A small company dedicated to Software Development. Their engineering team, to which you belong, is working on writing a Web Application as an MVP for a new customer.

The code name for this App is “Loggy”, which is meant to offer functionality for a personal journal where users can log their daily activities through text, voice and video.

The first step will be to write the main functionality, which is essentially a Microblogging System where all the posts are automatically annotated with voice, video or text.

As an initial step, you must create the skeleton of the model for the core Microblogging System under these assumptions:

* Activity logs recorded only by one user.
* Each log is dated with a timestamp that is used as the key for displaying it in the feed
* Each log should have a name, a description and a date.
* Each log has attached the actual content, which can be plain text, an image, an audio file or a video file
* The audio and video files are supported in multiple formats as they are recorded in through the browser and uploaded to the server using WebRTC API.
* The images are also supported in multiple formats.

### Task

#### Main class

* Write a Main class that you will use as the simulator for testing you Classes

#### Abstraction (Classes, Objects and Interfaces)

* Write the class log that will be used as a base for the modeling. The class log is an abstraction of what logs can do: create, read, update, and delete. Logs also include specific characteristics like name, description and date. They also include, by default, a unique internal ID and a unique short code (in the form abc-abc-abc) is assigned.
* Implement some mockup output for the methods create, read, update, delete.
* Instances of the class log should be created assigning one, some or none of the attributes. Write constructors.

#### Encapsulation (Classes, Objects and Interfaces)

* The properties should only be accessible through member methods. Write methods for accessing the data.

#### Inheritance (Extends)

* As there may be different types of logs and every one of them have their own characteristics, extend the class log as TextLog, PhotoLog, AudioLog and VideoLog and assign some validation for the content such as size and type that corresponds to the log type and triggers an error if the size limit is reached or if an invalid type is submitted when these values are set.

#### Polymorphism (Extends)

* Text, Image and Video and Audio may have different formats. Override the method for validation accordingly.
* When saving the attachment, a special action like transcoding, automatic translation, automatic close captioning etc… can be triggered. Text for example can only be translated. Images can be annotated. Video and Audio can be transcoded and close-captioned. Implement in the super class the support for those actions and extend the derivate classes for supporting one of them.

#### Composition (HAS\_A)

* Generalize the implementation of the actions using an Interface and concrete classes per feature, which should be incorporated into the sub classes in a composition.