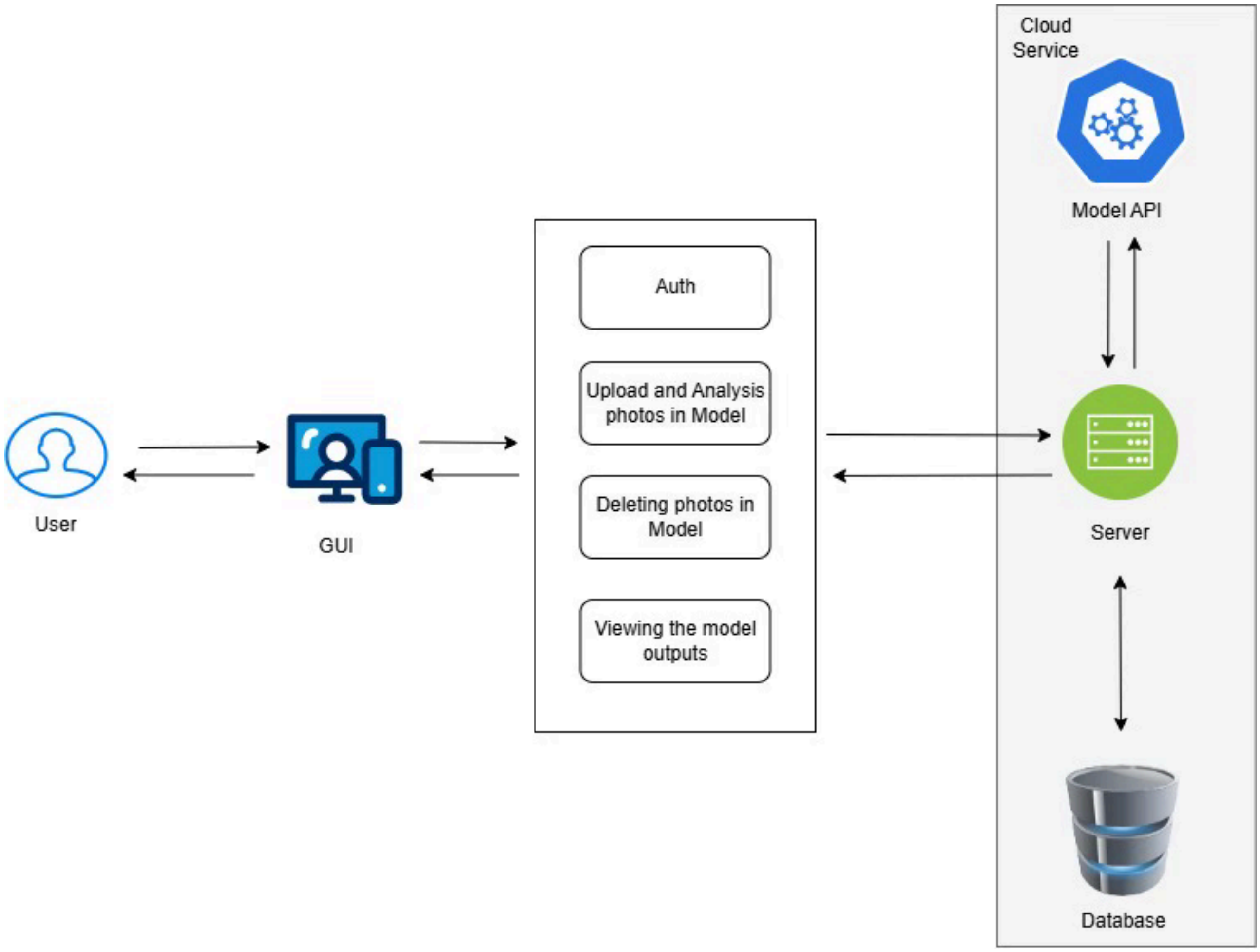
Updated Detailed Design

High-Level Architecture Overview



The system consists of four main components to ensure modularity and scalability:

1. **User Interface (UI):** A web-based interface for researchers to manage image uploads, analyses, and results.

2. **Backend Server:** Acts as the central hub to manage requests between the UI, the database, and the machine learning model API.

3. **Machine Learning Model Service:** Hosted as a cloud service, this component performs image analysis to extract insights on microglial cell morphology and mitochondrial membrane potential.

4. **Database:** Stores user data, images, and analysis results for efficient retrieval and management.

Detailed Component Design

1. User Interface (UI):

Purpose:
Provide an intuitive interface for users to interact with the system, manage their data, and visualize results.

Technologies:

- React.js for frontend development.

Key Functionalities:

1. **User Authentication:**
   - Registration, login, and password recovery for users.
2. **Image Management:**
   - Upload images (individually or in batches) for analysis.
   - View and delete previously uploaded images and their associated analyses.
3. **Analysis Visualization:**
   - Display analysis results in both graphical (e.g., heatmaps, charts) and textual formats.
   - Enable exporting results in CSV or PDF formats.
4. **Historical Data Access:**
   - Browse, search, and filter past analyses.

2. Backend Server:

Purpose:
Manage communication between the UI, ML model service, and database.

Technologies:

- Node.js with Express.js.

Responsibilities:

1. **User Management:**
   - Handle user authentication and authorization securely.
2. **Image Handling:**
   - Validate and store uploaded images.
   - Forward images to the ML model API for analysis.
3. **Analysis Management:**
   - Receive analysis results from the ML service.
   - Store results in the database and serve them to the UI.
4. **Data Security:**
   - Encrypt sensitive information and ensure HTTPS communication.

3. Machine Learning Model Service:

Purpose:
Perform detailed image analysis to provide insights into mitochondrial activity and microglial cell morphology.

Technologies:

- Python with TensorFlow or PyTorch.
- RESTful API for backend integration.

Key Functionalities:

1. **Image Preprocessing:**
   - Normalize pixel values and adjust image dimensions for model input.
   - Reduce noise to enhance image quality.
2. **Analysis:**
   - Evaluate microglial morphology and mitochondrial membrane potential.
   - Return numerical and textual insights along with graphical data.
3. **API Endpoints:**
   - `/analyze`: Accepts images and returns analysis results.
   - `/status`: Provides operational status of the service.

4. Database:

Purpose:
Store and manage all user data, image metadata, and analysis results.

Technologies:

- MongoDB (NoSQL database).

Stored Data:

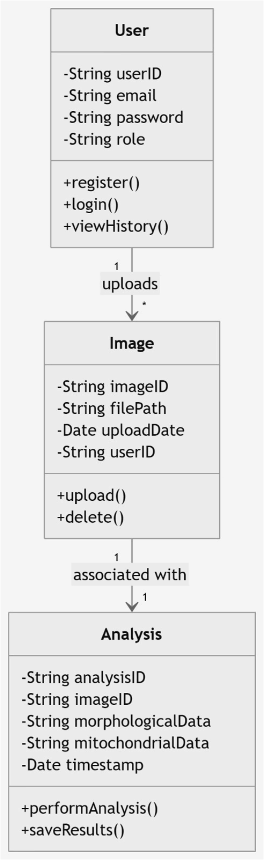
1. **User Data:** Authentication details and roles.
2. **Image Metadata:** File paths, upload dates, and user associations.
3. **Analysis Results:** Morphological data, mitochondrial membrane potential, and timestamps.

Design Considerations:

- Allow users to delete images and associated analysis results.
- Implement backup mechanisms for data recovery.

Communication Flow

1. **Image Upload:**
   - The user uploads images through the UI.
   - The backend validates and forwards images to the ML service.
2. **Image Analysis:**
   - The ML service processes the images and returns detailed analysis results.
3. **Database Storage:**
   - The backend stores analysis results in the database.
4. **Result Visualization:**
   - The UI retrieves results and displays them in graphical and textual formats for the user.



Relationships:

- A `User` can upload multiple `Image` objects.
- Each `Image` is associated with an `Analysis` object.

Technology Stack

- **Frontend:** React.js, Chart.js for visualizations.
- **Backend:** Node.js with Express.js.
- **Machine Learning:** Python (TensorFlow or PyTorch).
- **Database:** MongoDB.
- **Hosting:** AWS (S3 for image storage, EC2 for backend and ML service).

Considerations for Future Expansion

1. Add support for more advanced visualization tools and interactivity.
2. Incorporate multi-language support in the UI for global researchers.
3. Enable seamless integration with additional datasets or analytical tools.

This updated detailed design reflects the architecture shown in the provided diagram, aligning the system's components and functionality with project goals. Let me know if further refinements are needed!