Data Upload Module:

- Description: In this module, the endpoint will allow the user to upload images for their entire dataset. This is assuming that the user will have the same image type and image name corresponding to the images and label.
- Endpoint: '/upload'
- Method: POST
- Request parameters:
    - Images: A folder containing image files, with each filename is just a number. Ex: the first image will be named "1.png", then the second image will be "2.png" and so on. This would be their image ID.
    - Labels: A folder containing labels for the corresponding image files. The way the folder should work is to have the exact images in the images folder, but their name would contain the image ID number, followed by its label name and separated by a specific character.
- Response:
    - '201 OK': Upload successful.
    - '401 Bad Request': If there are any issues with the requests.

Example request:

```
upload_data = {
   "images": ["1.png", "2.png"],
   "labels": ["1 : Lionel Messi", "2 : LeBron James"]
}

response = requests.post("http://api-url/upload")
```

Training Data Module:

- Description: In this module, the endpoint will start the process for the image classification model training using the uploaded data. It should split the uploaded dataset into training and testing data. After running the training configurations, it will return a data container storing the model's parameters (weights and biases) that have been tuned, then it will apply those parameters onto the test dataset and calculate the statistics for it (error/loss, accuracy, etc.)
- Endpoint: '/train'
- Method: POST
- Request parameters:

    - Train-Test Split: Specify the percentages for training and testing data split. Optional: Validation split.
    - Regularization term (optional)
    - Loss function

- Example of usage:
  response = requests.post("http://api-url/train")

User Authentication Module

- Description: This module manages user authentication, ensuring secure access to the system's functionalities. Users are required to provide valid credentials, such as a username and password, to authenticate themselves before accessing protected resources.
- Endpoint:
    - {BASE_URL}/register
    - {BASE_URL}/login
- Method: POST

- Request Parameters:
    - Username: The username provided by the user.
    - Password: The password associated with the provided username.
- Response:
    - 200 OK: Authentication successful, providing an authentication token.
    - 401 Unauthorized: Authentication failed due to invalid credentials.

Database

- Description: This module manages interactions with the database, including storing, retrieving, and updating data. It uses SQLite3, and has a table for images, labels, models, training data, and users. Each table have unique IDs that can help associate with one another.

Inference Module

- Description: This module enables the inference process using trained models. It provides endpoints for sending input data to the deployed models and receiving predictions or inference results.
- Endpoint: {BASE_URL}/inference
- Method: POST
- Request Parameters:
    - Input data: The data to be processed by the model for inference.
- Response:
    - 200 OK: Successfully inference data.

    - 401 ERROR: Unsuccessfully inference data.

Upload Models Module

- Description: This module facilitates the upload of pre-trained machine learning or deep learning models to the system. It provides an endpoint for uploading model files along with necessary metadata such as model name, version, and description. The user will have to upload their own machine learning model in order to use it for training, testing and inference.
- Endpoint: {BASE_URL}/upload_model
- Method: POST
- Request parameters:
    - Model file: The file containing the pre-trained model weights and architecture. This will be in .h5 as we are using the tensorflow neural network library.
    - Parameters of model such as hyperparameters, optimizer, loss function, etc.
- Response:
    - 201 OK: Model upload successful.
    - 401 Bad Request: If there are any issues with the request or uploaded model.