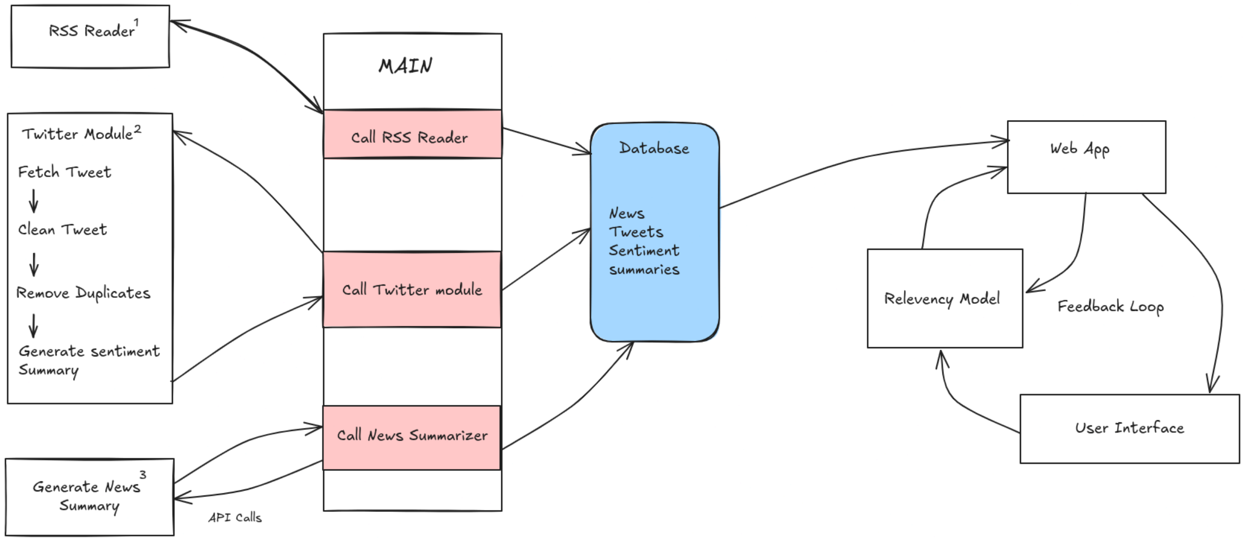
Low Level Design

Diagram



1. RSS Reader

Rss reader, is a module which will take RSS Feed and extract the news along with the metadata. And dump it in the database after checking for the duplicate content.

2. Twitter Module

Twitter module search for the database with the news headlines without any tweet sentiment summarize, extract tweets using twikit library, clean the tweets with the help of regex expressions, remove duplicate tweets to prevent redundancy and generate the sentiment summary, returning it to the main call, which will update the database.

3. News Summary

It extracts the news headlines from a database which has no news summary, get the article link, and extracts the whole article text from the respective site. And do a summarization using the LLM inference model.

It works like an asynchronous pipeline, although modules depend on each other in a sequential manner, still made to work in an asynchronous manner to optimize the flow and introduce parallelization.

APIs for Modules

Fast APIs have been used for the communication between the modules. These API addresses are mentioned below.

```
API_URL_TWEETS = "http://127.0.0.1:8000/twitter/tweets"
API_URL_NEWS = "http://127.0.0.1:8000/news/summarize"
```

Testing of the APIs has been done with the help of pytest and Fast APIs Test Client.

Models

Hugging Face’s T5-Base model is used for each summarization Task.

Database

PostgreSQL has been used as a database connected with the help of psycop2g library.

Web-App and Feeback Loop

Flask Application

Flask is used to create the web-application. It shows the news along with all the summaries of the news. It shows along with all the metadata and asks for a user input to check if a given tweet is relevant to the news or not. Which will later trains a model and same model will predict the relevant tweet to show for users.
SKlearn library is used to create and train a model.

DVC

DVC is used to track the data and pipeline. It tracks and can run the whole pipeline along with model training with a single command also showing the model evaluation matrices.

Docker

The web-page has been dockerized and can access the data with the database.

Prometheus and Grafana

Prometheus and grafana have been used to track the usage of the CPU and the network. They both got their own docker containers.

Here are the ports for each of the above:
Prometheus: localhost:9090
Grafana : localhost:3000
Postgres : localhost:5432
Flask : localhost:5000
Node_exporter : localhost:9100