**Experiment no- 03**

Name: Suraj P. Patil

Roll No: 3034

URN: 20131086

Class: TY(A)

Batch: T-2

**Title:**

*Draw a class diagram.*

A class diagram is a type of UML (Unified Modeling Language) diagram that depicts the structure of a system by showing its classes, their attributes, operations, and the relationships among them.

A class diagram for YouTube, a popular social media application, might include the following classes:

1. **Class: User**

* Attributes:
  + username: string
  + email: string
  + password: string
  + subscriptions: list of Channel
  + videos: list of Video
* Methods:
  + login(): void
  + logout(): void
  + subscribe(channel: Channel): void
  + unsubscribe(channel: Channel): void
  + upload(video: Video): void

1. **Class: Channel**

* Attributes:
  + name: string
  + owner: User
  + subscribers: list of User
  + videos: list of Video
* Methods:
  + addSubscriber(user: User): void
  + removeSubscriber(user: User): void
  + upload(video: Video): void

1. **Class: Video**

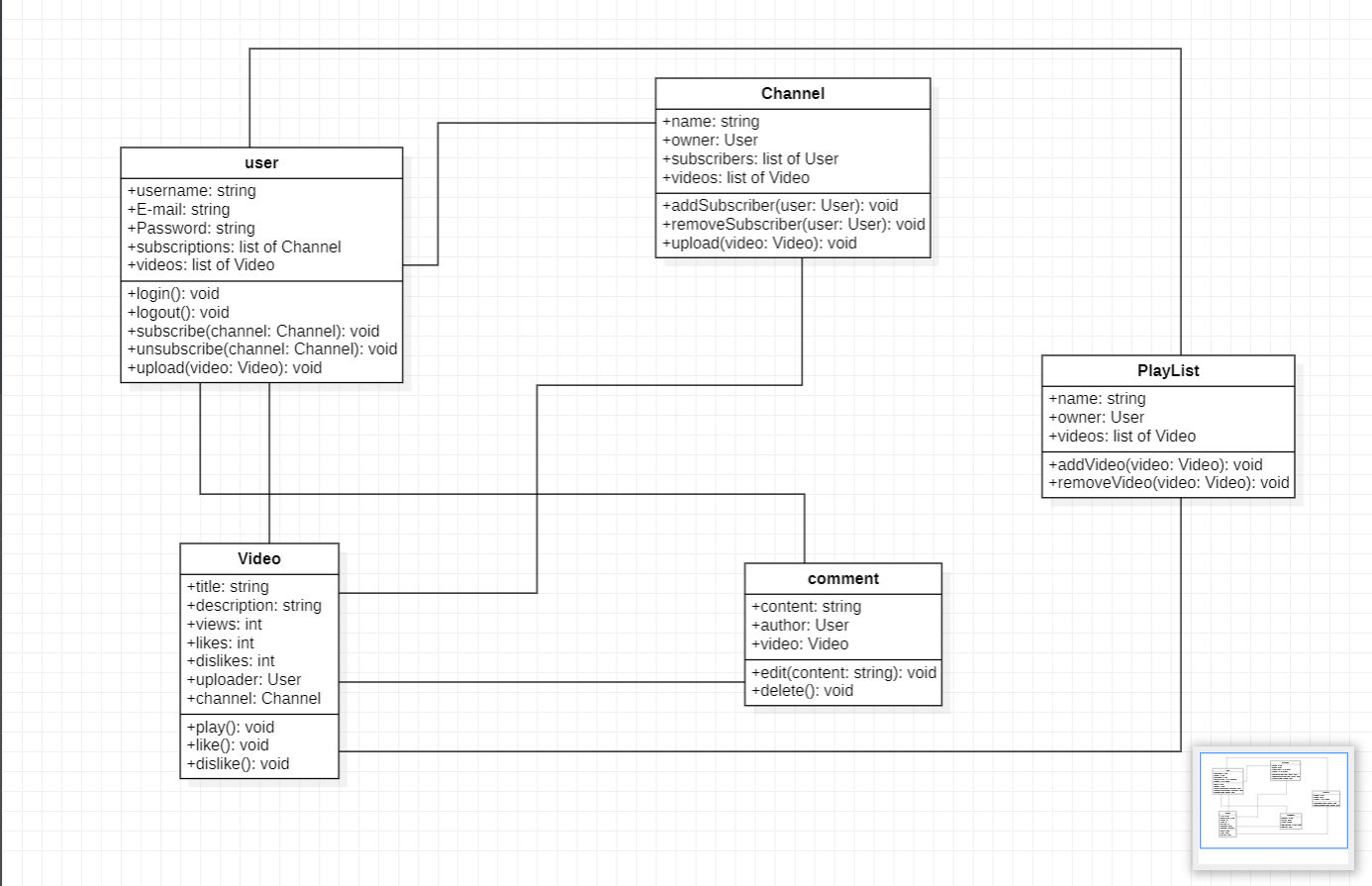
* Attributes:
  + title: string
  + description: string
  + views: int
  + likes: int
  + dislikes: int
  + uploader: User
  + channel: Channel
* Methods:
  + play(): void
  + like(): void
  + dislike(): void

1. **Class: Comment**

* Attributes:
  + content: string
  + author: User
  + video: Video
* Methods:
  + edit(content: string): void
  + delete(): void

1. **Class: Playlist**

* Attributes:
  + name: string
  + owner: User
  + videos: list of Video
* Methods:
  + addVideo(video: Video): void
  + removeVideo(video: Video): void



To join classes in a class diagram using StarUML, you can use different types of arrows depending on the relationship between the classes. Here are some common types of arrows and their meanings:

1. Association arrow: This arrow is used to indicate that one class is associated with another class in some way. You can add an association arrow by dragging a line from one class to another and adding an arrowhead to indicate the direction of the association.
2. Inheritance arrow: This arrow is used to indicate that one class is derived from another class. You can add an inheritance arrow by dragging a line with an arrowhead from the derived class to the base class.
3. Aggregation arrow: This arrow is used to indicate that one class contains one or more instances of another class. You can add an aggregation arrow by dragging a line with a diamond-shaped arrowhead from the containing class to the contained class.
4. Composition arrow: This arrow is used to indicate that one class is composed of one or more instances of another class. You can add a composition arrow by dragging a line with a filled diamond-shaped arrowhead from the composing class to the composed class.

To create any of these arrows in StarUML, select the appropriate arrow type from the toolbox on the left-hand side of the screen and drag it from one class to another. You can also add labels to the arrows to indicate the name and multiplicity of the relationship.

***Based on the relationships described in the class descriptions for the YouTube system, here is how the classes can be connected using different types of arrows:***

1. User class and Channel class: There is a many-to-many relationship between these classes, as a user can subscribe to many channels and a channel can have many subscribers. Therefore, we can use an association arrow between the User class and the Channel class.
2. User class and Video class: There is a one-to-many relationship between these classes, as a user can upload many videos but a video can only be uploaded by one user. Therefore, we can use an association arrow between the User class and the Video class.
3. Channel class and Video class: There is a one-to-many relationship between these classes, as a channel can have many videos but a video can only belong to one channel. Therefore, we can use an association arrow between the Channel class and the Video class.
4. Video class and Comment class: There is a one-to-many relationship between these classes, as a video can have many comments but a comment can only belong to one video. Therefore, we can use an association arrow between the Video class and the Comment class.
5. User class and Comment class: There is a one-to-many relationship between these classes, as a user can post many comments but a comment can only be posted by one user. Therefore, we can use an association arrow between the User class and the Comment class.
6. User class and Playlist class: There is a one-to-many relationship between these classes, as a user can create many playlists but a playlist can only belong to one user. Therefore, we can use an association arrow between the User class and the Playlist class.
7. Playlist class and Video class: There is a many-to-many relationship between these classes, as a playlist can contain many videos and a video can belong to many playlists. Therefore, we can use an association arrow between the Playlist class and the Video class.