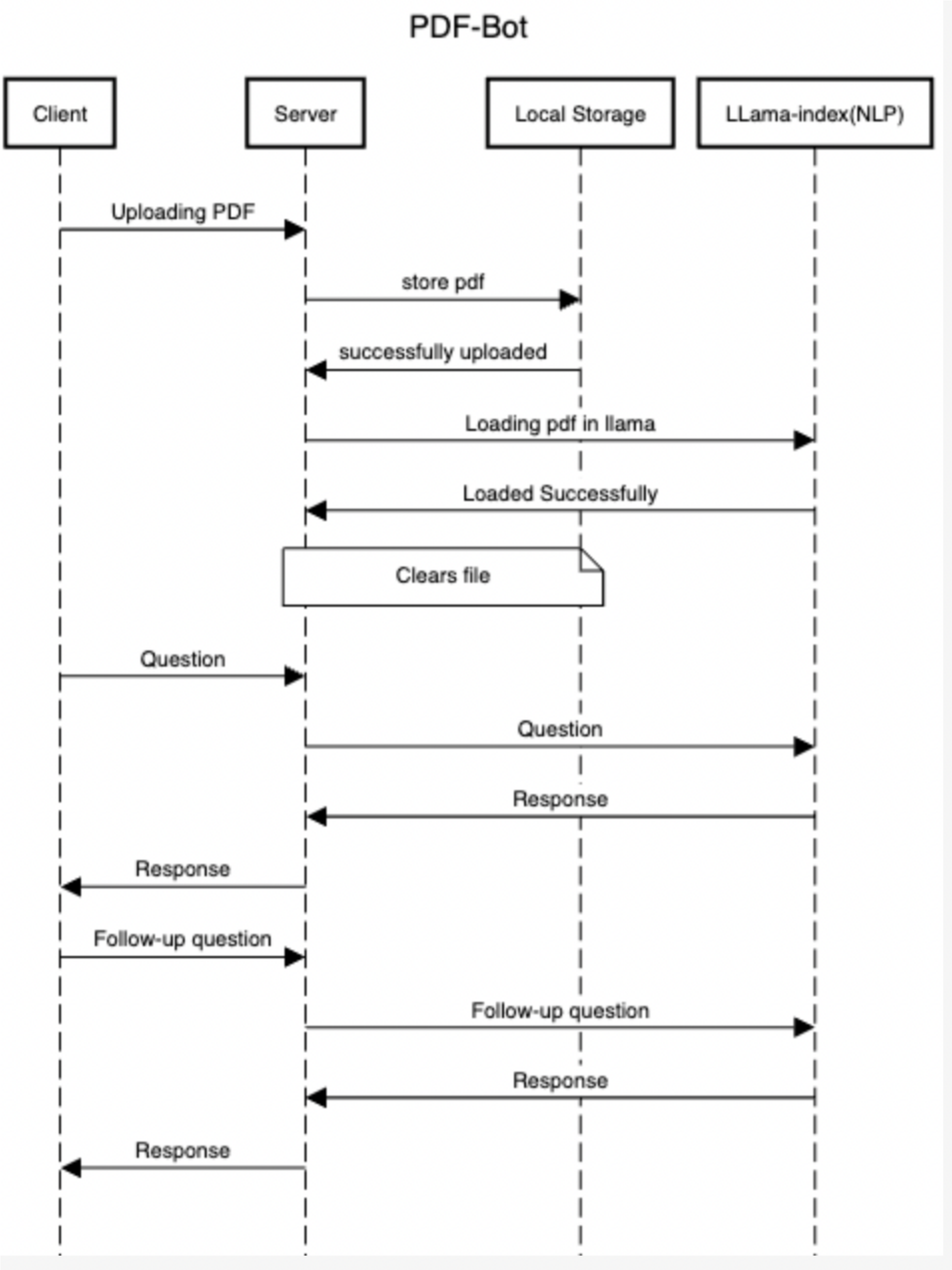
PDF-Bot
Client
Server
Local Storage
LLama-index(NLP)
Uploading PDF
store pdf
successfully uploaded
Loading pdf in llama
Loaded Successfully
Clears file
Question
Question
Response
Response
Follow-up question
Follow-up question
Response
Response

## Architecture:

- **Frontend:** React app handling user interactions. Utilizes Axios for HTTP communication.
- **Backend:** FastAPI app providing API endpoints for uploading PDFs and processing chat requests. Utilizes FastAPI, Pydantic, and uvicorn for server setup and routing. Interacts with the LlamaManager class for document processing and chat functionality.
- **Core Functionality (LlamaManager class):**
    - Manages document processing and chat engine.
    - **Document Processing:**
        - Loads PDFs from a designated directory.
        -
        - Creates a vector store index from the documents for efficient searching.
    - **Chat Functionality:**
        - Creates a chat engine based on the document index.
        - Utilizes the chat engine to retrieve relevant information from the index to answer user queries.

## Data Flow:

1. User uploads PDF via React, triggering an Axios request.
2. Backend receives the request, saves the PDF, and updates the document index using LlamaManager.
3. User sends a chat query via React, initiating another Axios request.
4. Backend processes the query using the chat engine within LlamaManager, utilizing the document index.
5. Backend responds with the result to React.
6. React updates the chat interface with the received response.