

DECEMBER 11, 2023

EXCHANGE RATE RECOMMENDATION SYSTEM

DESIGNING ARCHITECTURE

Author: Hira Arif



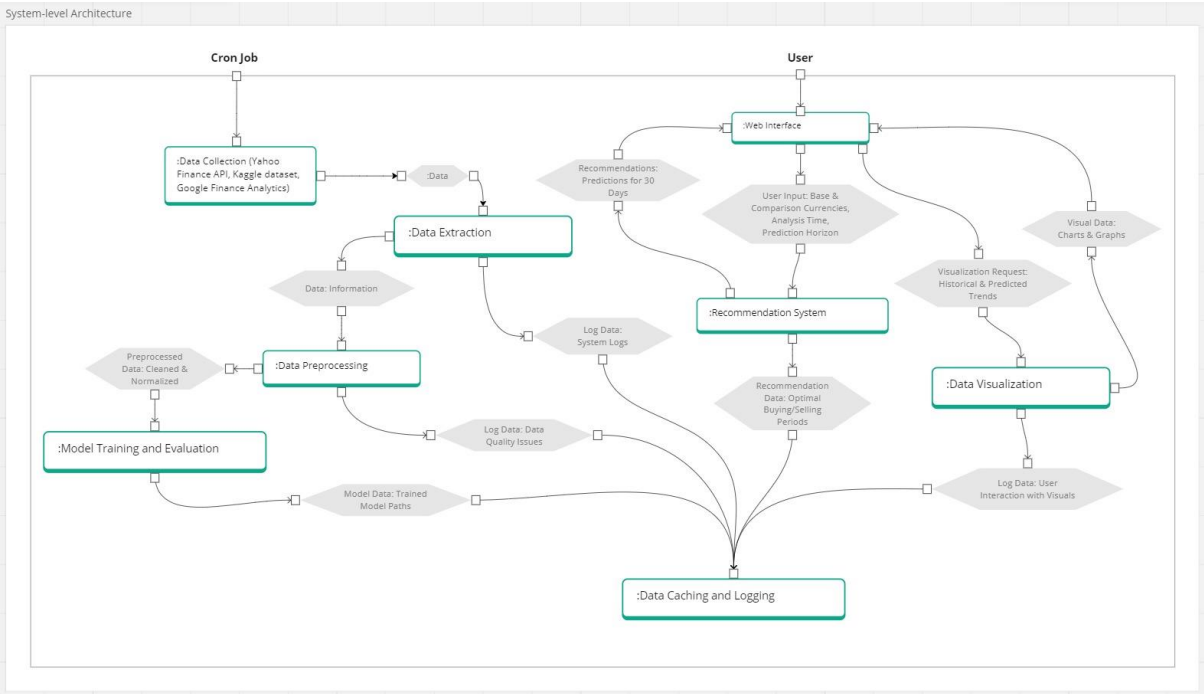
Table of Contents

1	Descriptive Architecture Creation:	3
1.1	Structure:	3
1.2	Behavior:	3
1.3	Interactions:	5
1.4	Non-Functional Requirements:	5
1.5	Implementation:	6
1.5.1	Technologies:	6
1.5.2	Development Tools:	6

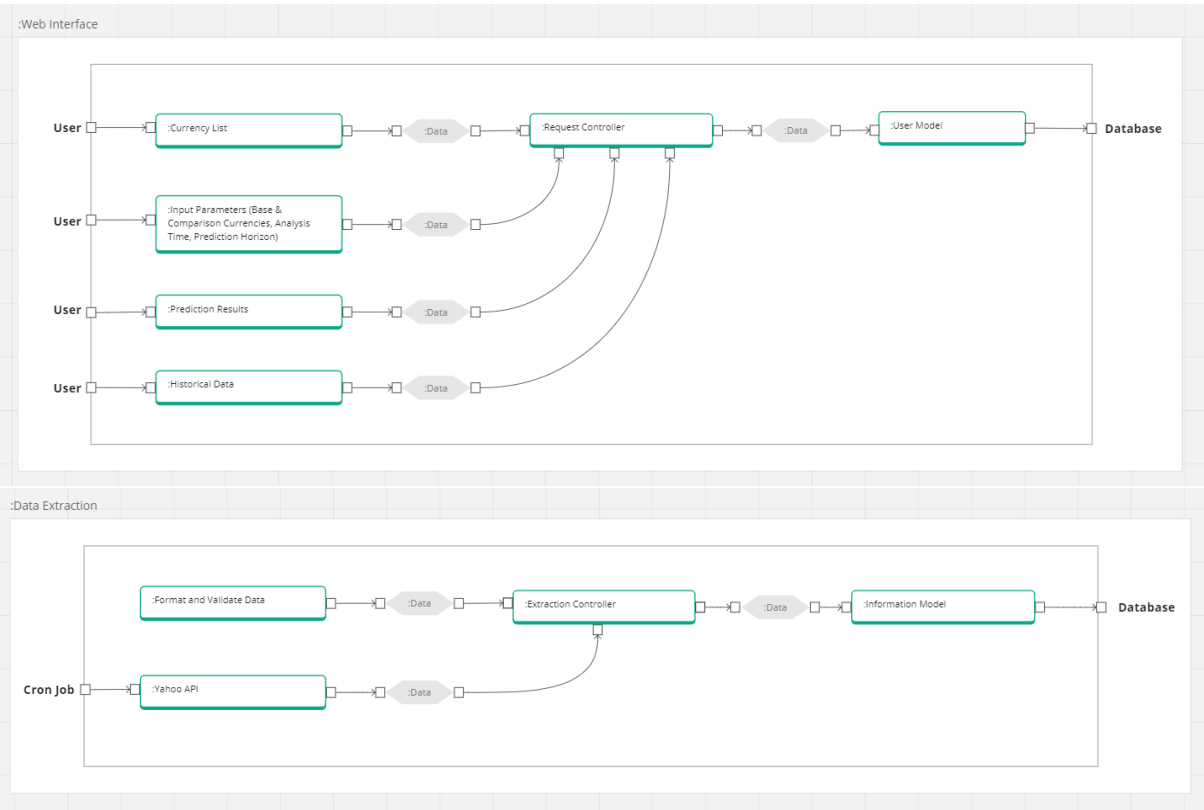
1 Descriptive Architecture Creation:

The following are the outcomes of five principal design decisions.

1.1 Structure:



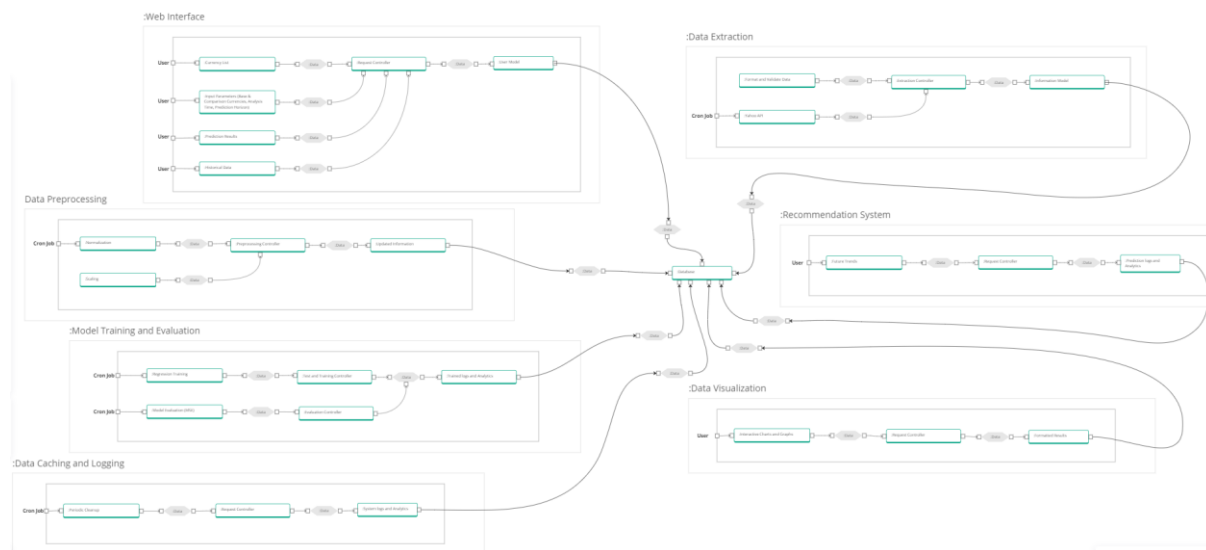
1.2 Behavior:



Exchange Rate Recommendation System



1.3 Interactions:



1.4 Non-Functional Requirements:

No Functional Requirement	Implementation Strategy	Impact on the System
Modularity and Extensibility (NFR01)	Design the system with modular PHP components, each limited to 2000 lines. Support seamless addition of new modules without disrupting the core.	Promotes a codebase that's easy to manage and extend, allowing the hassle-free integration of new functionalities.
Response Time (NFR02)	Optimize front-end code (HTML, CSS, JavaScript) and use asynchronous operations. Employ server-side caching and efficient PHP code for quick responses.	Ensures a snappy and responsive user interface, minimizing wait times for data retrieval.
Scalability (NFR03)	Leverage PHP's scalability features and implement server clustering. Optimize database queries for handling at least 1000 concurrent users without sacrificing response time.	Allows the system to smoothly scale, maintaining optimal performance under increasing user loads.
Reliability (NFR04)	Implement robust error handling using PHP exceptions and perform regular system maintenance. Communicate planned maintenance to ensure a system uptime of at least 99.9%.	Establishes a reliable system with minimal downtime, proactively addressing errors for seamless user experience.
Data Storage Efficiency (NFR05)	Optimize MongoDB storage to limit redundancy and regularly clean up obsolete data. Implement efficient indexing and querying.	Ensures efficient data storage, preventing unnecessary database growth and optimizing resource usage.

<i>Code Maintainability (NFR06)</i>	Document 80% of the PHP codebase with clear inline comments and adhere to PHP coding standards. Utilize Git for version control.	Facilitates easy maintenance, development, and collaboration, ensuring a well-documented and organized codebase.
<i>Error Recovery (NFR07)</i>	Implement a robust PHP exception handling system for graceful degradation. Set up automated alerts and aim for a quick recovery in case of critical failures.	Ensures uninterrupted user access by gracefully handling errors, minimizing system downtime and impact.
<i>Generalizability (NFR08)</i>	Design the system to be platform-agnostic, supporting Windows, Linux, and macOS. Ensure compatibility with various data sources without major code changes.	Enhances system flexibility, allowing deployment on diverse platforms and easy integration with different data sources.
<i>Resource Utilization (NFR09)</i>	Comply with data privacy regulations (e.g., GDPR) through secure PHP data handling. Document and audit data processes for transparency and legal compliance.	Builds trust by prioritizing user data protection and adhering to legal standards in data handling.
<i>Concurrent API Requests (NFR10)</i>	Optimize PHP scripts for handling concurrent API requests efficiently. Implement asynchronous processing and leverage PHP's threading capabilities.	Ensures responsive data extraction, preventing a significant increase in response time during high API request volume.

1.5 Implementation:

The Currency Exchange Rate Prediction System will be implemented as a web-based application using the REST principle and MVC architectural styles, employing HTML, CSS, and JavaScript for the frontend, and PHP for the backend, ensuring scalability, modularity, and a responsive user interface. The development tools, including Visual Studio Code, will be employed for efficient coding and collaboration.

1.5.1 Technologies:

- Frontend: HTML, CSS, JavaScript
- Backend: PHP
- Data Storage: MongoDB
- Version Control: GitHub
- Data Handling: APIs, CSV files, databases, web scraping
- Data Preprocessing: PHP and Python for scripting

1.5.2 Development Tools:

- Code Editing: Visual Studio Code
- Documentation: Inline comments for at least 80% of the codebase
- Model Training: Python scripts and appropriate tools