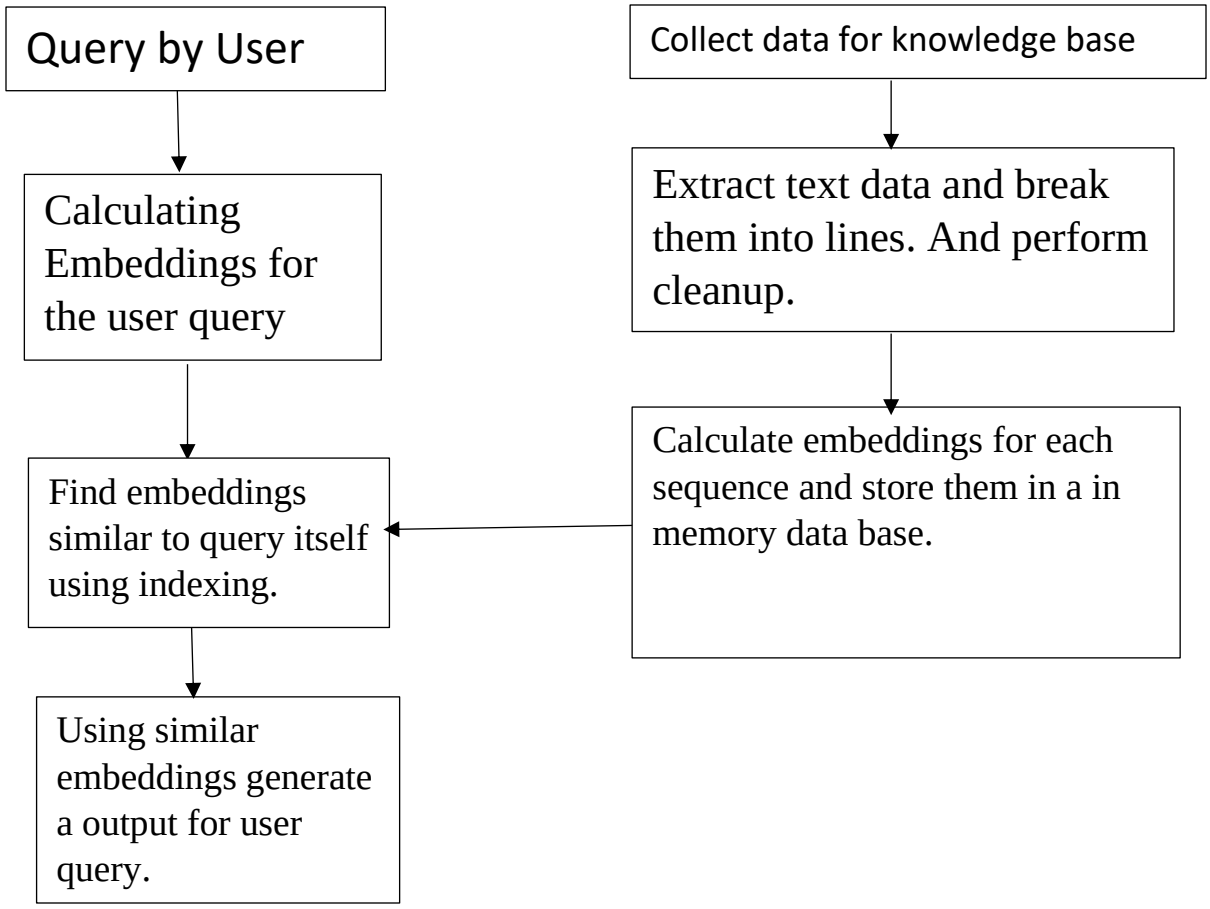
# Design Document:

Implementation diagram:



# Though Process:

- RAG is a new system that is leveraging power of generative LLMs and search methods.
- Search mechanism:
    - For the implementation of RAG Vector data bases are very popular choice. Here the vector representation of data is stored in database and then different mathematical algorithms are used to calculate similarity. Few algorithms to name are “Cosine Similarity” and “Euclidian distance”. This are frequently used algorithms.
    - For implementation of in-memory Vector Database, FAISS library is used. This library is implemented in C++ and wrappers are implemented for python use. FAISS is primarily implemented for FAIR and one of the key importance is, important algorithms are also implemented with GPU optimization.

- LLM Used:
    - For implementing RAG two important steps are required i.e., Retrieval and Generation.
    - Retrieval is a method of extracting relevant knowledge from source. For implementing knowledge base embeddings are created. These embeddings are created using NLU part of transformers. Idea is to squeeze the sequence data into higher dimension. These vectors are then stored in vector databases.
    - Generation is a method of leveraging full power of transformers, where the context is provided to the model, and it generates a output. To make this output more relevant the context is created using the relevant information using retrieval process.

- Choosing Single model for retrieval and generation:
    - Using same base model for both retrieval and generation ensures consistency. The query and source embeddings (vector representations) are aligned, making the comparison meaningful.
    - By combining retrieval mechanisms with generative capabilities, the model can adapt to various scenarios based on the external data it accesses.
    - RAG filters available data to the necessary minimum, speeding up response time and reducing inference cost.

- Choosing LLM model:
    - For the task specific “ashakthy/biology” model is used.
    - This model is used as it is finetuned for biology task and gpt2 is used as base model for fine tuning.
    - After some trials with different models, a fine-tuned model is performing much better. And given the computational limit, this was the best possible option.

Future Works:

- Implementation of data ingestion pipeline to continuously add new data in the index.
- Fine tunning a LLM model to further improve the quality of system.
- Implementation of Test-driven development for better coverage of code development and integration.
- Implementation of Code in more modularize way to implement microservices.
- Implementation of better UI with streamlit.