# 1. News Crawler Requirements

## 1.1. **Introduction**

News crawler gather and collect relevant news articles and information from various news sources. The news crawler is designed to search and index news content based on specific keywords, topics, or categories. This process allows the news recommendation system to provide users with personalized and up-to-date news articles that align with their interests and preferences. By continuously crawling and analyzing news data, the recommendation system can offer a curated selection of articles that are most likely to be of interest to the user, enhancing their overall news reading experience.

## 1.2. **Functional Requirements**

Functional requirements of news crawler system outline the features and functionalities necessary for the crawler to effectively gather, process, and manage news articles from various online sources. These requirements include tasks such as Data collection, Content filtering, Dynamic pipeline operations, Monitoring, Search and retrieval, Security, scalability, and compliance with legal and ethical standards. ### 1.2.1. **Data Collection** - **Web crawling**: The crawler should be able to navigate through web pages, following links to discover and retrieve news articles from designated sources.. - **Content Extraction**: The crawler shall be able to extract relevant information such as the article title, author, publication date, content, and URL. ### 1.2.2. **Content Filtering** - The system shall provide options to filter news articles based on predefined criteria such as keywords, categories, or publication dates. ### 1.2.3. **Dynamic pipeline operations** - **Automation** | **Manual pipeline**: The system shall have scheduling capabilities to automate/or manual the crawling process at predefined intervals. - **Performance Optimization**: The crawler should be optimized for performance to handle a high volume of concurrent crawling requests efficiently - **Error Handling**: It should incorporate mechanisms for handling errors gracefully, including retrying failed crawling tasks and logging error messages for troubleshooting. ### 1.2.4. **Monitoring** - System captures various events, activities, and messages generated during the crawling process: - Start and end times of crawling tasks - URLs visited and crawled - Success or failure of each crawling attempt - Errors encountered during crawling - Resource consumption metrics (e.g., memory usage, CPU utilization) - Relevant metadata of crawled content (e.g., publication date, author) - It shall provide monitoring tools to track the status of crawling tasks and detect any failures or issues. ### 1.2.5. **Security** - The system should implement authentication and authorization mechanisms to restrict access to authorized users and protect sensitive data.

## 1.3. **Non-Functional Requirements**

Non-functional requirements (NFRs) of a news crawler refer to the characteristics and constraints that govern the system’s operation and performance, rather than specific features or functionalities. These requirements describe how the system should behave, perform, and interact with its environment. ### 1.3.1. **Performance** - The system shall be capable of handling a high volume of concurrent crawling requests efficiently by multi-processing or multi-thearding. - It shall have low latency for both crawling and retrieval operations. ### 1.3.2. **Scalability** - The system architecture shall be designed to scale horizontally to accommodate increased workload and data volume. - It shall support distributed processing and storage to distribute the load across multiple nodes. ### 1.3.3. **Reliability** - The system shall be robust and resilient to handle failures gracefully. - It shall incorporate mechanisms for retrying failed crawling tasks and recovering from errors. ### 1.3.4. **Maintainability** - The system shall be modular and well-documented to facilitate ease of maintenance and future enhancements. - It shall support version control and automated testing for codebase stability. ### 1.3.5. **Compliance** - The crawler should comply with copyright laws and terms of service of crawled websites to ensure legal and ethical use of content. - System shall not violate any content usage policies or regulations.

## 1.4. **Cloud Platform Requirements**

### 1.4.1. **Compute Engine:**

* Provision virtual machines to host the crawler service.
* Ensure sufficient CPU and memory resources to handle crawling tasks efficiently. ### 1.4.2. **Storage**
* Utilize Google Cloud Storage (GCS) / BigQuery for storing crawled data, logs, and configuration files. ### 1.4.3. **Networking**
* Configure Virtual Private Cloud (VPC) networks to secure communication between components. ### 1.4.4. **Identity and Access Management (IAM)**
* Define IAM roles and permissions to control access to GCP resources.
* Implement least privilege principles to restrict access to sensitive data and services. ### 1.4.5. **Monitoring and Logging**
* Write log to BigQuery datatable and use Locker with dashboard to track
* ~~Utilize Stackdriver Monitoring to track resource usage, performance metrics, and health checks.~~
* ~~Configure Stackdriver Logging to capture logs and events generated by the crawler service.~~ ### 1.4.6. **Security**
* Enable VPC Service Controls to restrict data access within specified geographical boundaries.
* Implement Firewall Rules and Security Groups to control inbound and outbound traffic. ### 1.4.7. **Deployment and Orchestration**
* Use Google Kubernetes Engine (GKE) to deploy and manage containerized crawler applications.
* Leverage Cloud Build for continuous integration and deployment pipelines. ### 1.4.8. **Data Processing**
* Utilize Google Cloud Dataflow for stream and batch processing of crawled data.
* Integrate with BigQuery for data warehousing and analytics capabilities. ### 1.4.9. **Scalability**
* Configure autoscaling policies to automatically adjust compute resources based on workload demand.
* ~~Utilize managed services like Memorystore for Redis to handle caching and improve performance.~~ ### 1.4.10. **Cost Management**
* Implement budget alerts and billing reports to monitor and control costs.
* Utilize pricing calculators to estimate the cost of running the crawler service based on resource usage. ### 1.4.11. **Integration and APIs**
* Integrate with Google Cloud APIs for services like Cloud Storage, Pub/Sub, and BigQuery.
* Leverage third-party APIs for additional functionality such as text analysis or content enrichment. ## 1.5. **Assumptions and Dependencies**

### 1.5.1. **Internet Connectivity**

* The system assumes a stable internet connection for accessing and crawling online news sources.
* It may be affected by network latency and bandwidth limitations. ### 1.5.2. **Third-Party Dependencies**
* The system may depend on third-party libraries or APIs for certain functionalities such as HTML parsing or text extraction.
* It shall manage dependencies and ensure compatibility and version constantly with external services. ## 1.6. **Data sources Information requirements**

| Requirement | Description | Example |
| --- | --- | --- |
| Source URL | The URL of the news source website or RSS feed. | https://www.example.com/news |
| Source Type | Specify if the source is a website, RSS feed, social media platform, etc. | Website |
| Content Categories | List of categories or topics covered by the news source (e.g., politics, sports). | Market, Trading, Macro |
| Language | Language(s) in which the news articles are published. | Vietnamese |
| Geographic Region | The region or country covered by the news source. | Viet Nam |
| Update Frequency | How often the news source publishes new articles (e.g., hourly, daily). | Daily at 6AM |
| Access Restrictions | Any access restrictions or subscription requirements for accessing the source. | None |
| Data Format | Format of the data provided by the source (e.g., HTML, XML, JSON). | HTML |
| Content Extraction Rules | Rules or patterns for extracting relevant information from the source’s content. | Headline: <h1>, Article Body: <p> |
| Authentication | Authentication method required to access the source (e.g., API key, OAuth). | API Key |
| Pagination | If the source paginates its content, specify pagination rules for retrieving articles. | Pagination parameter: page=1 |
| Filtering Options | Any options for filtering articles based on criteria such as keywords or dates. | Keywords: “COVID-19”, Date Range: 2022-01-01 to 2022-12-31 |
| Error Handling | Procedures for handling errors or failed requests when accessing the source. | Retry 3 times, log errors |