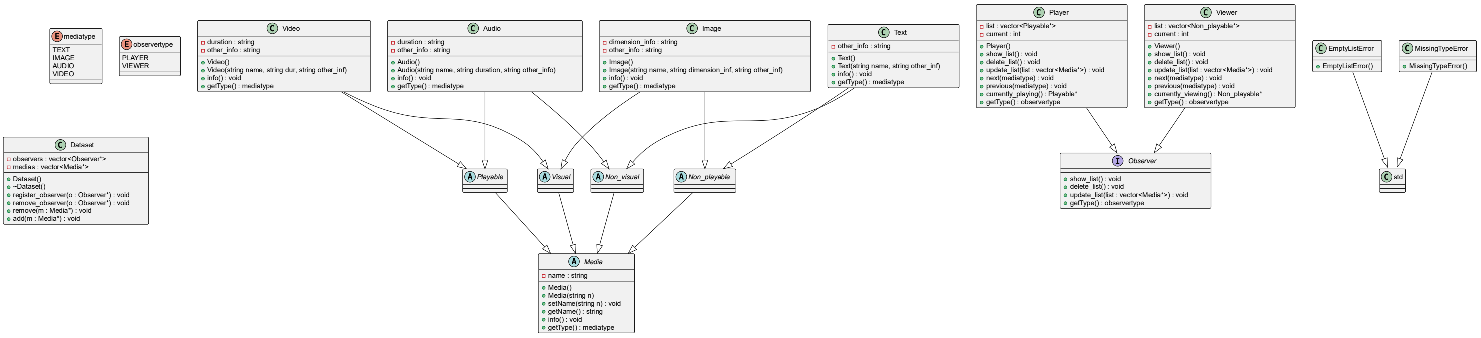
CSE241 2025 Spring Programming Assignment 6 Report

UML Diagram



This UML class diagram illustrates the design of a multimedia management system that handles various types of media content and supports observer-based interaction.

At the core of the design is the abstract class **Media**, which defines the basic interface for all media objects, including a name and two pure virtual functions: info() and getType(). Media objects are further categorized by their capabilities:

- **Playable** and **Non_playable** determine whether a media item can be played (e.g., audio, video) or not (e.g., text, image).
- **Visual** and **Non_visual** indicate whether a media item has a visual component.

Concrete media types inherit from these categories:

- **Audio** and **Video** are both Playable, with Video also being Visual.
- **Image** and **Text** are Non_playable, with Image being Visual.

The system also implements the **Observer pattern**, where the abstract **Observer** interface defines how external entities (such as players or viewers) can interact with the dataset.

- A **Player** manages a list of Playable media and allows navigation (next/previous) through the items.
- A **Viewer** manages Non_playable media and supports similar navigation.

The **Dataset** class holds a collection of media and a list of registered observers. It provides functionality to add and remove media while ensuring all observers are updated accordingly.

To handle edge cases gracefully, two custom exceptions are defined:

- **EmptyListError** is thrown when an operation is attempted on an empty list.
- **MissingTypeError** is thrown when the next or previous item of a specific type cannot be found.

This architecture cleanly separates concerns (data storage, media types, user interaction) and enables extensibility for future media types or observer roles.