About Diagrams:

Q) What is crc diagram, class diagram and use case diagram and why do we use them?

CRC-

- **A card Contains 3 components**
  - **Class name**
  - **Responsibilities - something that a class knows or does**
  - **Collaborating Classes - another class with which it interacts to fulfill its Responsibilities.**

- **Use**
  - **helps in designing Class diagram**
  - **Class responsibilities are the class's attributes and methods**
  - **used by - developer and application domain expert**

Class Diagram-

These show the classes in the software and how they interrelate.

- **Contains**
  - **Classes**
  - **Their attributes**
  - **methods**
- **Use**
  - **Forward and reverse engineering**
  - **Analysis and design of the static view of an application**
  - **Used by- developer and application domain expert**

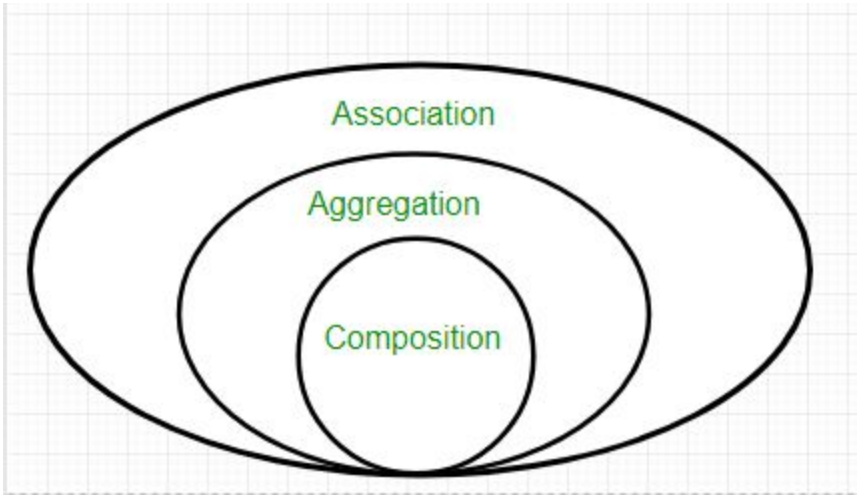
Use Case Diagram-

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 - An actor is who or what initiates the events involved in that task.**
  - **The system modeled**
- **Use**
  - **determining features (requirements).**
  - **notational simplicity makes usecase diagrams a good way for developers to communicate with clients.**
  - **generating test cases.**

These diagrams serve the purpose of the base of the project. These are the blueprint of the project , are very important before starting the coding. Basically, they tell the basic Design.

Q) What are different types of relationships?

1. **Association:** Association is a relation between two separate classes which establishes through their Objects. Association can be **one-to-one, one-to-many, many-to-one, many-to-many**.It has two forms: Composition and Aggregation
    a) **Aggregation:**
        i) It represents **Has-A** relationship.
        ii) It is a **unidirectional association** i.e. a one way relationship. For example, bank can have different employees but vice versa is not possible and thus unidirectional in nature
        iii) In Aggregation, **both the entries can survive individually** which means ending one entity will not effect the other entity.eg: bank will exist if some employees leave and similarly employees will exist if they are kicked out of the bank.
    b) **Composition:**
        i) Composition is a restricted form of Aggregation in which two entities are highly dependent on each other.
        ii) It represents **part-of** relationship.
        iii) In composition, **both the entities are dependent** on each other.
        iv) When there is a composition between two entities, the composed object **cannot exist** without the other entity.
        v) 

This photo should be attached in ppt

2. Generalization, specialization: generalization = upcasting, specializtion = downcasting

3. Inheritance: Inheritance can be defined as the process where **one class acquires the properties (methods and fields) of another**. With the use of inheritance the information is made manageable in a hierarchical order. The class which inherits the properties of other is known as **subclass (derived class, child class)** and the class whose properties are inherited is known as superclass **(base class, parent class)**. **extends is the keyword used to inherit the properties of a class**.

To be Shown in diagram:

What are Primary and Secondary Actors, include, extend, system boundary?

About Our Project:

Q) Many to one relationship or what type of relationship.

Q)what is the main dependency of your project without which it can't run?

- **Opening of the media player window**

Q) what are the functionalities to be implemented?

- **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**
- **And frame rate control.**
- **To play media from any location**
- **Set speed for playing media.**

Q) where it can be further used?