Annotation Tool Backbone

Overview

The project is a web-based data annotation system that allows users to select multiple images and annotate them using a polygon-shaped bounding box annotation tool. The system also provides the option to add labels to the annotated objects in the images or videos.

To access the annotation page, users need to log in using the user management system. The login process is integrated with the annotation user management team. The system also allows users to pick images that have been assigned to them by the project management team.

The system provides an annotation tool API that allows users to update task progress and track the progress of tasks assigned to them. The progress status of each image under annotation is displayed on the system.

The system also allows users to upload images to the database and toolbox. The system is built using the React framework and Konva library, which provide the necessary tools for polygon-shaped bounding box annotation in images and videos.

Users can resume annotation from the last saved stage, and they can download annotated images and save them to the database for future reference.

The images can be passed through an AI/CV pipeline for pre processing before annotation can be started. Overall, the project aims to provide a user-friendly and efficient data annotation system for use in various applications.

Recommended Technologies

React + Node.js + Express + MongoDB example Overview

The back-end server will use Node.js + Express for REST APIs.
The front-end side will be a React.js client with React Router, Axios & Bootstrap.
We will use MongoDB for our database.
We will store images in *TBD*

Use cases

1. Data Annotation ToolBox Setup.
2. Data Annotation BackBone Setup.

Technical Requirements

1. **User Management:**
    a. The system shall provide a user login process to allow users to access the annotation page.
    b. Integrate this login with the help of Annotation User Management Team.

2. **Task Management:**
    a. The system should provide user to pick the images that have been assigned to a particular annotator to annotate.
    b. Integrate this feature with the help of Project Management Team.

3. **Progress Tracking:**
    a. The system shall allow users to update task progress using an annotation tool API.
    b. The system shall enable users to track the progress of tasks assigned to the users.
    c. The system shall display the progress status of each image under annotation.

4. **Annotation ToolBox:**
    a. Provide user an option to upload image in the database and toolbox.
    b. The system should provide a user draw polygon shaped bounding box annotation in images or videos using React and Konva.
    c. The user should be able to add a label to the annotated object in the image/video.

5. **Resume/Download Annotation Image:**
    a. The user should be able to resume the annotation from the last saved stage.
    b. The user should be able to download the annotated image and save it to database for future reference.

6. **AI/CV Pipeline:**
    a. The system should provide a feature to pre process the image using AI/CV pipeline.
    b. Integrate this feature with the help of AI/CV pipeline team.

Non functional requirements

1. **Security**: The system should ensure that sensitive information is kept secure and only accessible by authorized users.

2. **Performance**: The system should be able to handle a large number of concurrent users without degrading performance.

3. **Usability**: The system should be easy to use, with a clear and intuitive user interface.

4. **Scalability**: The system should be scalable to accommodate future growth in the number of users and tasks.

5. **Reliability**: The system should be reliable, with a high uptime and minimal downtime.

Database Design - MongoDBschema

User Collection

Type	Key	Value	Description
Primary Key	User_id	String	A unique user Id of each user
Foreign Key	Image_Ids	List of String	An array of images a user is annotating / uploaded.
	Task_id	String	Id of the Task.

Image Collection

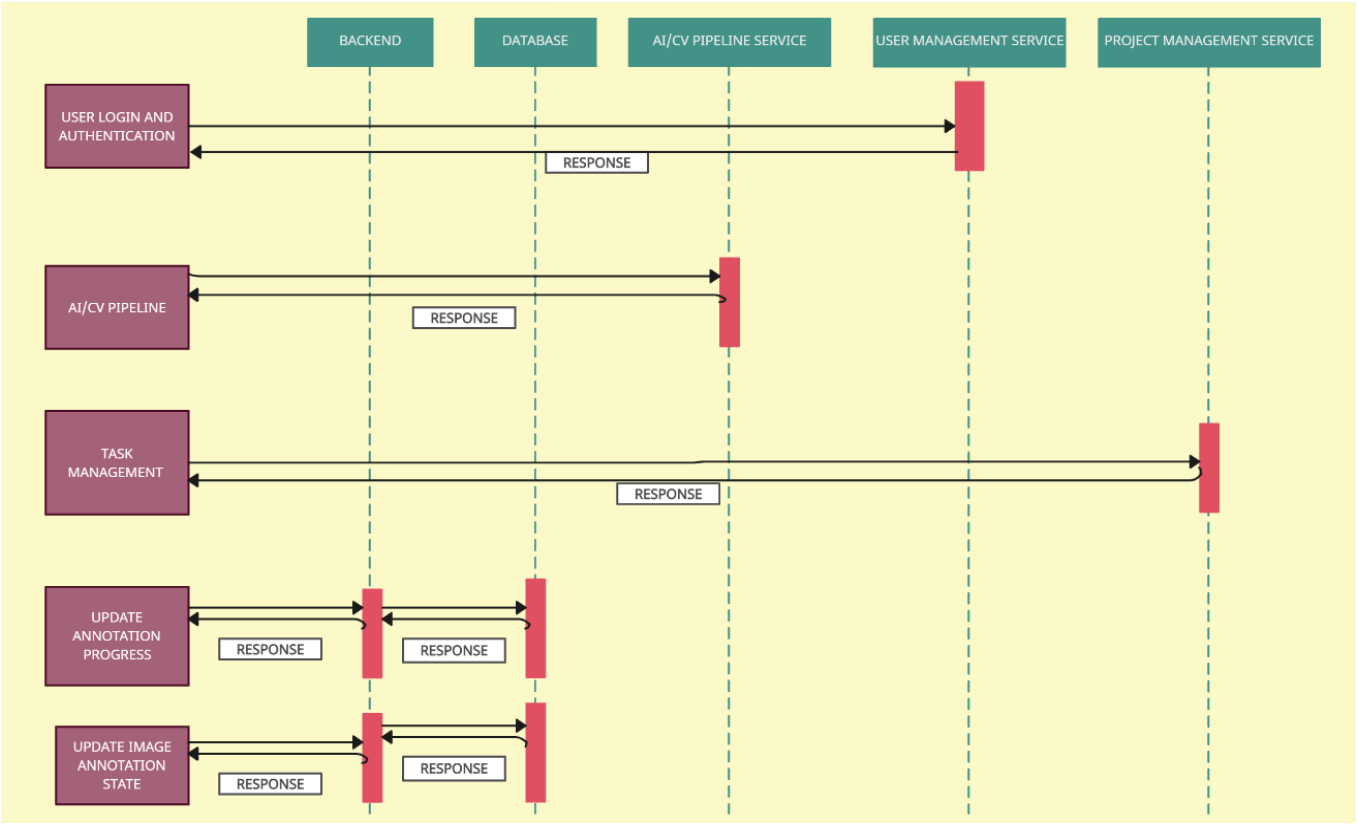
Type	Key	Value	Description
Primary Key	Image_id	String	A unique image id for each of the image uploaded to database.
	Image_url	String	The original url of the image
	Preprocessed_image_url	String	The image url after being pre processed using ai/cv pipeline
	Annotation_image_url	String	The annotated image url of the image
	Status	String	The status of the image annotation progress.

	Bounding_polygons	List of Bounding Polygons Objects	This array will store the state of annotation image, like the bounding polygons on the image and its label.

Bounding polygons Collection

Type	Key	Value	Description
Primary Key	Object_id	String	A unique id for each of the bounding polygon
Foreign Key	Image_Id	String	Image id to link the bounding polygon to the image
	Coordinates	Array of coordinates	Array to store the coordinates of the bounding polygon
	Label	String	Label for the annotation bounding box
	Colour	String	Color code for the border of the polygon.

UML diagram



Workflow

- First a user has to log in to access the annotation tool.
- In the home screen of the annotation tool there would be a dashboard for a user to view the images under annotation.
- In the home screen of the annotation tool there would be a toolbox to start annotation on the selected image.
- In the home screen of the annotation tool there would be a button box to mark the status of the image.
- User can upload the image to store it in database for annotation.
- USer has an option to pre process the image using ai/cv pipeline.
- User can resume the annotation anytime in future.
- User can download the annotated image.
- User can see the Task details of task assigned to it and the assigned images to be annotated.