1. **Problem statement: Design a system for vms(video management system) for an organization (Camera Permission Management) Microservices**

I. Structure:

The above system should have three level of hierarchy

- Super admin (can edit view any camera)
- Branch admin (can add / edit camera but cannot delete camera added by superadmin. Can view camera assigned by superadmin)
- Supervisor (just view cameras assigned by superadmin / branch admin)

II. Requirements:

- Super admin can view, add, remove, modify camera/branch admin/supervisor.
- Branch admin can view, modify cameras (also can edit cameras of supervisor).
- The supervisor can view the camera.
- All the authentication/authorization should be performed using jwt tokens(login as well as refresh tokens).
- Users should be informed with proper error messages in case of any error.
- All the operations should be performed using apis/events.
- The whole application should be containerized (container/service).
- The data should be stored in a database of any choice(structured/non structured).
- Filter sources based on location/ recent activity.
- Logs should be there for each change made by any user.
- There should be api for user activity view accessible to only superadmin (with token)
- Code should be properly documented.

III. Software requisites:

- Fastapi framework for api management
- Any supportive python library for jwt tokens management
- Any supportive python library for database management
- Docker/Kubernetes

Any other required library

User Management:

- User Registration
- User Permission Management

Camera Management:

- Camera Add
- Camera Delete
- Camera Edit
- Camera View

User Authentication:

- User Login
- User Logout
- User Refresh Token (in 10 mins)

Redis:

- Store User Encrypted Token to be accessible after for User Management & Camera Management.

User Management Database can be structured based on requirements.

Camera Management Fields:

IP
Location
SourceName
Added By
Edited By
Added On
Updated On

Minimal System Architecture Can Be modified as per requirement.: