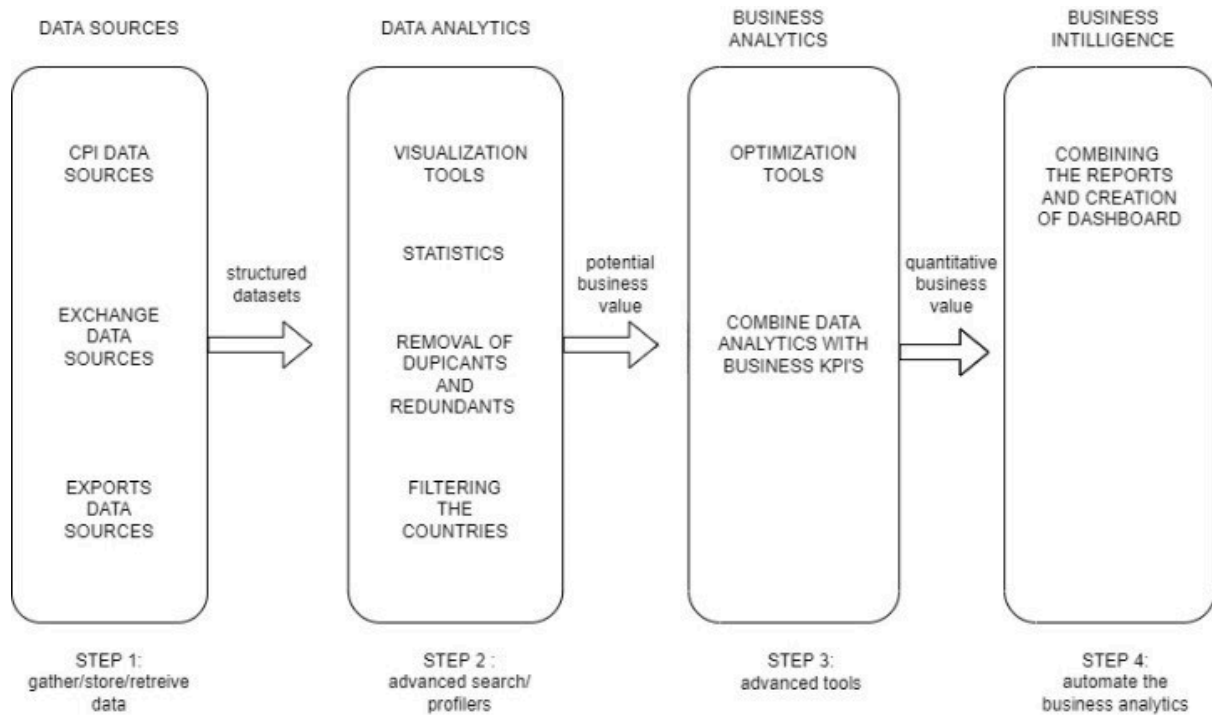


3. Design Details

3.1 Functional Architecture

The functional architecture of this project includes the following modules and components:

1. Data Acquisition Module:
 - Fetch data from The World Bank API or CSV files and store it in a structured format (CSV, SQL database).
2. Data Preprocessing and Cleaning:
 - Clean and preprocess raw data to remove missing values, outliers, and irrelevant entries.
 - Restructure the unorganized data into a format that is easier to analyze (e.g., converting raw data into time-series format for each country).
3. Financial Analysis Engine:
 - Calculate YoY, CAGR, and other financial metrics based on user-selected parameters (country, time period).
 - Analyze trends and provide statistical summaries of CPI, Exchange Rates, and Exports for different time periods.
4. Visualization Engine:
 - Generate graphs and charts that display trends for CPI, Exchange Rates, and Exports, dynamically updating based on user selections.
 - Provide options for data comparisons (monthly, quarterly, yearly), with drill-down capabilities for further analysis.
5. User Interface:
 - Provide a user-friendly interface with slicers for selecting country, time period, and other parameters.
 - Display interactive visualizations, allowing users to explore and analyze the data effectively.



3.2 Optimization

Optimization strategies are key to ensuring the system runs efficiently, particularly with large datasets and real-time data interactions:

1. Data Storage Optimization:
 - Use optimized data storage solutions (e.g., SQL databases or cloud storage) to ensure fast data retrieval and processing.
 2. Model Optimization:
 - Ensure efficient calculation of financial metrics (YoY, CAGR) with optimized algorithms to handle large datasets without performance degradation.
 3. Visualization Optimization:
 - Ensure that visualizations are responsive, adjusting quickly when slicers are applied, through the use of caching and efficient rendering techniques.
 4. Data Query Optimization:
 - Use indexing and efficient querying techniques to retrieve relevant data quickly when users apply filters.
-

4. KPIs

4.1 KPIs (Key Performance Indicators)

To assess the success of the project, the following KPIs will be tracked:

1. **Accuracy of Financial Metrics:**
 - Ensure that calculations for YoY and CAGR are accurate and consistent with financial standards.
2. **Visualization Interactivity:**
 - Measure user engagement with dynamic charts, assessing how frequently users interact with the slicers and graphs.

3. Data Processing Speed:

- Track the time taken for data preprocessing, analysis, and the generation of visualizations, ensuring that the system operates efficiently.

4. User Satisfaction:

- Gather feedback from users on the usability of the dashboard and the clarity of the visualizations, using surveys or user interviews.

5. System Performance:

- Monitor system performance, ensuring that it can handle large datasets and high user traffic without slowing down.
-

5. Deployment

The deployment of the financial analysis dashboard will involve the following steps:

1. Environment Setup:

- Set up the necessary environment for deploying the dashboard, whether on-premise or using a cloud platform like AWS or Azure.

2. Model and Visualization Deployment:

- Deploy the financial analysis engine and interactive dashboards on a web server, ensuring real-time data updates and scalability.

3. Testing:

- Conduct rigorous testing to ensure that the system works under various scenarios, such as different data volumes and user interactions.

4. User Training and Documentation:

- Provide user documentation and training materials to help users understand how to interact with the dashboard and perform analyses.

5. Monitoring and Maintenance:

- Implement monitoring tools to track system performance and errors, with regular maintenance and updates based on user feedback and system performance.