# API Contracts: RAGent - Multi-Agentic Conversational AI System

## 1. POST /chat

Description: Accepts user message, retrieves relevant RAG context, and returns LLM response.

Input Format (JSON):

```
{
  "user_id": "string",
  "message": "string",
  "history": [{"role": "user", "message": "previous message"}]
}
```

Response Format (JSON):

```
{
  "user_id": "string",
  "message": "string",
  "rag_context": "string",
  "response": "string"
}
```

## 2. POST /upload_docs

Description: Upload internal documents (PDF/TXT/CSV/JSON) to populate the RAG vector store.

Input: Multipart/form-data (List of files)

Response:

```
{
  "message": "string",
  "files": ["filename1.csv", "filename2.txt"]
}
```

## 3. POST /crm/create_user

Description: Creates a new user profile in the CRM.

Input Format (JSON):

```
{
  "name": "string",
  "email": "string",
  "company": "string"
}
```

Response:

```
{
  "user_id": "string",
  "message": "User created successfully."
}
```

### 4. PUT /crm/update_user

Description: Updates user details by user ID.

Input Format (JSON):

```
{
  "user_id": "string",
  "name": "string (optional)",
  "email": "string (optional)",
  "company": "string (optional)"
}
```

Response:

```
{
  "message": "User updated successfully."
}
```

# API Contracts: RAGent - Multi-Agentic Conversational AI System

## 5. GET /crm/conversations/{user_id}

Description: Retrieves the full conversation history for a given user.

Response:

```
[
  {
    "role": "user",
    "message": "string",
    "topic": "string",
    "status": "string",
    "timestamp": "ISO format"
  },
  ...
]
```

## 6. POST /reset

Description: Clears memory (e.g., conversation history) for a given user.

Input Format (JSON):

```
{
  "user_id": "string"
}
```

Response:

```
{
  "message": "Conversation memory cleared."
}
```