Suggestions from review 2 :

## Research and literature review:

Conduct research and literature review to identify papers and studies related to video processing , entity selection and identification of potential areas of interest in the video.

Dataset preparation:

Review and refine the datasets to ensure completeness and accuracy.

Validate the dataset against our project requirements and objectives.

Some datasets:

* IMAGE NET DATASET: annotated video clips over wide range of categories.

COCONET DATASET: Includes videos with object notations

KITTI DATASET(AUTOMOBILES):  images collected from a moving vehicle equipped with cameras, along with annotations for various tasks such as object detection, tracking, and scene understanding.

# Methodical entity selection:

#### Selection criteria:

The selection criteria for entities [based on project requirements and objectives. Like the most commonly occurring entities or entities with an outline bigger than the specified threshold.]

Design constraints , assumptions and dependencies:

# Constraints:

Hardware constraints - CPU power and GPU availability for large videos for speed and efficiency.

Time constraints - Time availability for dataset collection, model creation and model refinement for better efficiency.

Budget constraints - Restricted access due to software licenses, pay to use infrastructure and services .

Scalability - Extending the models for larger data and catering to increasing user demands and requirements .

Domain expertise - Knowledge of the datasets , models and approaches to the problem .

# Assumptions:

**Video Quality and Resolution**: Assumption that the videos provided to the software will have sufficient quality and resolution for effective entity detection.

**Entity Localization Accuracy**: Assumption that the algorithm will accurately localize the detected entities within the video frames. This involves assuming that the bounding boxes or segmentation masks generated by the algorithm will tightly enclose the entities without including extraneous objects.

**Integration with Existing Workflows**: Assumption that the software will seamlessly integrate into existing video processing workflows and systems. This involves assuming that the software will provide APIs, SDKs, or other integration points for easy integration with third-party applications and platforms.

# Dependencies:

Video Processing Libraries: FFmpeg: for handling video input/output and basic video processing tasks.

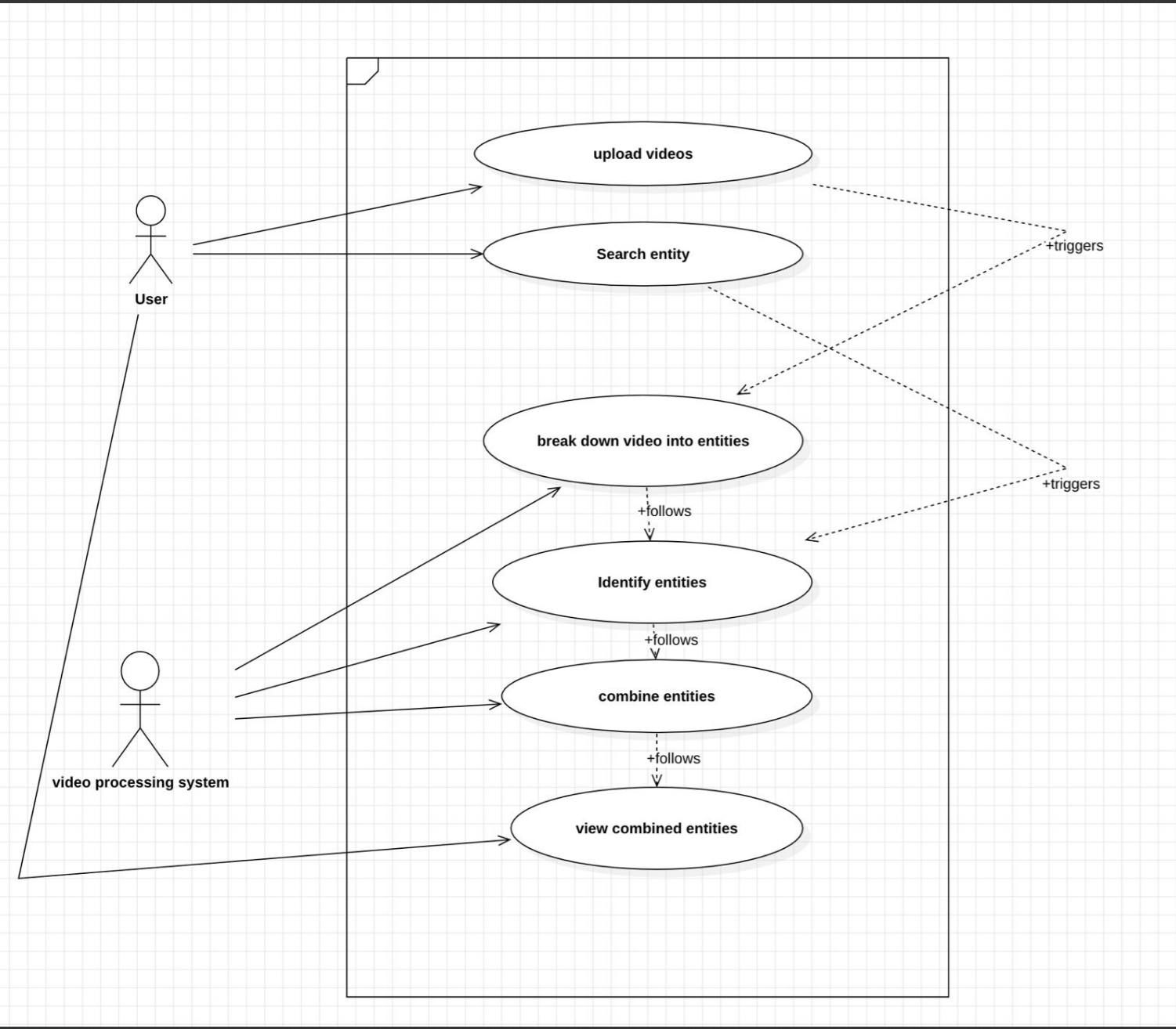
Classification Models: CNN (Convolutional Neural Network): VGG, ResNet, Inception, etc.

Transfer Learning Models: Pre-trained models like MobileNet, EfficientNet, or models from TensorFlow Hub.

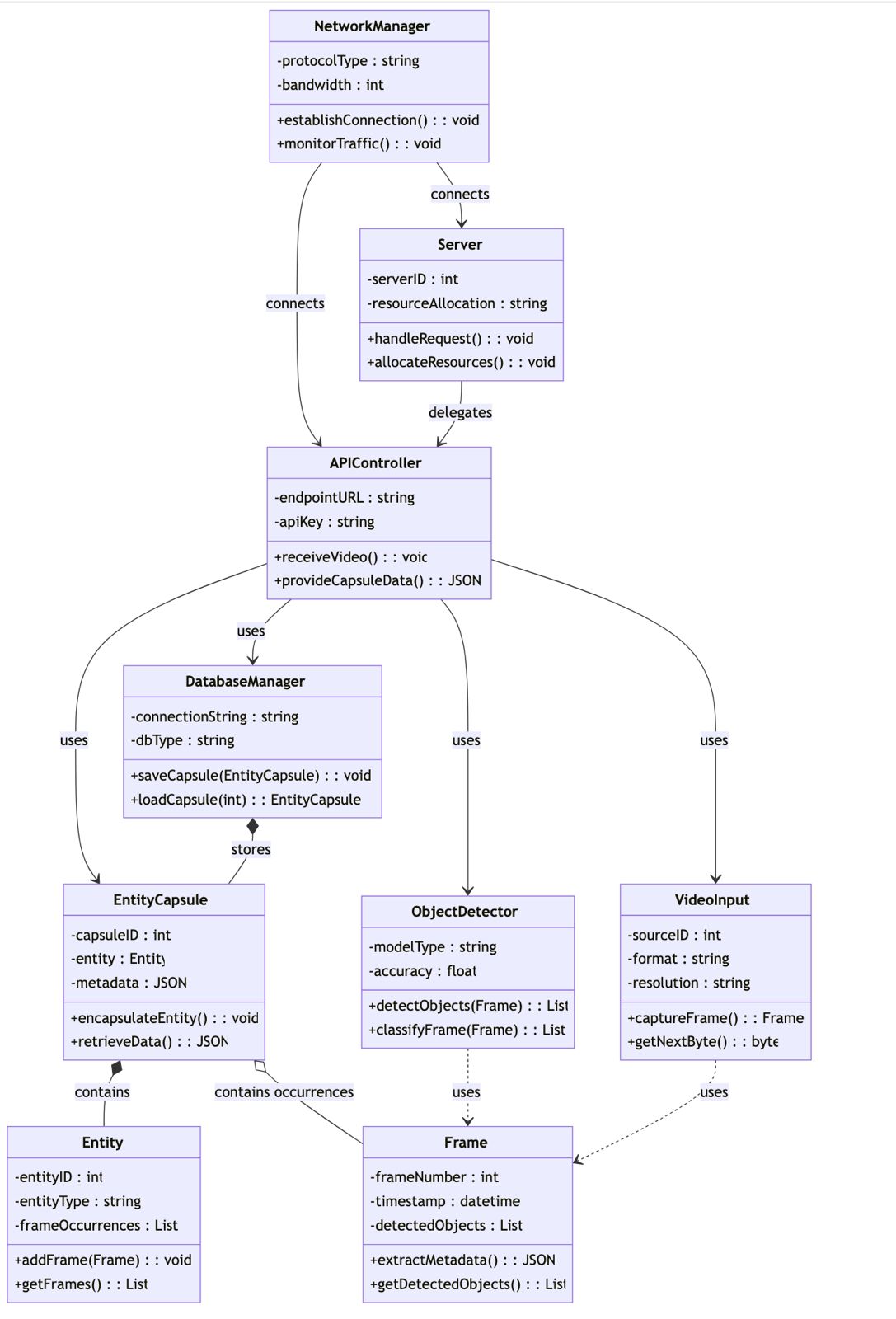
Computer Vision Libraries: OpenCV for image and video processing, including pre-processing and post-processing tasks.

Deep Learning Frameworks: TensorFlow, PyTorch, Keras (which can be built on top of TensorFlow or Theano)

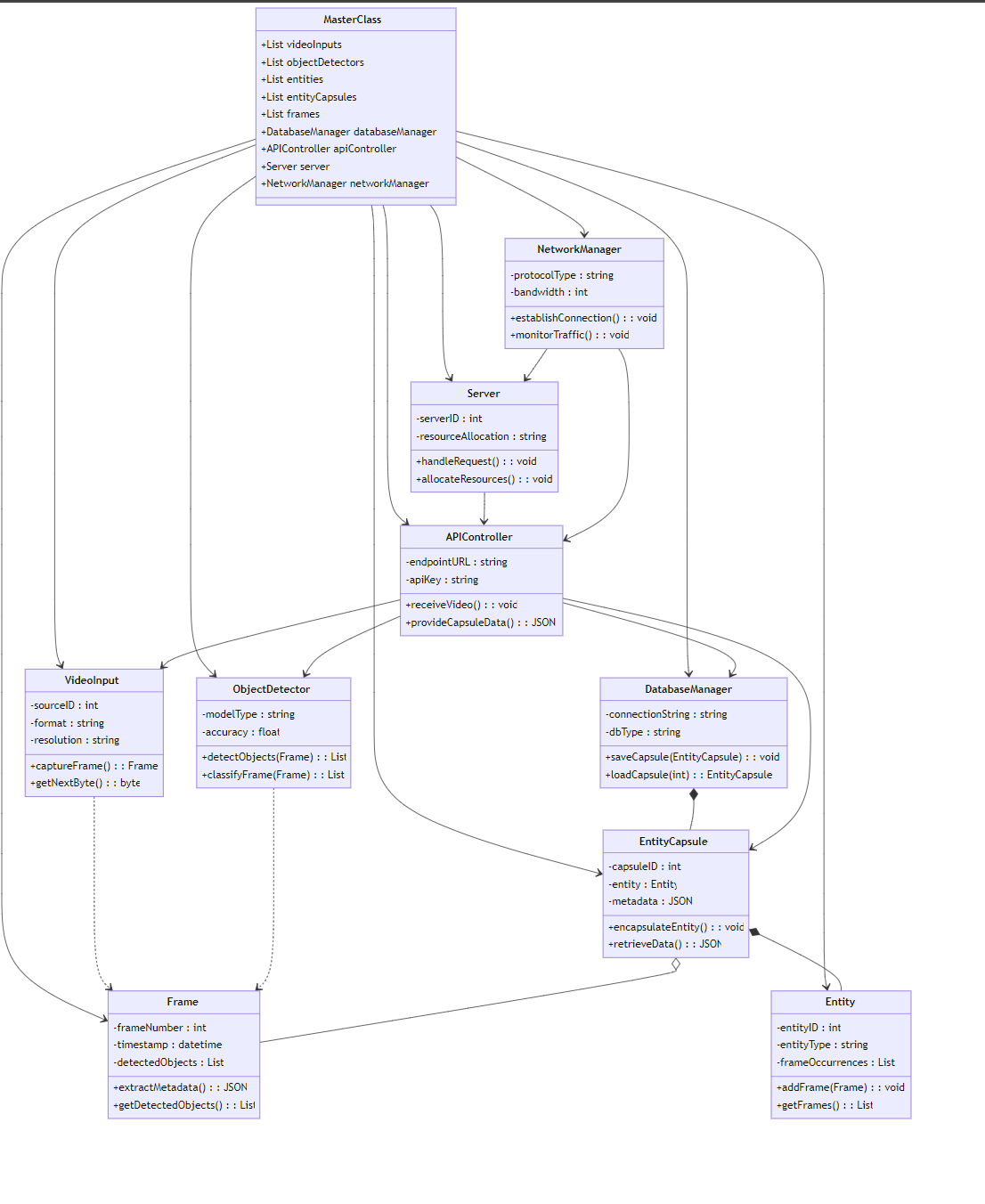
# USE CASE DIAGRAM



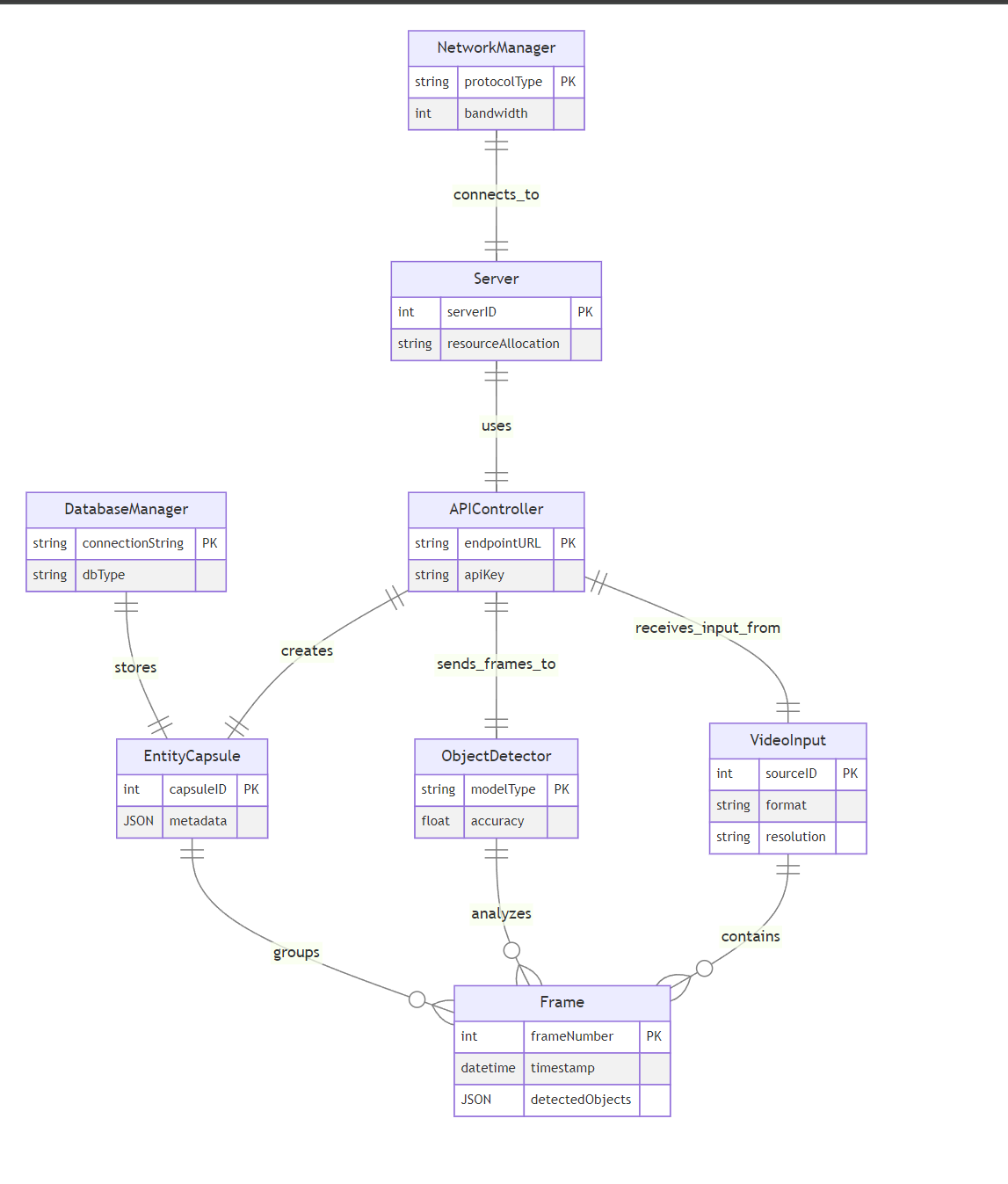
Class diagram:



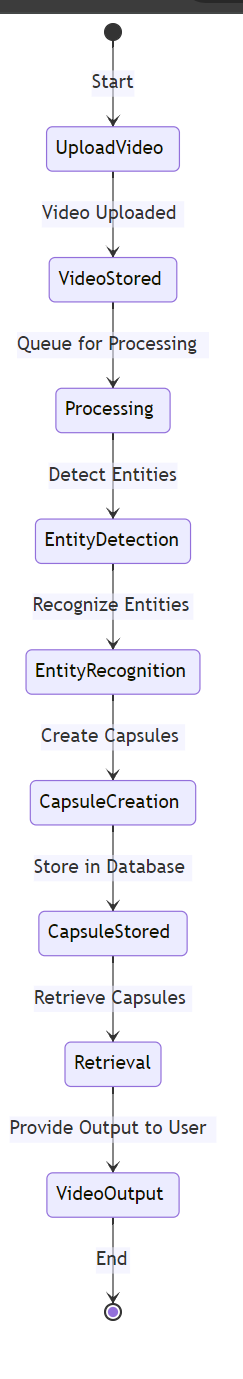
Master-class diagram:



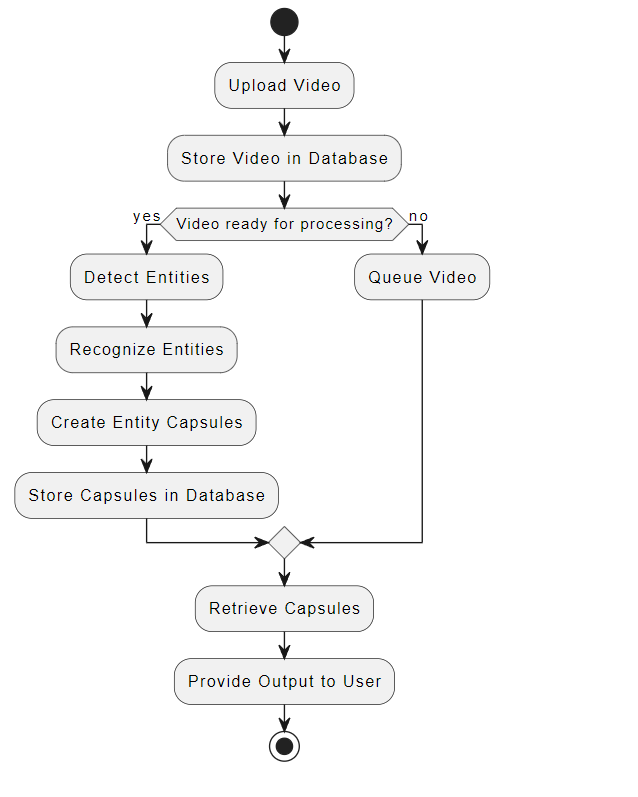
e-r diagram:



# STATE DIAGRAM



# Activity Diagram



# DEPLOYMENT DIAGRAM

