SCALABLE LLM BASED CHAT SYSTEM

1. **Android App (Frontend)**
    - **User Interface**: Users type chat messages and see responses in real-time.
    - **OkHttp Client**: Handles HTTP requests to the ChatGPT API or Firebase Cloud Functions.
2. **OkHttp Client**
    - **Function**: Sends user messages to ChatGPT API or Firebase Cloud Functions, depending on whether additional logic is needed.
    - **Response Handling**: Receives the response and updates the UI with the chatbot's reply.
3. **Firebase Cloud Functions**
    - **Function**: Acts as a gateway between the Android app and ChatGPT API.
        - Can handle **preprocessing** (e.g., adding metadata, API key management).
        - Can forward the request to the **ChatGPT API**.
    - **Caching**: Optionally caches responses or manages rate limits.
4. **ChatGPT API**
    - **Function**: Processes the chat message sent from the user and generates a response using the language model.
    - **Returns**: Sends the processed response back to the OkHttp client in the Android app.
5. **Firebase Firestore/Realtime Database**
    - **Function**: Stores user information, chat histories, and app settings.
    - **User Data**: Managed by Firebase Authentication (for logging in and user management).
    - **Chat History**: Optional, but can be stored for later retrieval by the user.

Data Flow

1. **User Interaction**: A user sends a message via the Android app.
2. **OkHttp Request**: The app uses OkHttp to send the message as a POST request to the ChatGPT API (or via Firebase Cloud Functions).
3. **ChatGPT Processing**: The ChatGPT API processes the request and generates a response.
4. **Response Handling**: OkHttp receives the response and updates the UI.
5. **Data Storage**: Firebase Firestore can optionally store chat history for later use.

System Architecture