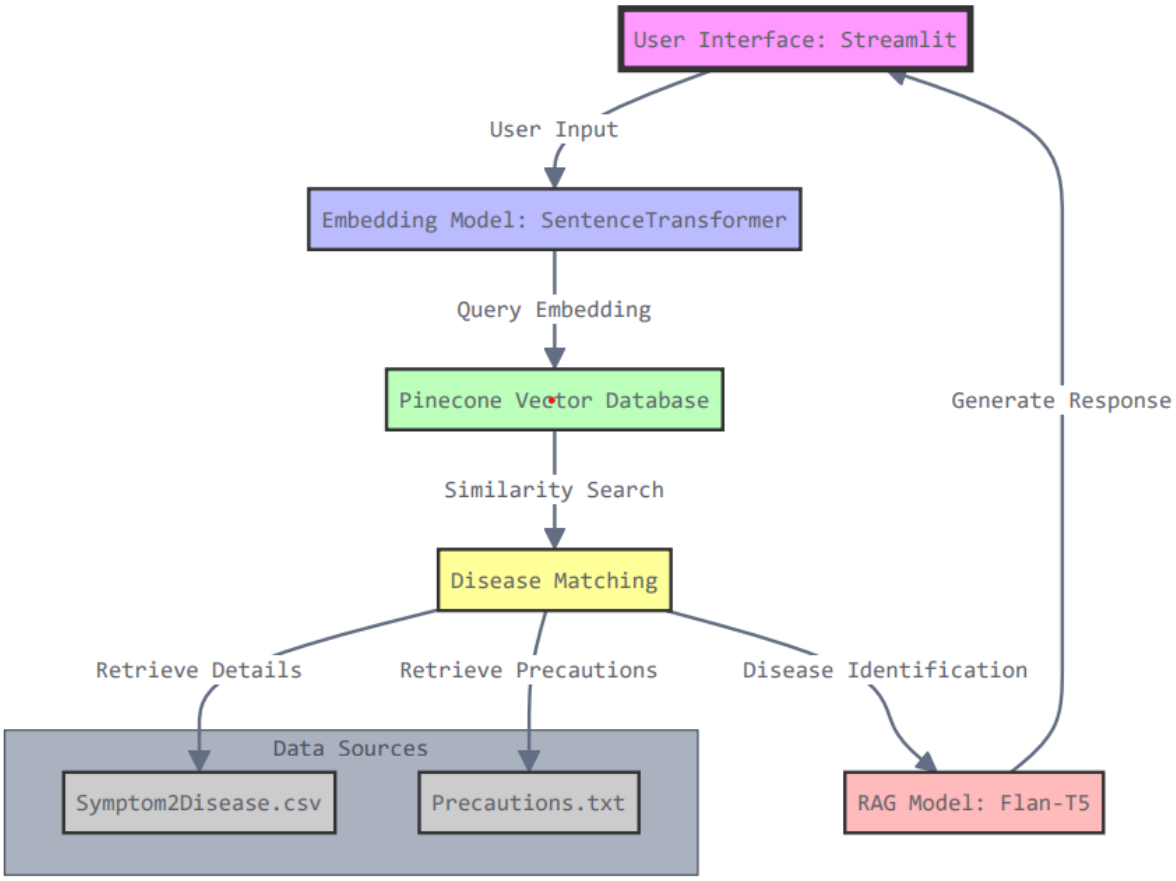
Name – Vipul Viresh Patil (002833236)

Osborne Victor Lopes (002642712)

System Architecture Diagram



Implementation Details

- **Initialization**:
    - Pinecone client and index are initialized or reused.
    - SentenceTransformer (all-MiniLM-L6-v2) generates query embeddings.
    - The flan-t5-base model generates conversational responses.
    - Disease data is loaded from a CSV file.
    - Precautions are parsed and mapped to diseases from a text file.
- **Chatbot Functionality**:
    - The bot accepts queries through a Streamlit interface.
    - It identifies if the query asks for precautions or general symptoms.
    - For symptom-related queries:
        - It generates embeddings for the user input.
        - Queries Pinecone for the closest matching disease.
        - Returns the result with a confidence score and an explanation.
    - For precautions:
        - Matches the disease name in the query to the precautions list.
- **Session Handling**:
    - Maintains conversation history using st.session_state.

Performance Metrics

- Accuracy of Matching:
    - Depends on embedding quality (384-dimensional vectors).
    - Evaluated by the confidence score from Pinecone results.
- Responsiveness:
    - Speed is influenced by Pinecone query latency and model inference times.
- Error Handling:
    - Extensive try-except blocks ensure the system gracefully handles initialization and runtime errors.

Challenges and Solutions

- **Challenge**: Handling missing or ambiguous data.
    - **Solution**: Default responses are used when no matching disease or precaution is found.
- **Challenge**: High query latency for real-time interaction.
    - **Solution**: Used efficient pre-trained models and serverless Pinecone for optimized performance.
- **Challenge**: Precaution text structure.
    - **Solution**: Implemented a parser to handle disease-specific precaution lists.

Future Improvements

- **Data Expansion**:
    - Enrich the symptom-disease and precautions datasets with additional medical conditions.
- **Interactive Feedback**:
    - Allow users to correct or refine results to improve system learning.
- **Model Enhancements**:
    - Fine-tune the RAG model for domain-specific conversations.
- **UI Improvements**:
    - Introduce visual explanations for disease-confidence scores.
- **Integration**:
    - Include an API for external access to the chatbot capabilities.