

# Media Player

**By:**

**IIT2019211 - Divy Agrawal  
IIT2019210 - Aditya Aggarwal  
IIB2019017 – Ambika Singh Kaushik**

# Roadmap



# Basic Concepts

Let's first discuss about UML Diagrams

# UML Diagram

## Use Case Diagram

Describe what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how.

- Contains
  - ◆ Use Case
  - ◆ Actors
  - ◆ The System Modeled
- Use
  - ◆ Determining features (requirements)
  - ◆ Notational simplicity makes use case diagrams a good way for developers to communicate with clients
  - ◆ Generating test cases

## CRC Diagram

Describes the functionality of various classes present in the application. Also tells that which classes in the system are in collaboration.

- Contains
  - ◆ Class Names
  - ◆ Class Responsibilities
  - ◆ Class Collaborations
- Use
  - ◆ Helps in designing class diagram
  - ◆ Class responsibilities are the attributes and operations of the class
  - ◆ Used by developer and application domain expert

## Class Diagram

Describes the attributes and methods of the class and the constraints imposed on the system. Provide the static view of the project.

- Contains
  - ◆ Class Names
  - ◆ Class Attributes
  - ◆ Class Operations
- Use
  - ◆ Forward and Reverse Engineering
  - ◆ Analysis and design of the static view of the application
  - ◆ Used by developer and application domain expert

# Relationship Types

Represents "is-a" relationship  
The child classes "inherit" the common functionality defined in the parent class.

*Generalization*

*Association*

A structural link between two peer classes.  
A solid line connecting two classes.

A special type of association. It represents a "part of" relationship.  
Objects of both classes have separate lifetimes.  
A solid line with an unfilled diamond connecting two classes.

*Aggregation*

*Composition*

A special type of aggregation where parts are destroyed when the whole is destroyed.  
A solid line with a filled diamond connecting two classes.

Is a directed relationship  
Used to show that some UML element or a set of elements requires, needs or depends on other model elements for specification or implementation

*Dependency*

# About our project

Functionalities, Main Dependency, Future Scope

# Project Tasks

- ★ To create a simple media player
- ★ A positioning slider to jump to certain points in the media clip.
- ★ A play/pause button.
- ★ A volume button that provide volume control.
- ★ A media properties button that provides detailed media information
- ★ To play media from any location
- ★ Set speed for playing media.

One to one Relationship - a user could open only one media player window and play a media.

# Dependency

The whole functionality of the player depends upon:

- ★ Open button which will fetch the media path and load it.
- ★ Media playing in the player
- ★ The media player supports media of some specific extensions only like: mp3, mp4, wav, FLV, so, this is one of the major dependency of the media.



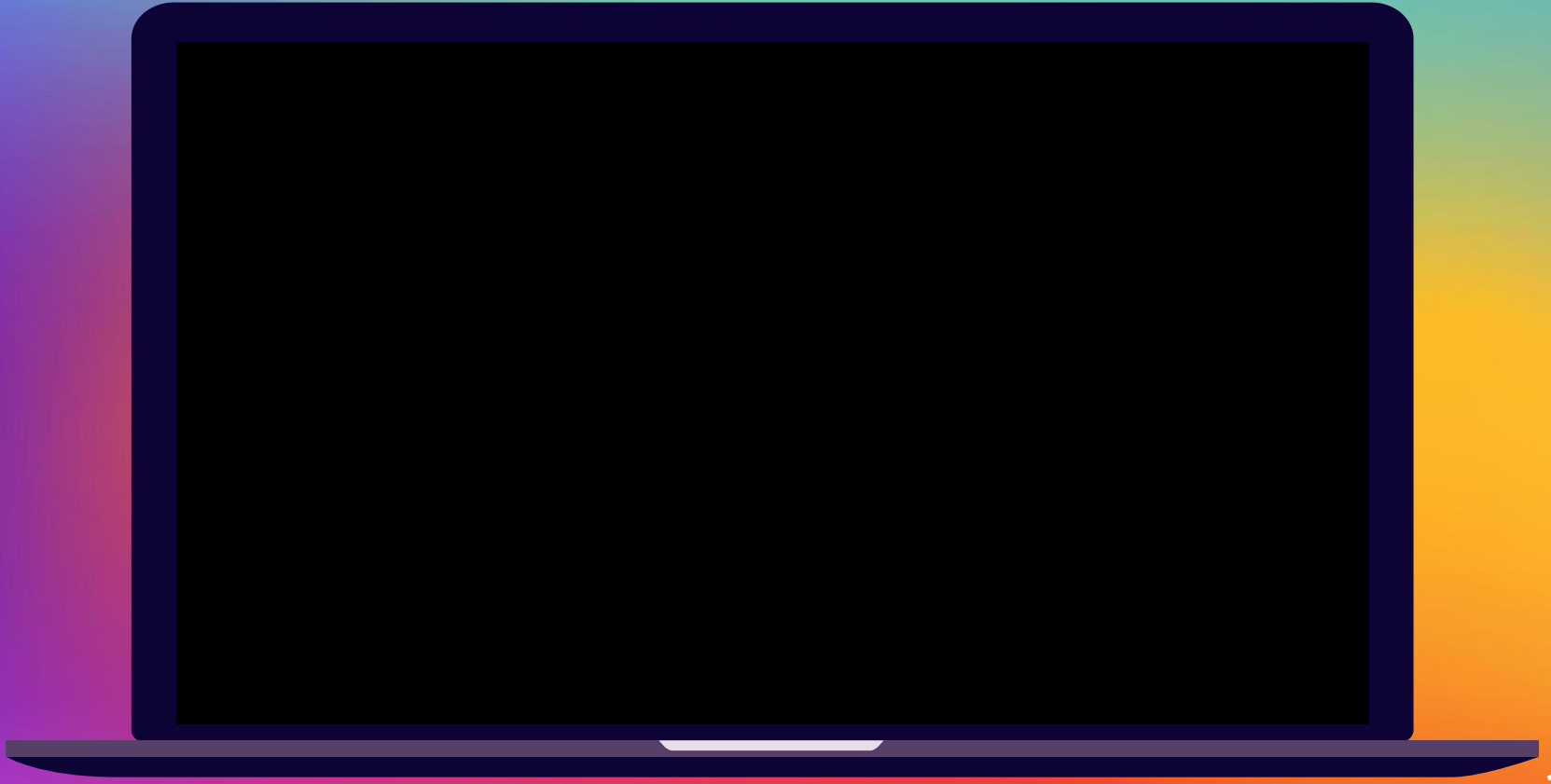
## Further Extension

- ★ We can add a button for recording which can provide good quality with less video size.
- ★ We can add an option of creating a playlist in which we can select multiple videos at a single time.
- ★ We can add an option of adding subtitles in the player.
- ★ We can add an option of opening more types of media with different extensions.

# UML Diagram

Use Case, CRC, Class Diagram

# Output



Thank You

