**SYSTEM REQUIREMENTS**

FUNCTIONAL

REQUIRMENTS

## Registration and login:

* + System shall allow users to sign up and create their personal channel.
  + System shall allow users to log in.
  + System shall allow users to log out.
* Video uploading:
  + System shall allow users to upload videos. o System shall allow users to delete videos.
  + System shall allow users to add videos to their archive.
  + System shall allow users to add caption to their videos.
  + System shall make users choose their video category.
  + System shall allow users to create playlists.
  + System shall allow users to add and remove likes on videos.

**SYSTEM REQUIREMENTS**

* + System shall allow users to add comments on videos.
  + System shall allow users to add and remove dislike.
* Creating playlists:
  + System shall allow users to add videos to their playlist.
  + System shall allow users to rename their playlists.
  + System shall allow users to change the order of the videos added to their playlists.
* Search:
  + System shall allow users to search for videos using keywords.
* Subscribe:

**SYSTEM REQUIREMENTS**

* + System shall allow users to subscribe to channels.
  + System shall allow users to unsubscribe to channels.
* Notifications:
  + System shall allow users to choose the sensitivity of the notification.
  + System shall send notification to the user based on their preference:
* Recommendations:
  + System shall show users recommended videos based on:
    - Their recent watched videos
    - Their history
    - Categories they watch most

**SYSTEM REQUIREMENTS**

NON-FUNCTIONAL

REQUIREMENTS

## Usability:

* + System should be easy to navigate.
  + System should be accessible to users with disabilities.
  + System should be optimized for different devices and screen sizes.
  + Systemshould be intuitive .
* Security:
  + System should send otg code to user after registration.
  + System should contain hash functions for passwords.
  + System should ensure the security and privacy of

users’ data.

* + - Login credentials
    - Personal information
* Reliability:
  + System should be always reliable and available to users.
  + System should have minimal downtime for maintenance and upgrades .

**SYSTEM REQUIREMENTS**

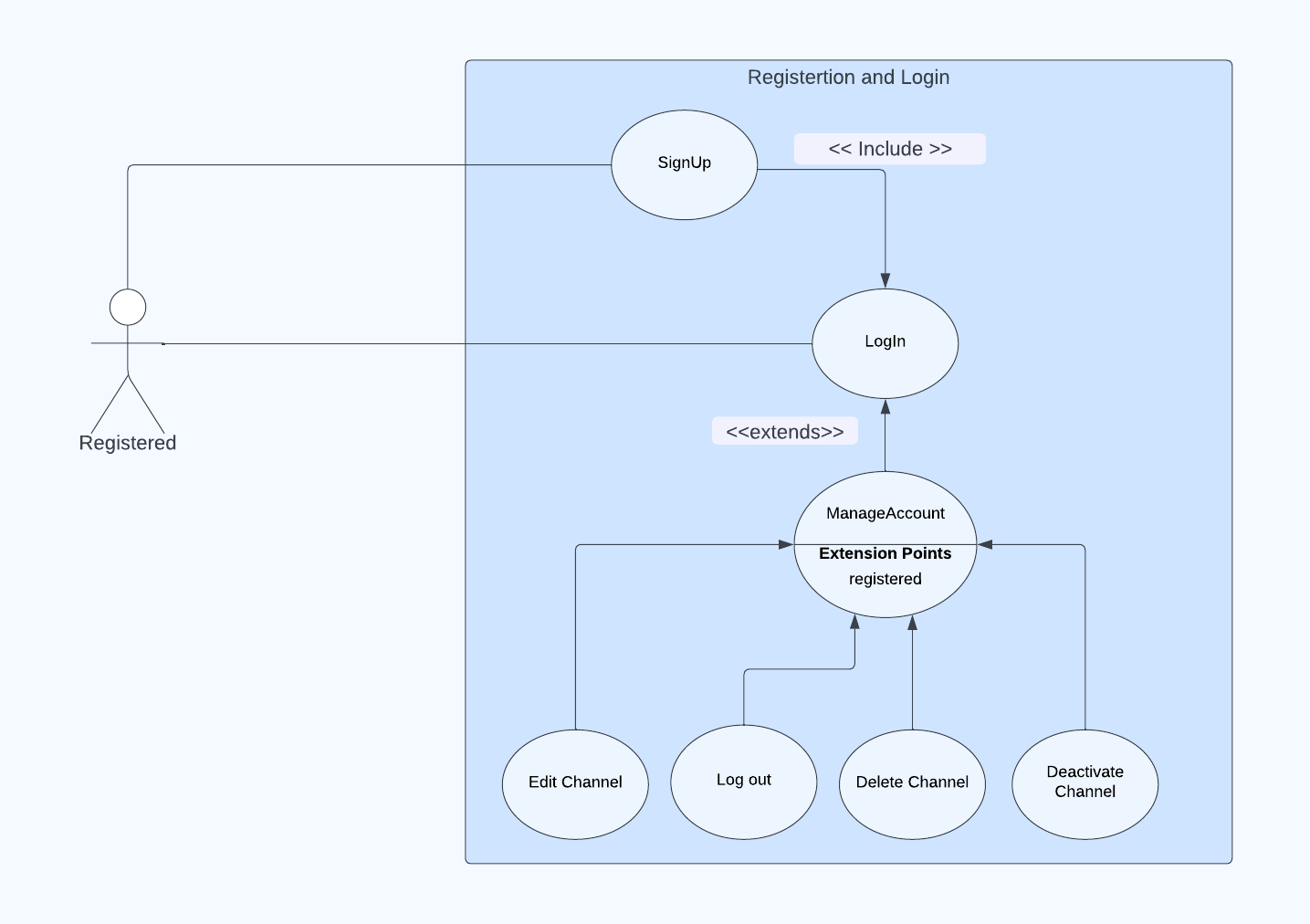
* + System should not lose data.
* Scalability:
  + System should be able to scale up or down to

accommodate changes in users demand, traffic, and content.

* Performance:
  + System should load quickly.
  + System should be able to handle large volumes of users and videos without experiencing slowdowns or crashes.
* Consistency:
  + System should provide same videos to users in all devices.
* Compatibility:
  + System should be compatible with various devices.
  + System should be compatible with various platforms.
    - Desktops
    - Laptops
    - Mobile devices

*USE-CASE AND USE-CASE DESCRIPTION:*

## Registration and Login:



Use case: Signup

Function: Sign up.

Description: User can sign up, create channel.

Input: Sign up. Source: From user. Output: New account. Destination: User.

Main successful scenario: User enter his information (name, email, password) if the information in valid send him a confirmation message after that the account is created. Requirements: None.

Pre-condition: The user is sign up.

Post condition: Create new account.

Use case: Login

Function: Log in.

Description: User can log in to his account.

Input: Email and Password.

Source: From user.

Output: User logged in his account.

Destination: User.

Main successful scenario: User enter his information (email, password) if its right log in successfully otherwise send error message and ask him to enter the correct information. Requirements: The user already has an account.

Pre-condition: The user has account.

Post condition: The user logged in and can manage your account, like, comments.

Use case: Manage account

Function: Manage Account.

Description: User can edit channel, delete channel, deactivate channel, and log out.

Input: None.

Source: From user.

Output: User update your channel.

Destination: User.

Main successful scenario:

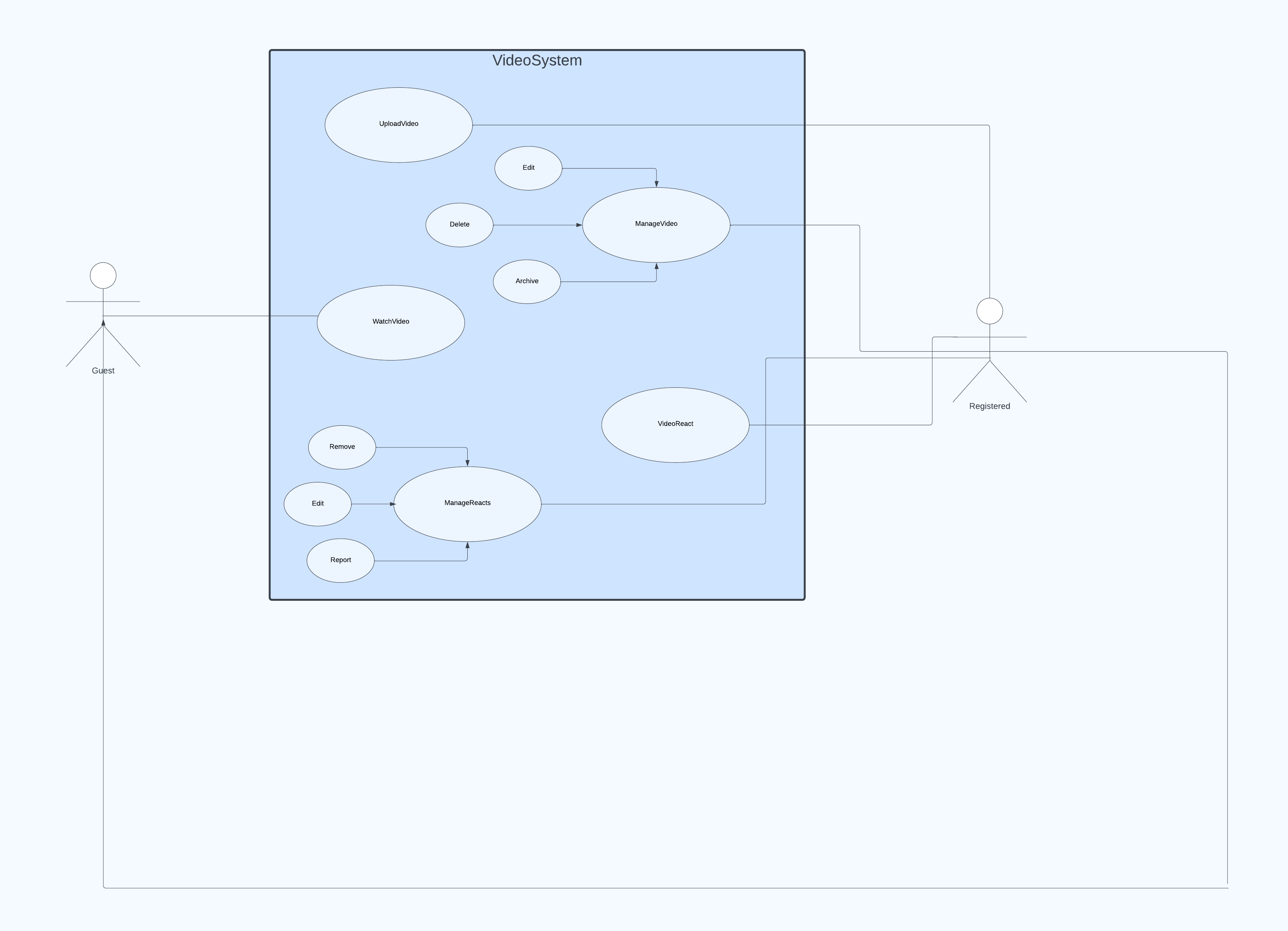
* If the user want edit channel can change username, password, profile picture and status.
* If the user deletes his channel, then he will be out of system boundary and the has 30 days to remove his account from system.
* If the user logs out, but he can't edit his account.
* If the user deactivate channel, will be not active in the system.

Requirements: The user already has an account.

Pre-condition: The user is logged.

Post condition: The modification desired by user have been made.

Video system:



Use case: upload video

Function: Upload new video.

Description: The user can upload a video on their channel or account, and other users can like or dislike the video. The user can also view the number of views on the video.

Inputs: The selected video is the input for this function. The video is selected from the user's device.

Main Scenario: The user logs in to their account. The user clicks on the "Upload" button. The user selects the video they want to upload from their device.

The video is processed and uploaded to the user's channel on the Video Sharing App. Other users can view, like, and dislike the video.

Source: The video is sourced from the user's device.

Output: The output of this function is a new video uploaded to the system.

Destination: The uploaded video is stored on the system.

Pre-Condition: The user must log in to their account before uploading a new video.

Post-Condition: The user has a new video uploaded to their channel, which can be viewed, liked, and disliked by other users.

Use case: Watch video

Actor: Registered User or non registered user

Description: The user wants to watch a video on the Video Sharing App. They can search for a specific video or browse through different videos on the home page. Once they find a

video they want to watch, they can click on it to start playing the video. While watching the video, they can pause, play, and adjust the volume.

Inputs: Keywords for search or browsing, video controls (play, pause, volume).

Outputs: Video is played on the user's screen.

Main Scenario: The user either searches for a specific video or browses through different videos on the home page. The user clicks on the video they want to watch. The video starts playing on the user's screen. While watching the video, the user can pause, play, and adjust the volume as needed.

Pre-conditions: No pre-condition

Post conditions: The user has watched the selected video

Use case: Manage video

Actor: Registered User

Description: The registered user wants to manage the videos on their channel. They can edit the video details, such as the title, description, and tags. They can also delete a video from their channel or set it to private so that only certain users can view it.

Inputs: Video details to edit (title, description, tags), delete or private video button click.

Outputs: Video details are updated or video is deleted/set to private.

Main Scenario: The user logs in to their account on the Video Sharing App. The user navigates to their channel and clicks on the video they want to manage. The user can edit the video details, such as the title, description, and tags. The user can choose to delete the video from their channel or set it to private so that only certain users can view it.

Preconditions: User must be registered and logged in to the Video Sharing App, user must have at least one video uploaded to their channel.

Post conditions: The selected video's details are updated or the video is deleted/set to private.

Use case: Video react

Actor: Registered User

Description: The registered user wants to react to a video by choosing a specific react. They can choose from a variety of reacts to express their feelings about the video and make comments. The react that the user selects will be displayed on the video page along with the number of users who have reacted to the video

Inputs: Emoticon selection

Outputs: User's emoticon is displayed on the video page

Main Scenario: The user logs in to their account on the Video Sharing App. The user navigates to the video they want to react to and clicks on the like or dislike button and the user can make a comment .The user selects an emoticon to express their feelings about the video. The selected react is displayed on the video page along with the number of users who have reacted to the video.

Pre-conditions: User must be registered and logged in to the Video Sharing App Post conditions: The number of users who have reacted to the video is updated accordingly.

Use case: Manage react

Actor: Registered User

Description: The registered user wants to manage their video reacts or comments by viewing the react or comments they have selected for different videos. They can view the list of their reacts or comments on their profile page and delete any reacts that they no longer want displayed.

Inputs: Delete react button click

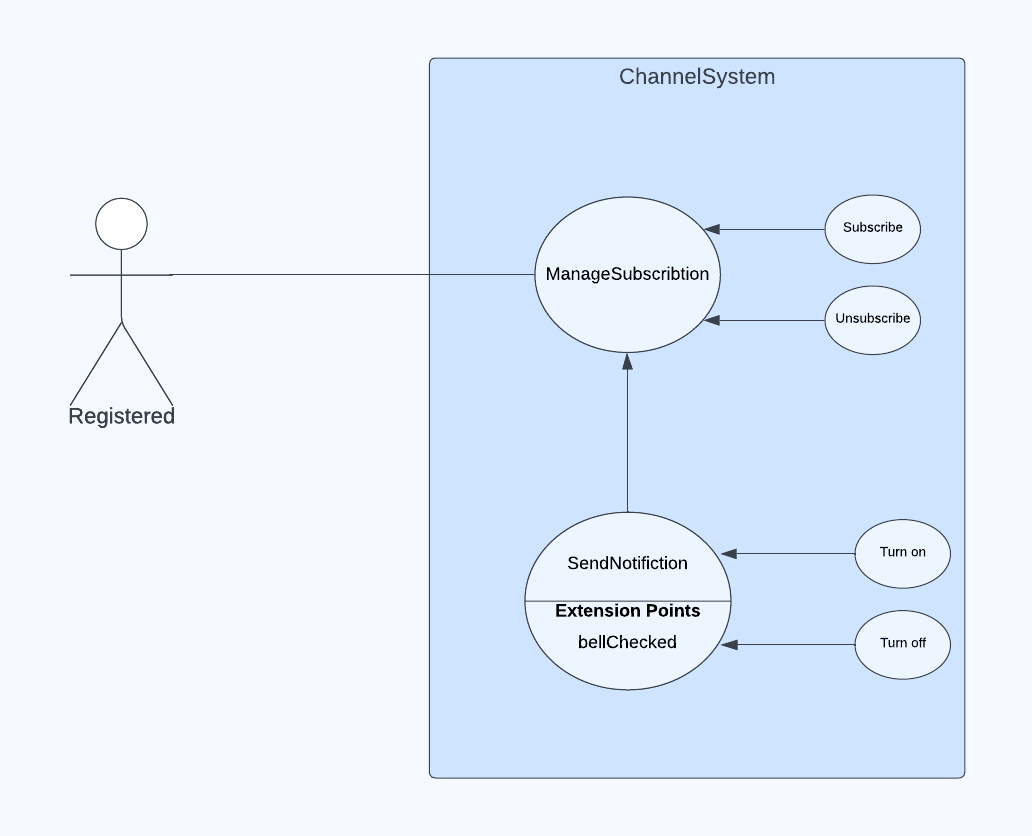
Outputs: Selected react is deleted from user's profile page

Main Scenario: The user logs in to their account on the Video Sharing App. The user navigates to their profile page and clicks on the "React" or comment section. The user can view the list of their reacts or comments and choose to delete any reacts or comments that they no longer want displayed on their profile page.

Pre-conditions: User must be registered and logged in to the Video Sharing App. User must have reacted to at least one video or a comment

Post conditions: The selected react or comment is removed from the user's profile page and is no longer displayed for other users to see.

Channel Subscription:



Use case: Manage Subscribtion

Function: Subscribe to a channel and enable notifications.

Description: Users can subscribe to a channel and choose to enable notifications by toggling the bell icon, which will allow them to receive alerts whenever the channel uploads a new video.

Input: The user clicks on the "Subscribe" button and may also click on the bell icon

Source: From user.

Output: The function allows the user to receive updates about new videos and other content from the subscribed channel.

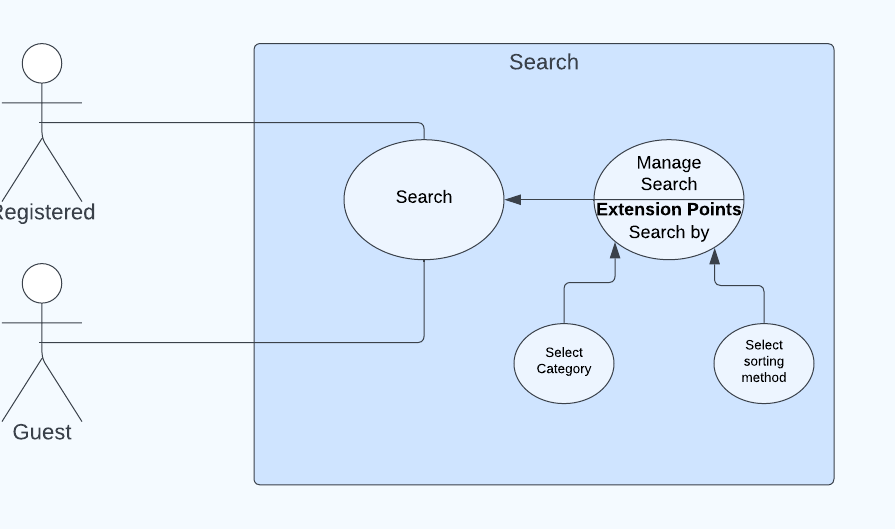
Destination: User.

Main Scenario: The user clicks on the "Subscribe" button, and if they are logged into the system, the function adds the user to the channel's list of subscribers and sends a success message to the user to confirm their subscription.

Pre-condition: The user must be logged in.

Post-condition: The user has successfully subscribed to the channel and can now receive notifications about new content.

Search for video:



Use case: Search

Function: Search for a video.

Description: The search for video use case allows both guests and registered users to search for videos using keywords or video names. The system then displays a list of all related videos, which the user can select and watch and users can sort the displayed list by popularity, rating, upload date, or select a category to find most related videos.

Input: the search query entered by the user in the search bar.

Source: From user.

Output: A list of videos that match the user's search query

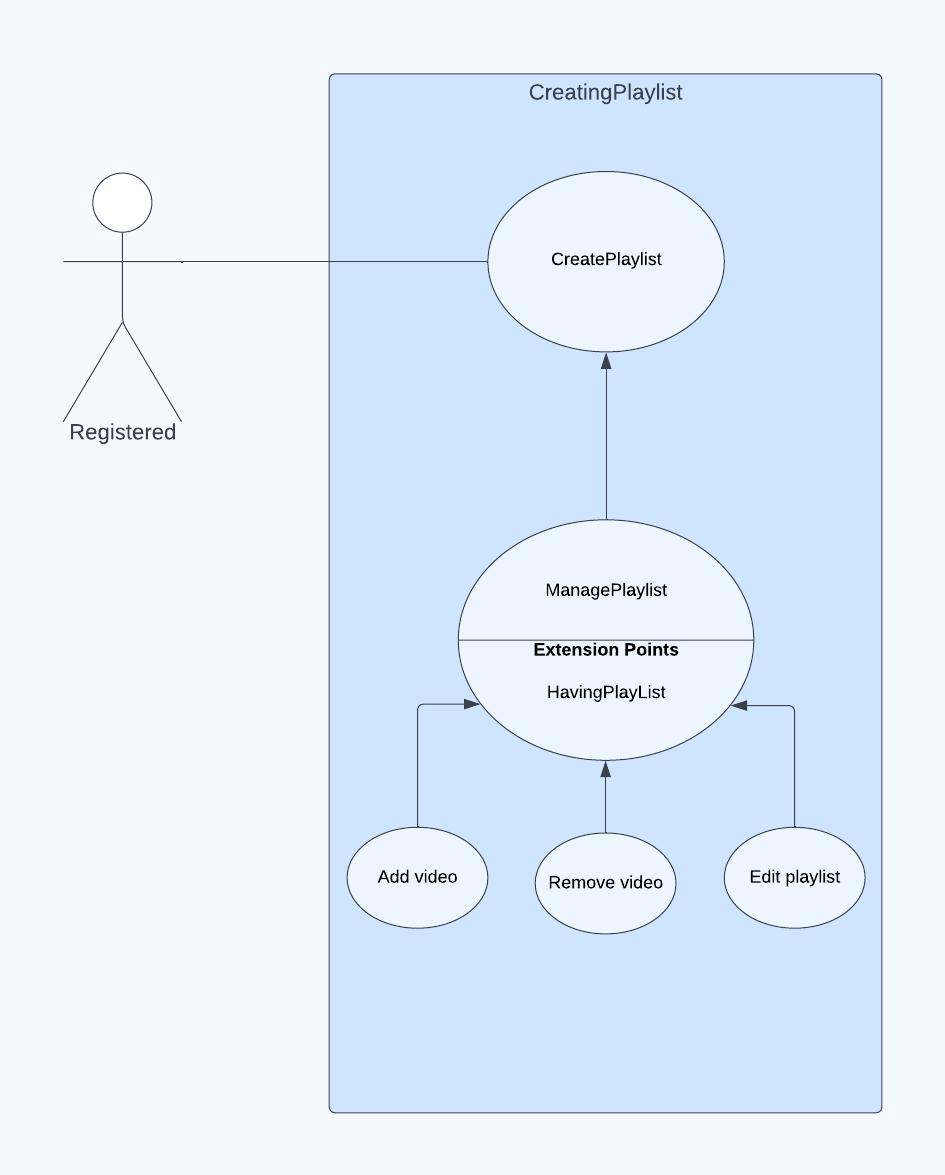
Destination: User or guest.

Main Scenario: The user enters a video name or relevant keywords into the search field. The system then searches for videos that match the user's query and displays a list of relevant videos on the search results page. The user can browse through the list of videos and select one to watch, and the system plays the selected video for the user

Pre-condition: The webpage is displayed with search bar for the user.

Post-condition: The user can easily find all the videos related to their search inputs.

CreatingPlaylist:



Use case: Create playlist

Function: CreatingPlaylist

Description:

User can create a new playlist.

On creating a new playlist, user can now manage their playlist. Input: Name, initialize video/s, initialize privacy, initialize order. Source: From user.

Output: New playlist is created.

Destination: User.

Main Scenario: On wanting to create a new playlist users needs to name their playlist, initialize video/s in the playlist, initialize playlist’s privacy and its sorting order.

Requirements :User must be registered.

Pre-condition: User must be logged in.

Post-condition: A playlist is now created in user’s channel.

Use case: Manage playlist

Function: ManagingPlaylist

Description: User can manage their playlists.

On having a playlist, user can now manage their playlist.

Input: Add video, remove video, edit playlist.

Source: From user.

Output: Playlist is updated.

Destination: User.

Main Scenario: On wanting to update playlist users are able to add videos to their playlist, remove videos from their playlist, edit their playlist (from being able to rename to change privacy to change the sorting method of their playlist).

Requirements:

Playlist must be already created

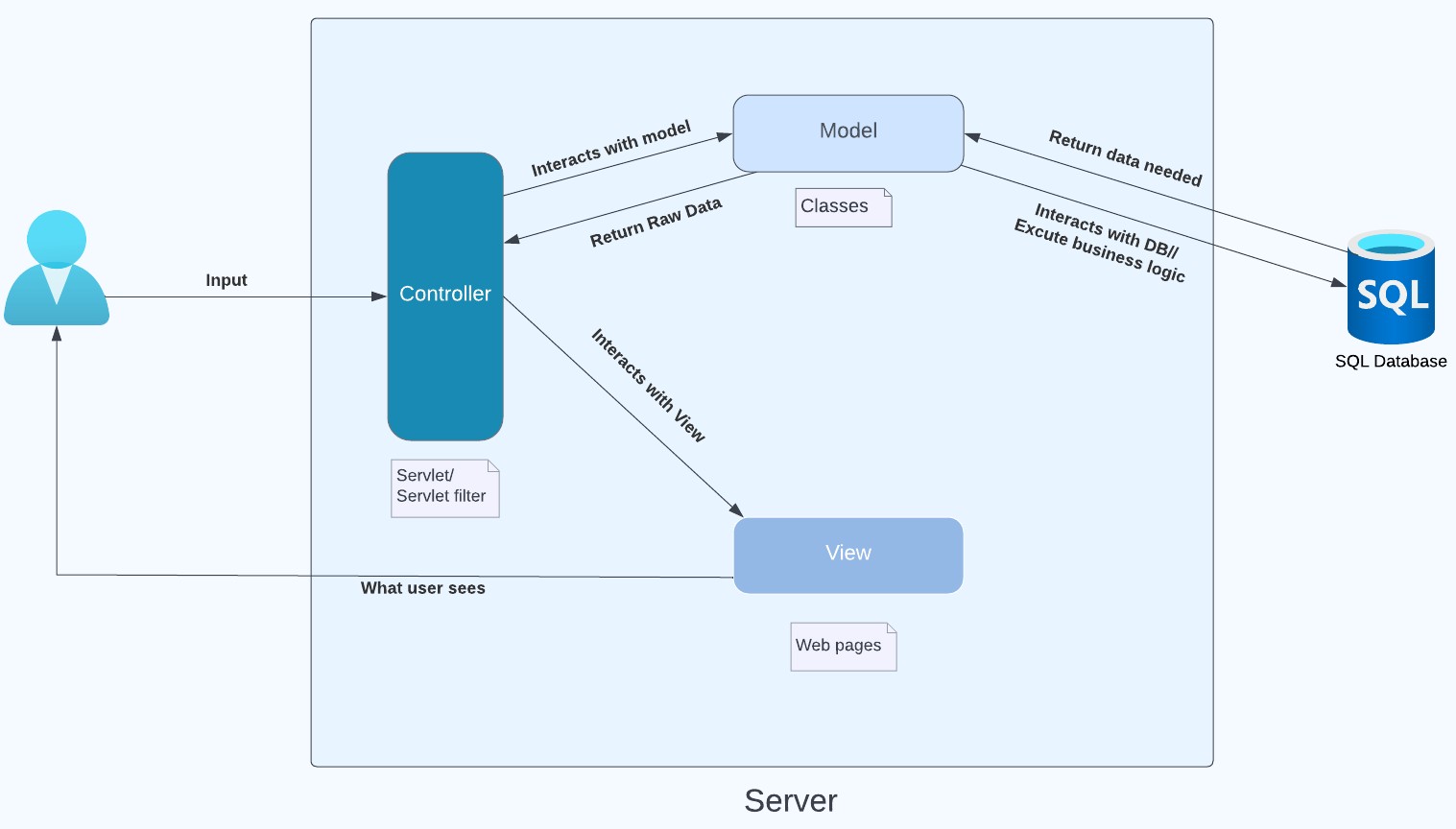
Pre-condition:

User must be logged in.

Post-condition:

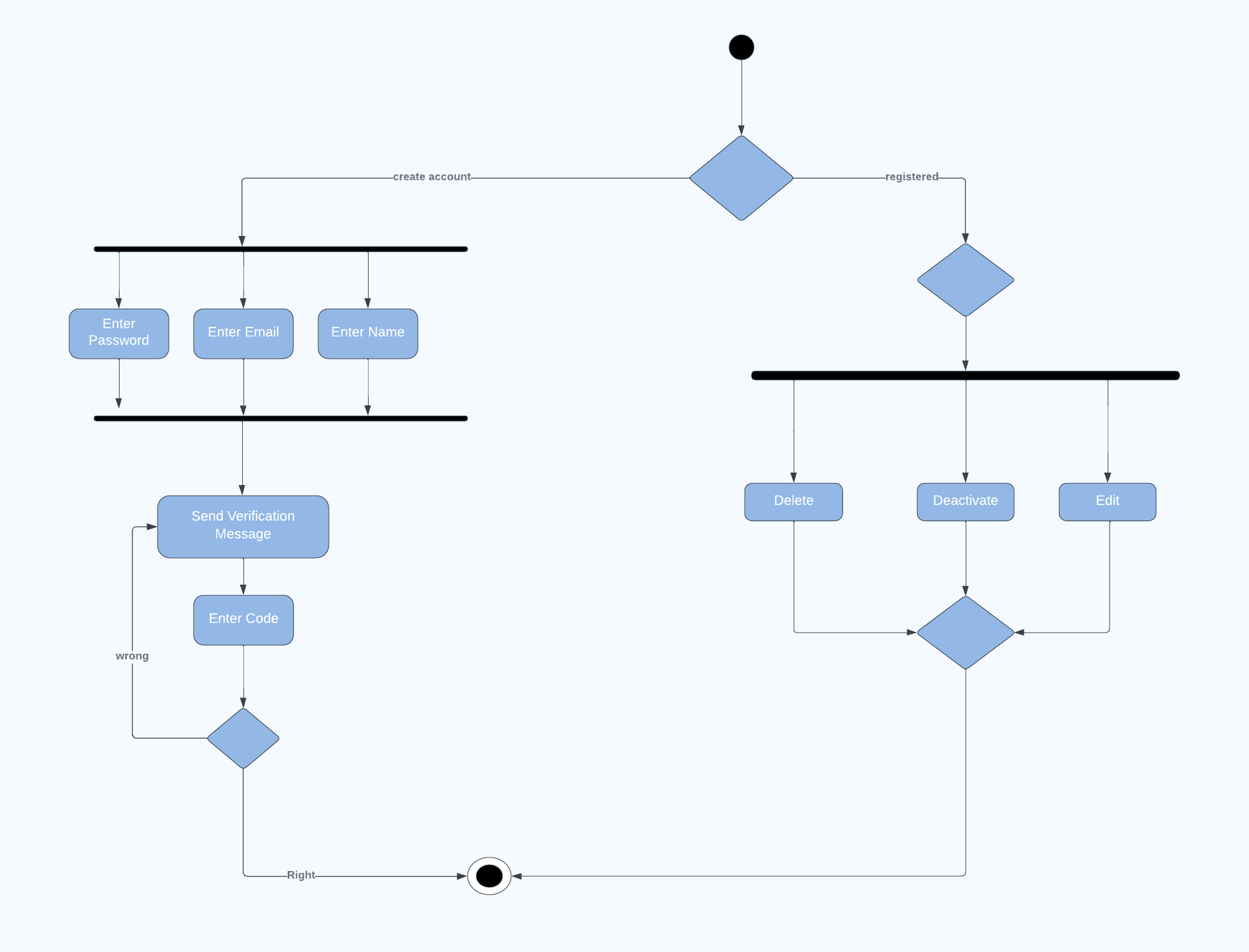
The playlist is now updated in user’s channel.

# SYSTEM ARCHITECTURE:

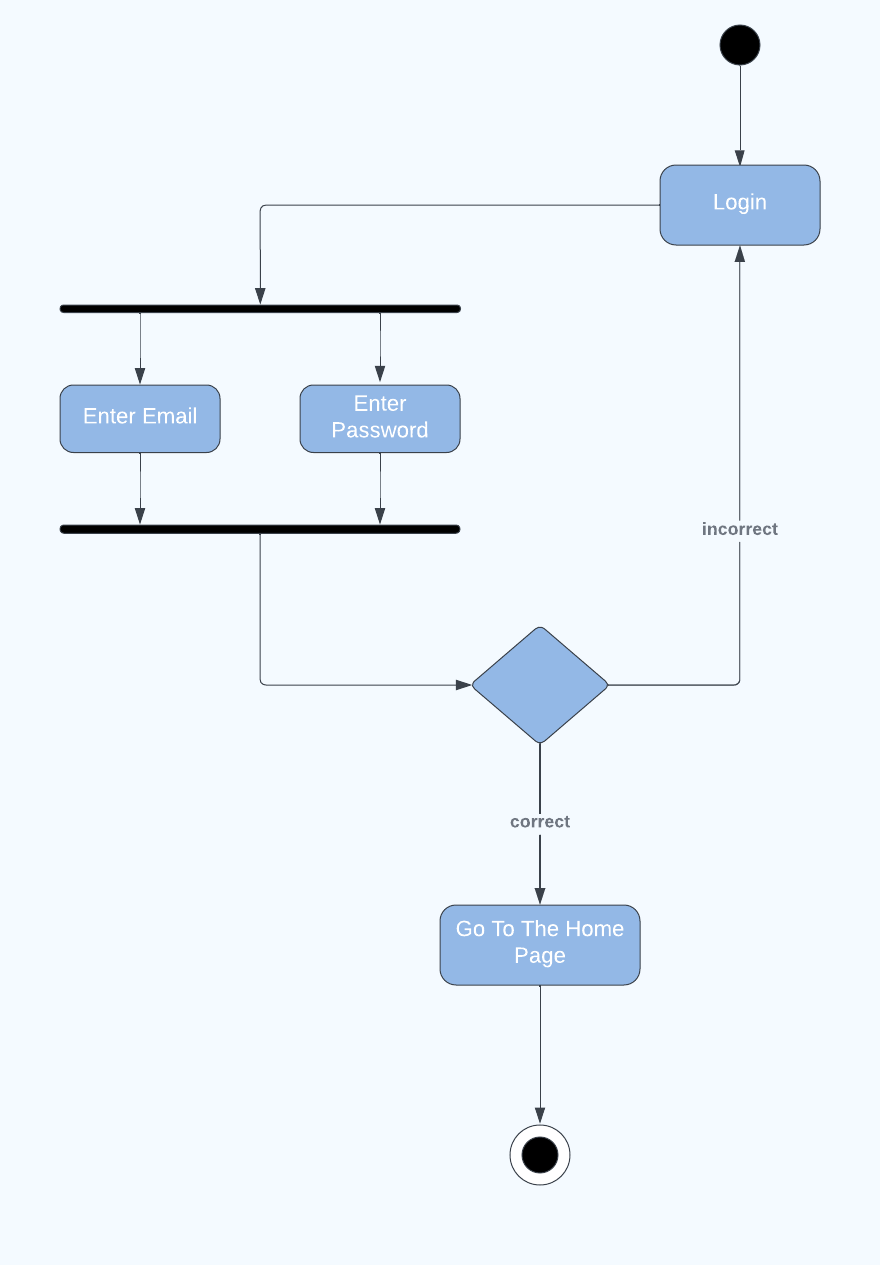


*ACTIVITY DIAGRAM:*

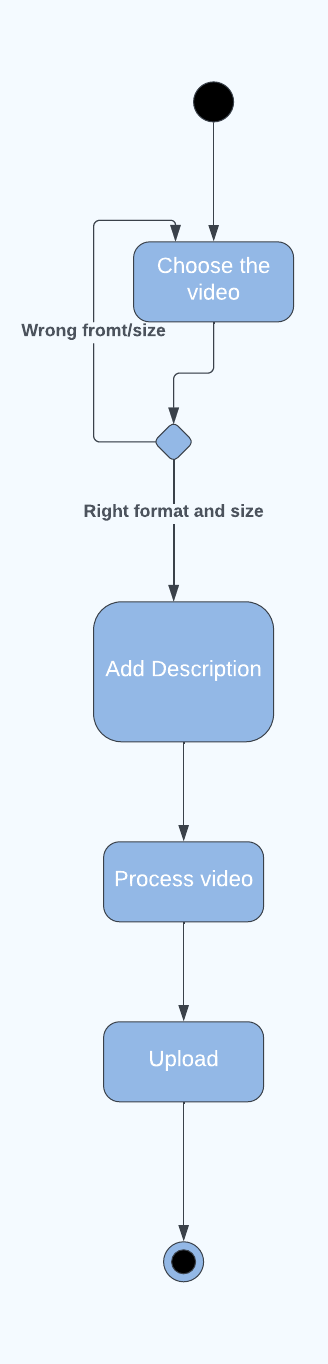
Sign in/up:



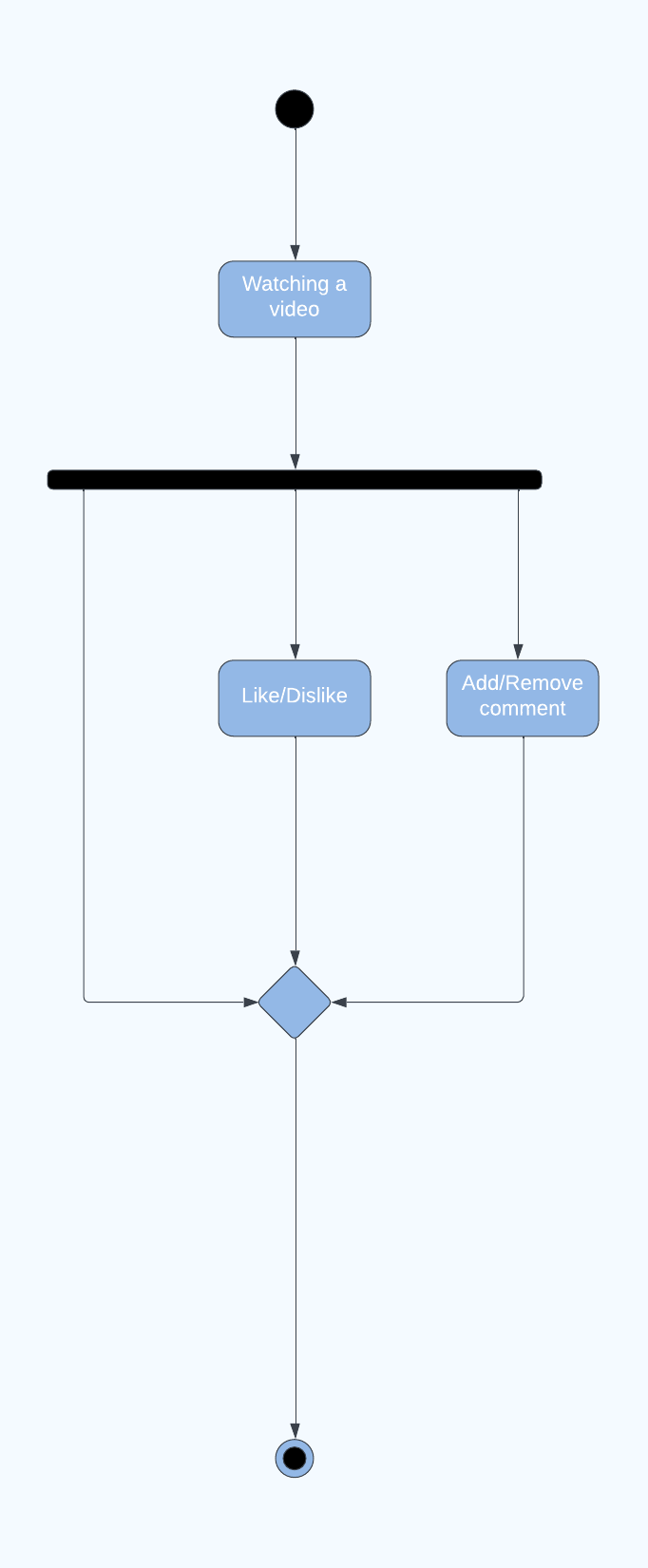
Login:



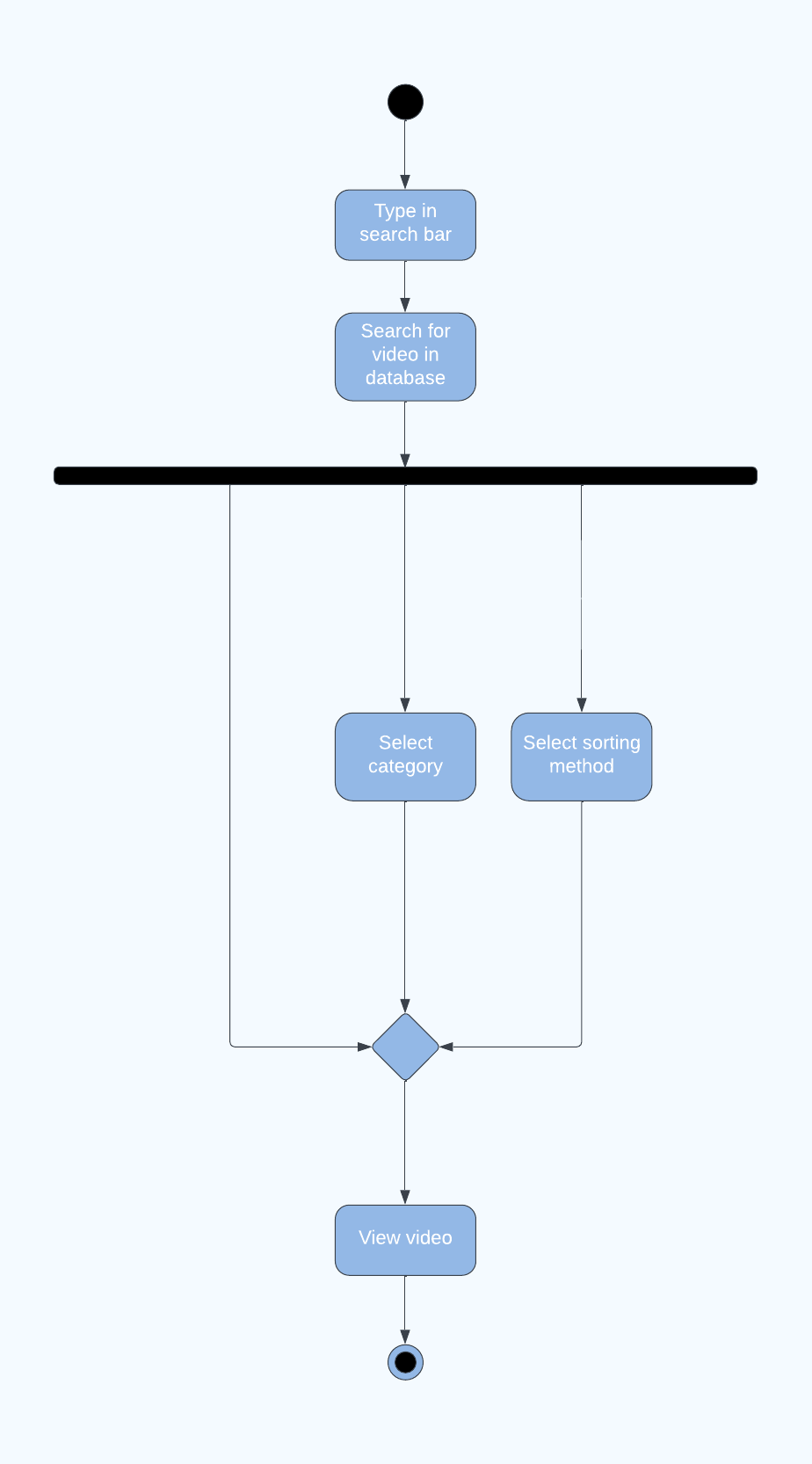
Video uploading:



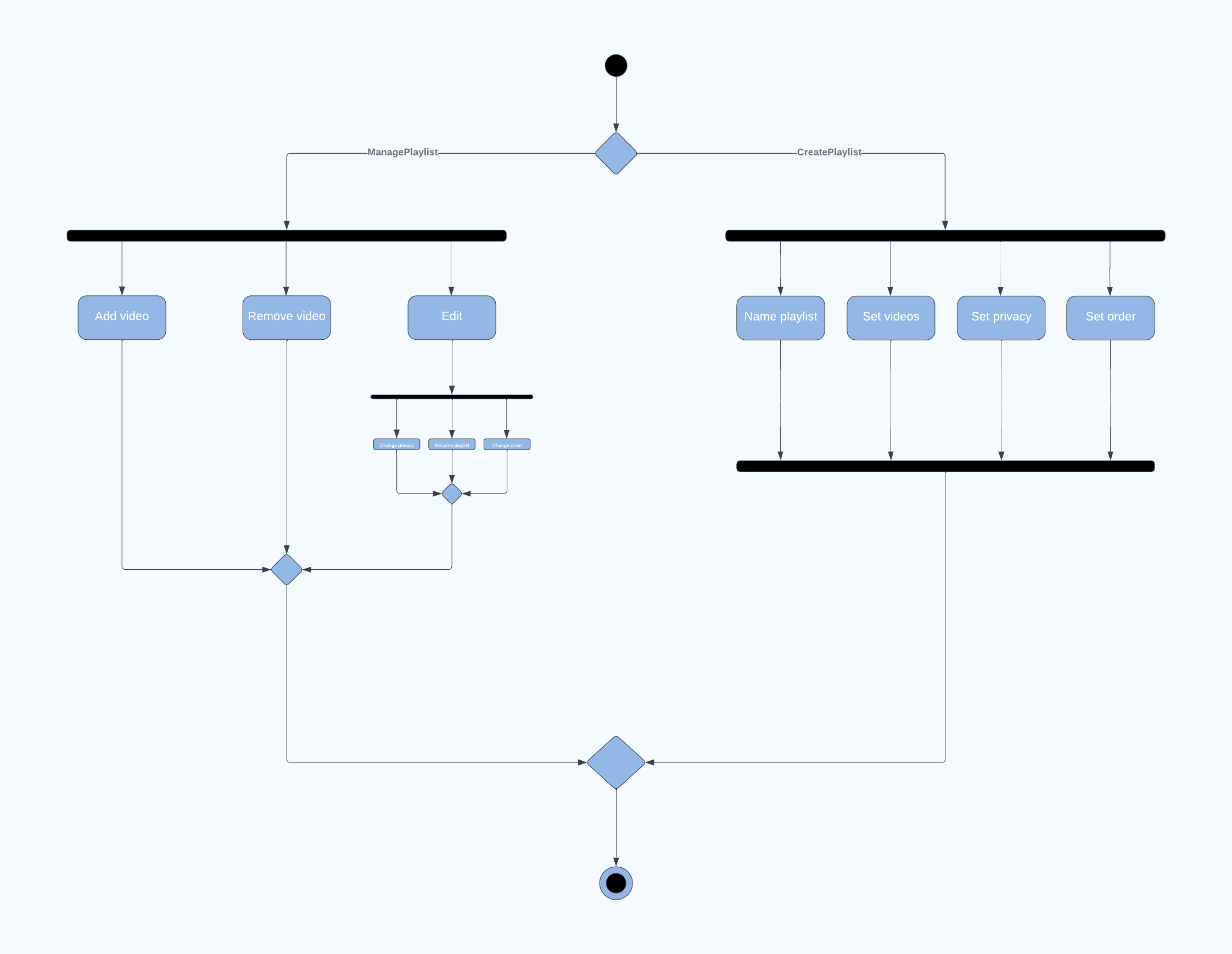
Video reacts:



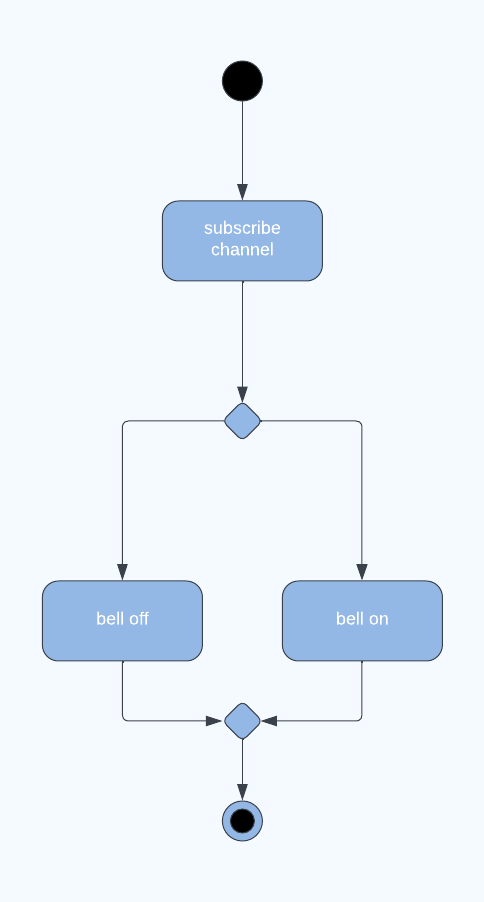
Search:



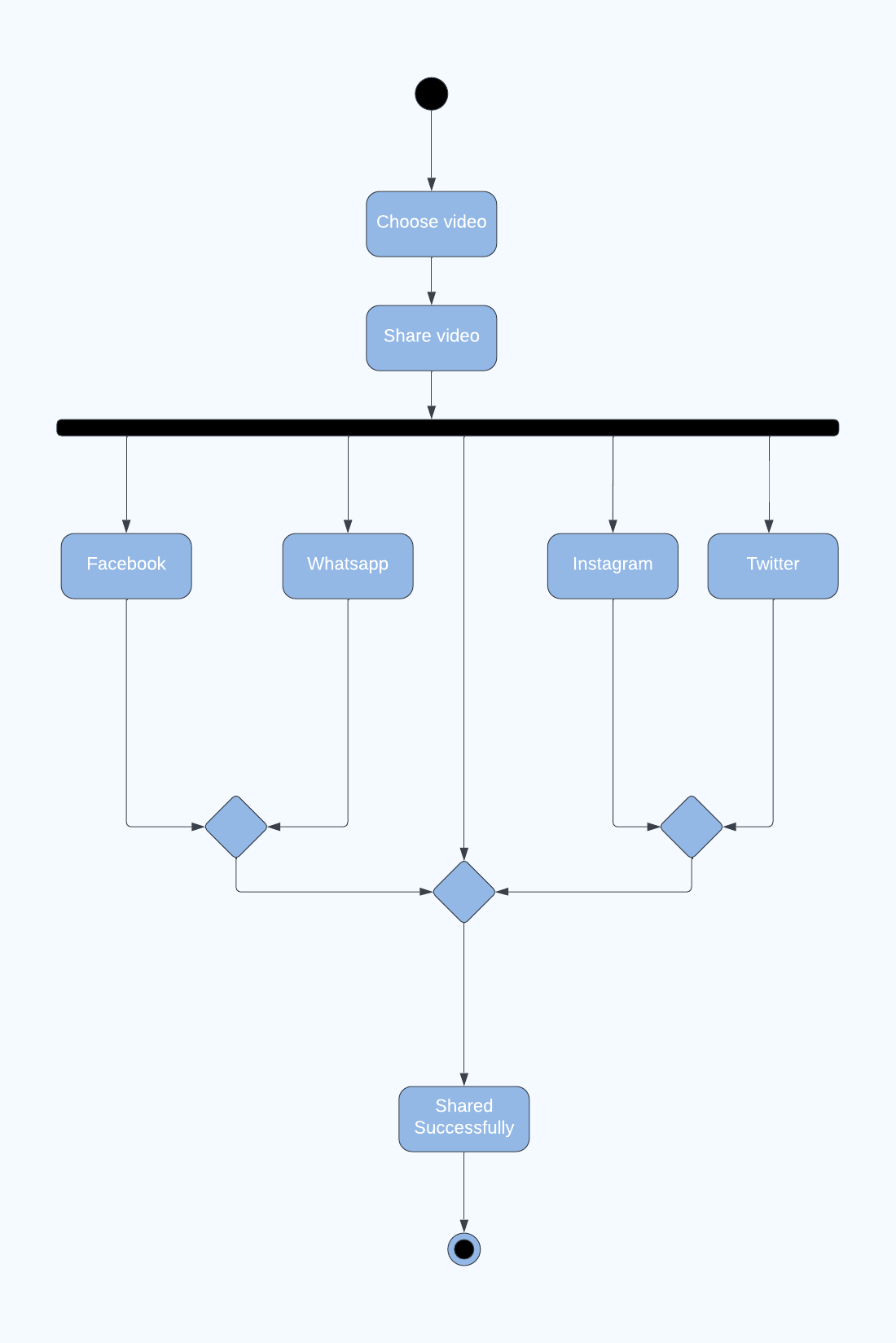
Playlist:



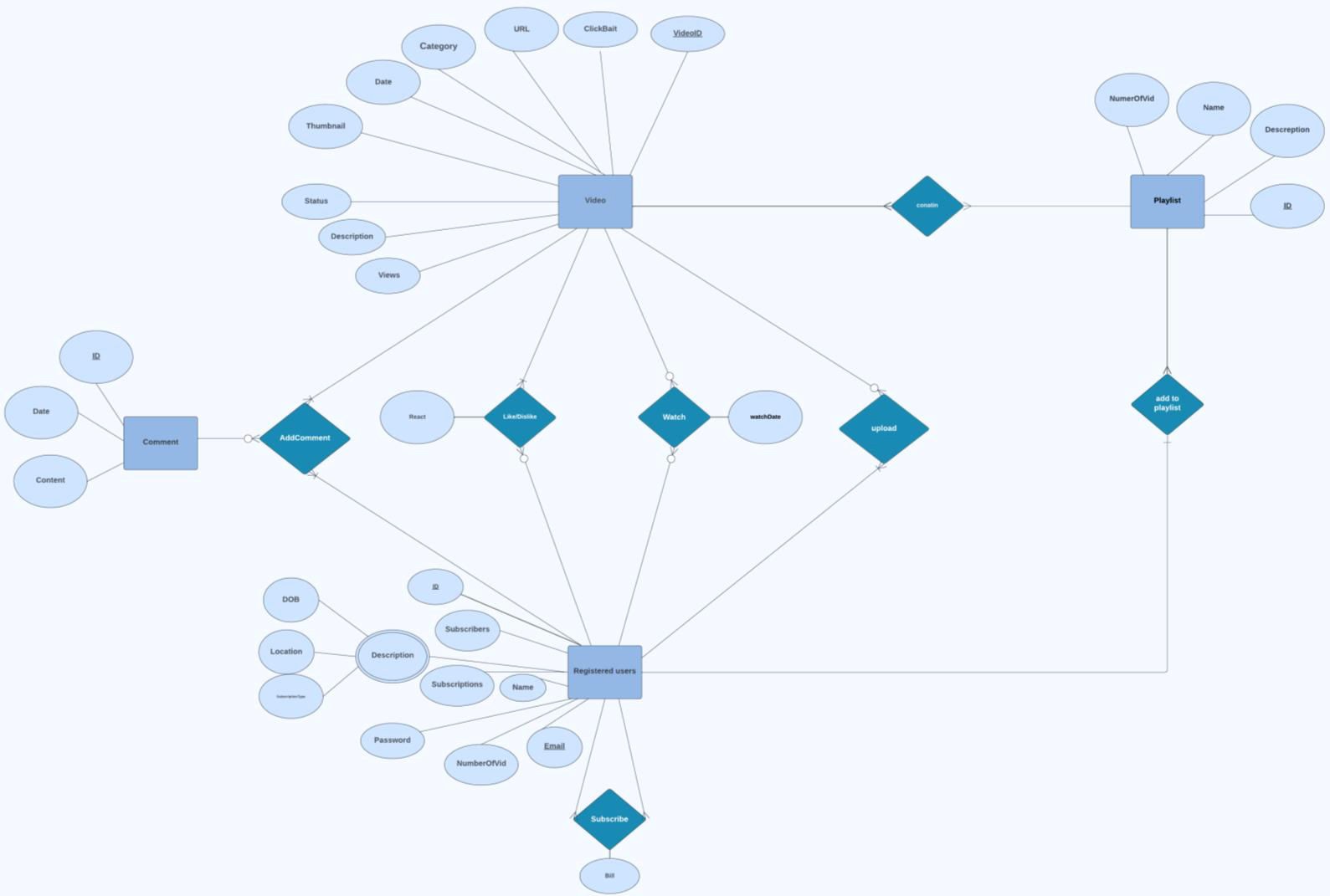
Subscribtion:



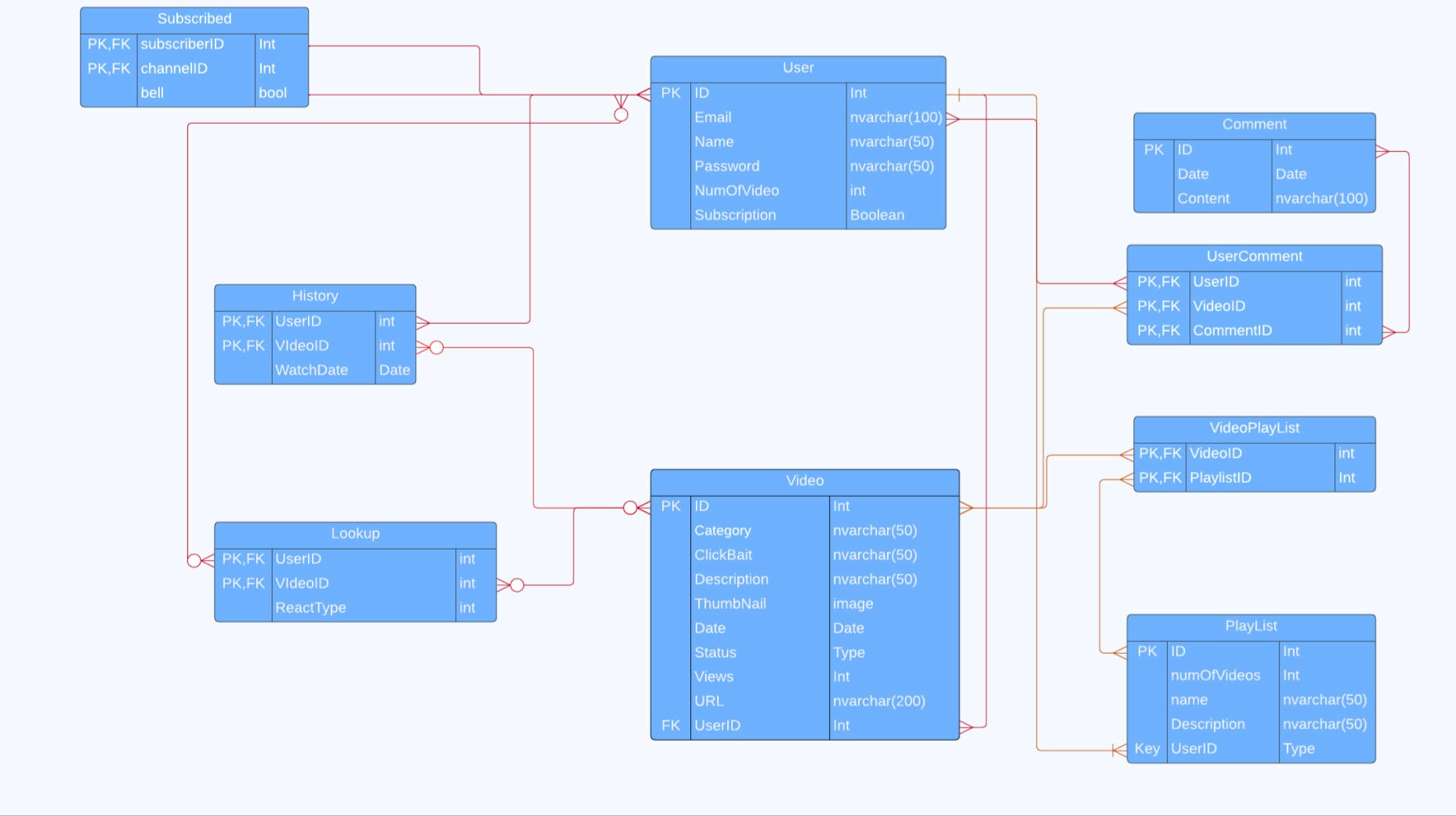
Share:

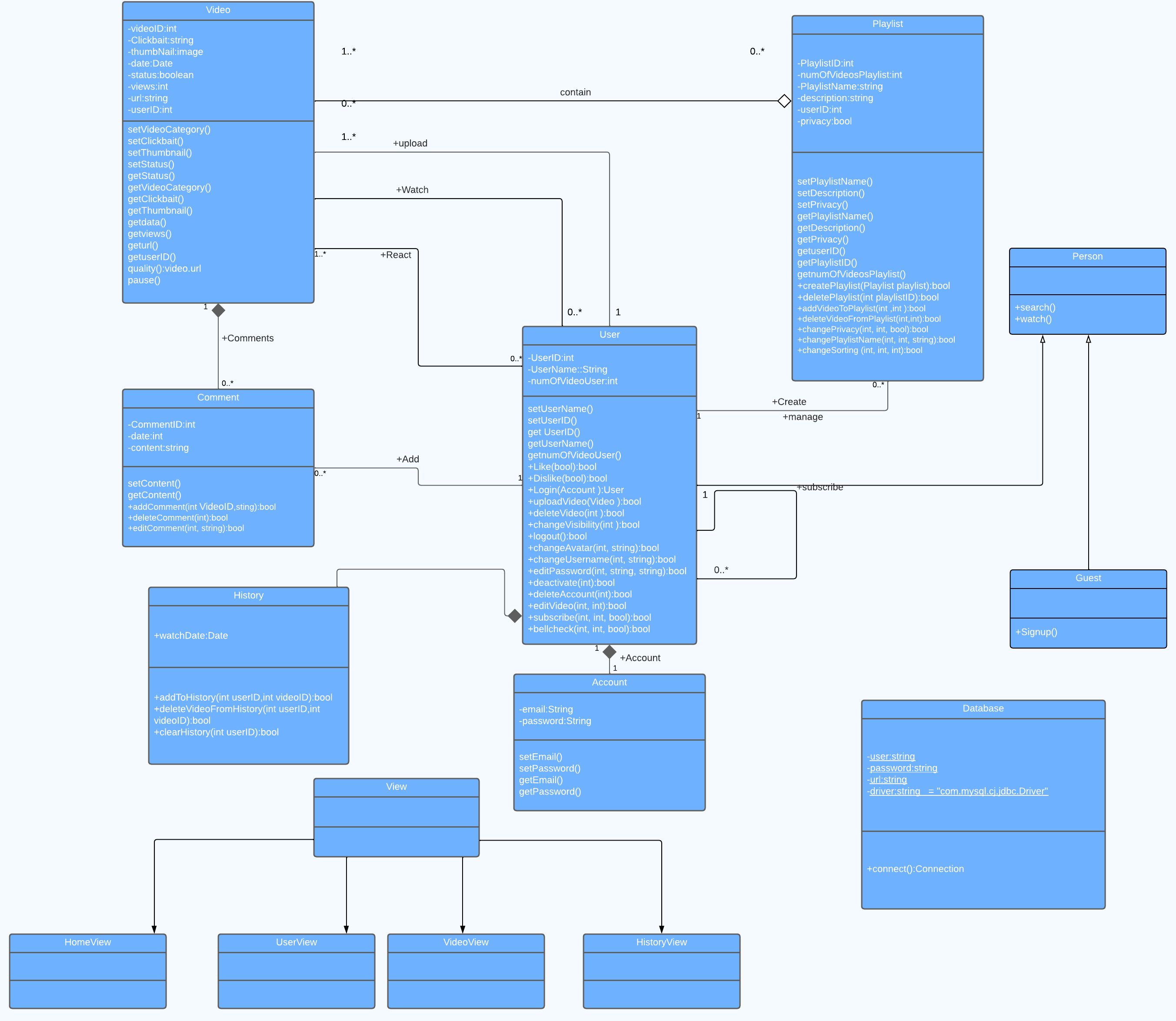


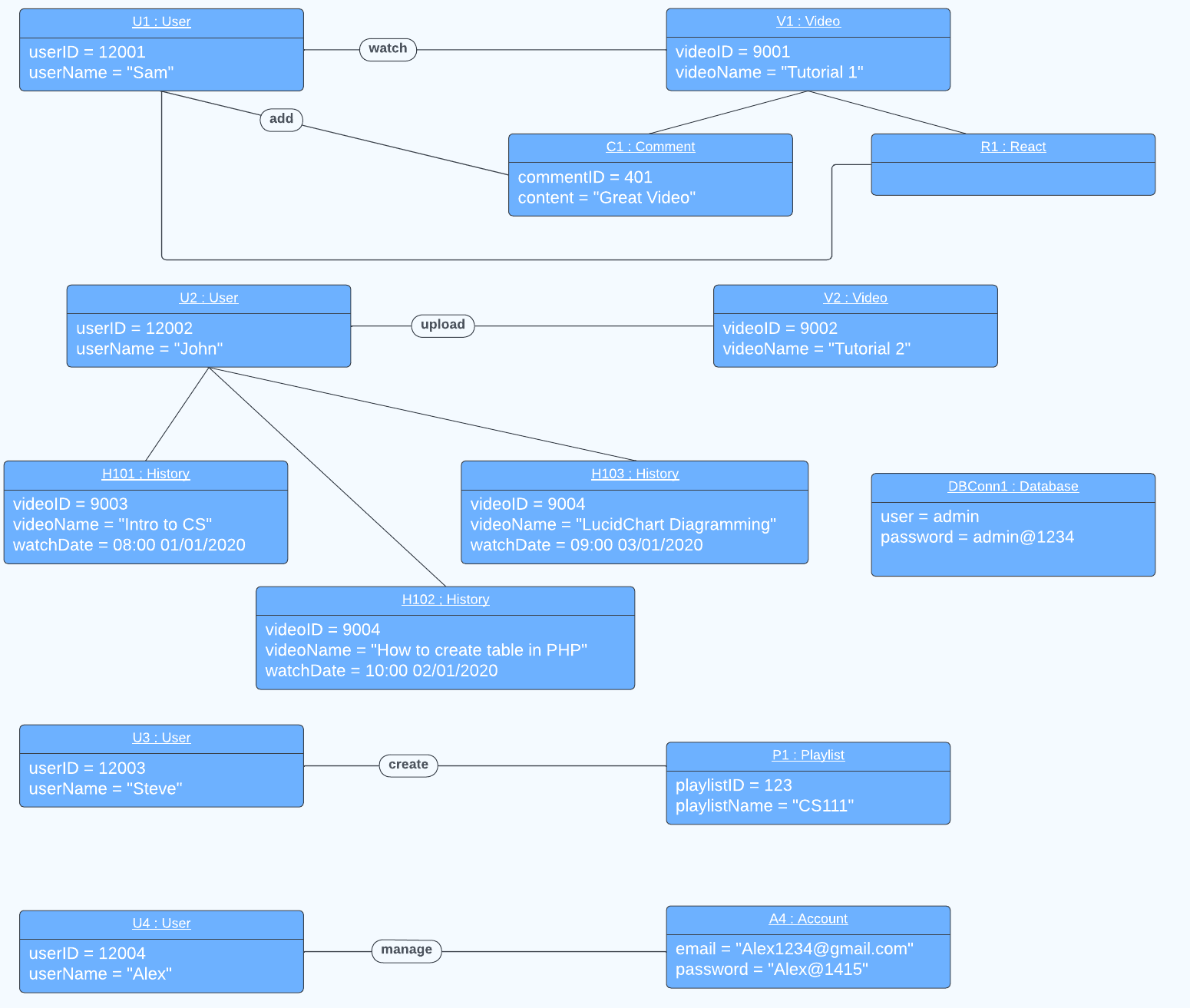
*ERD:*



*TABLES:*

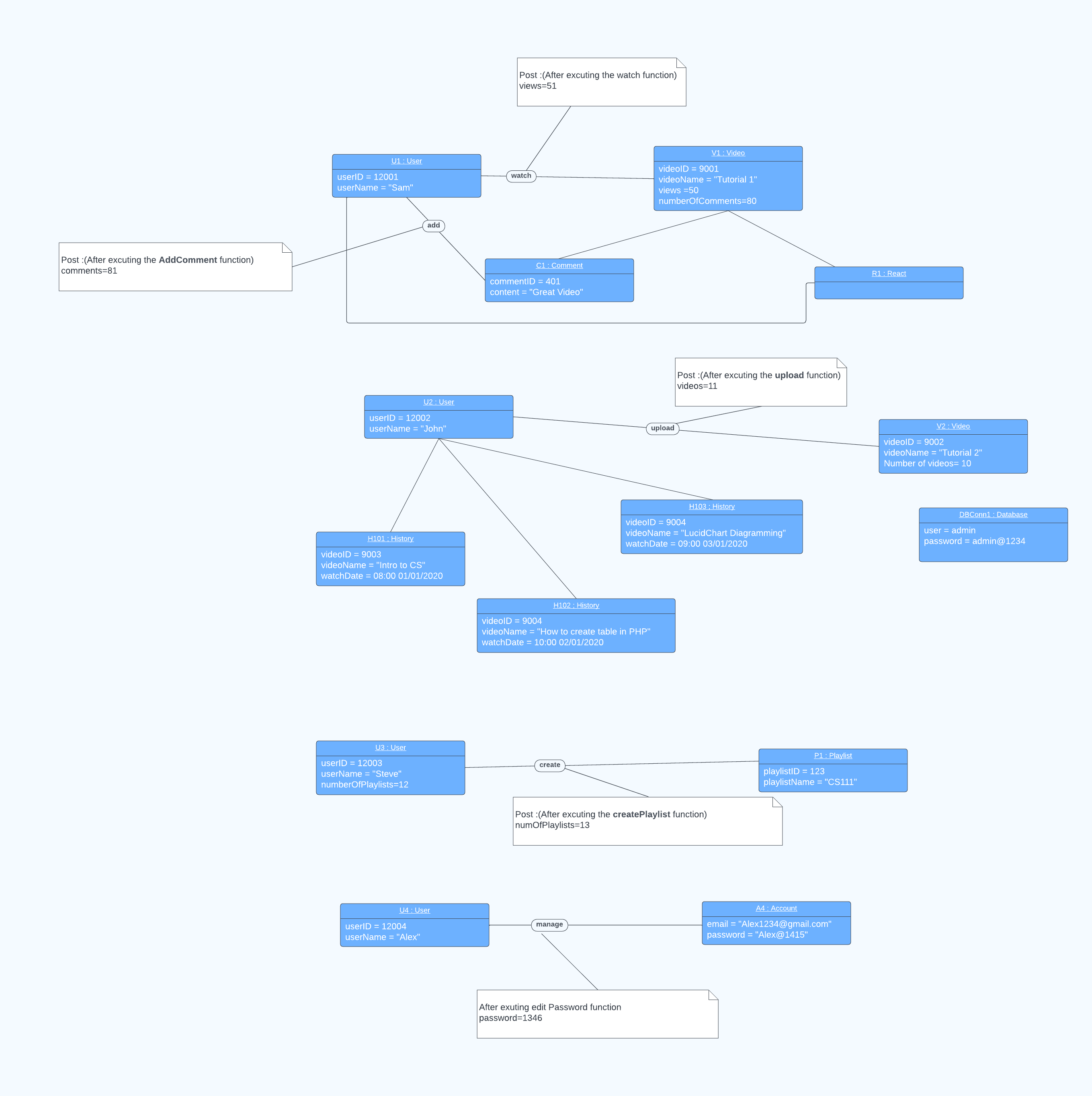


*CLASS DIAGRAM:*

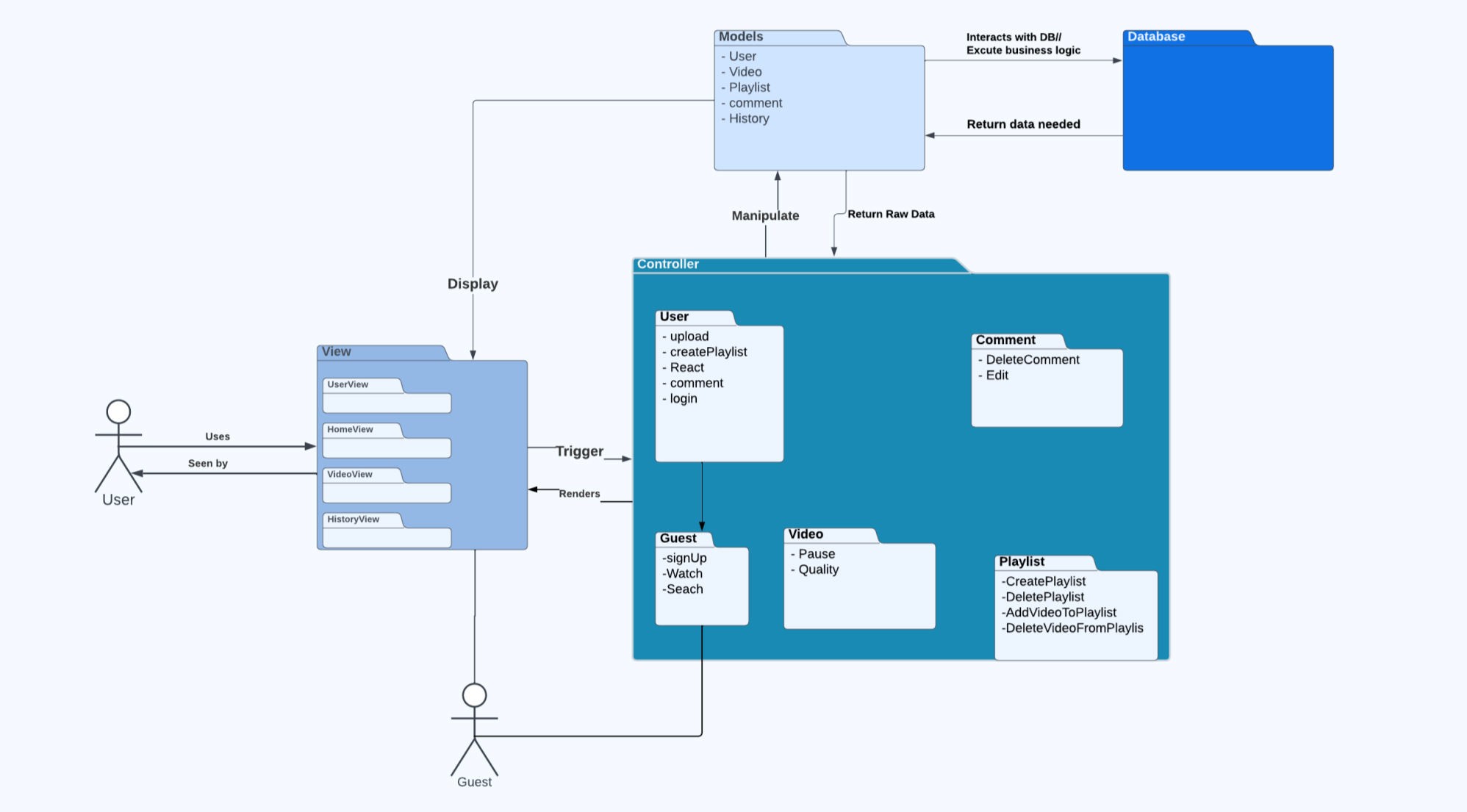


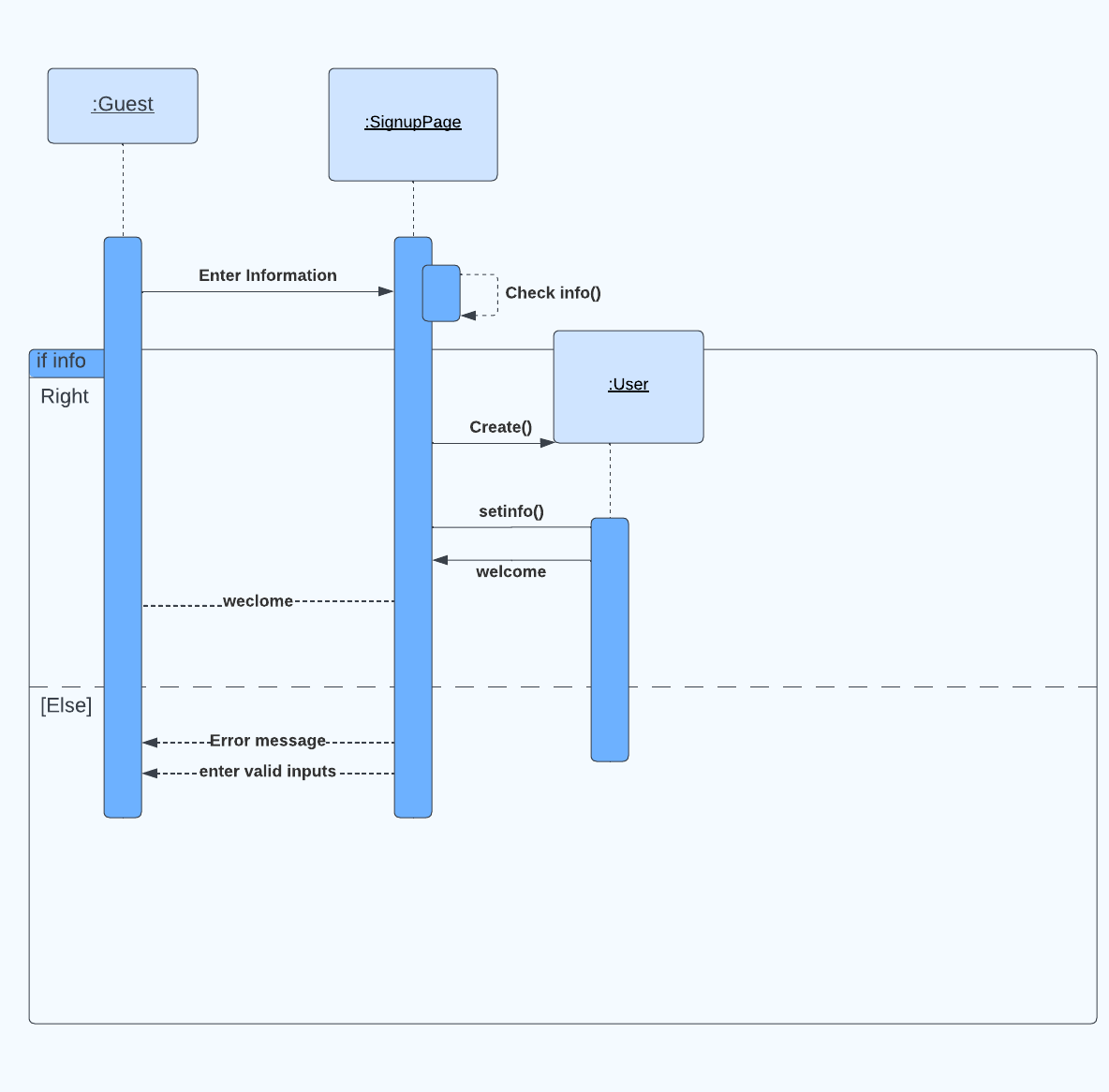
*OBJECT DIAGRAM:*

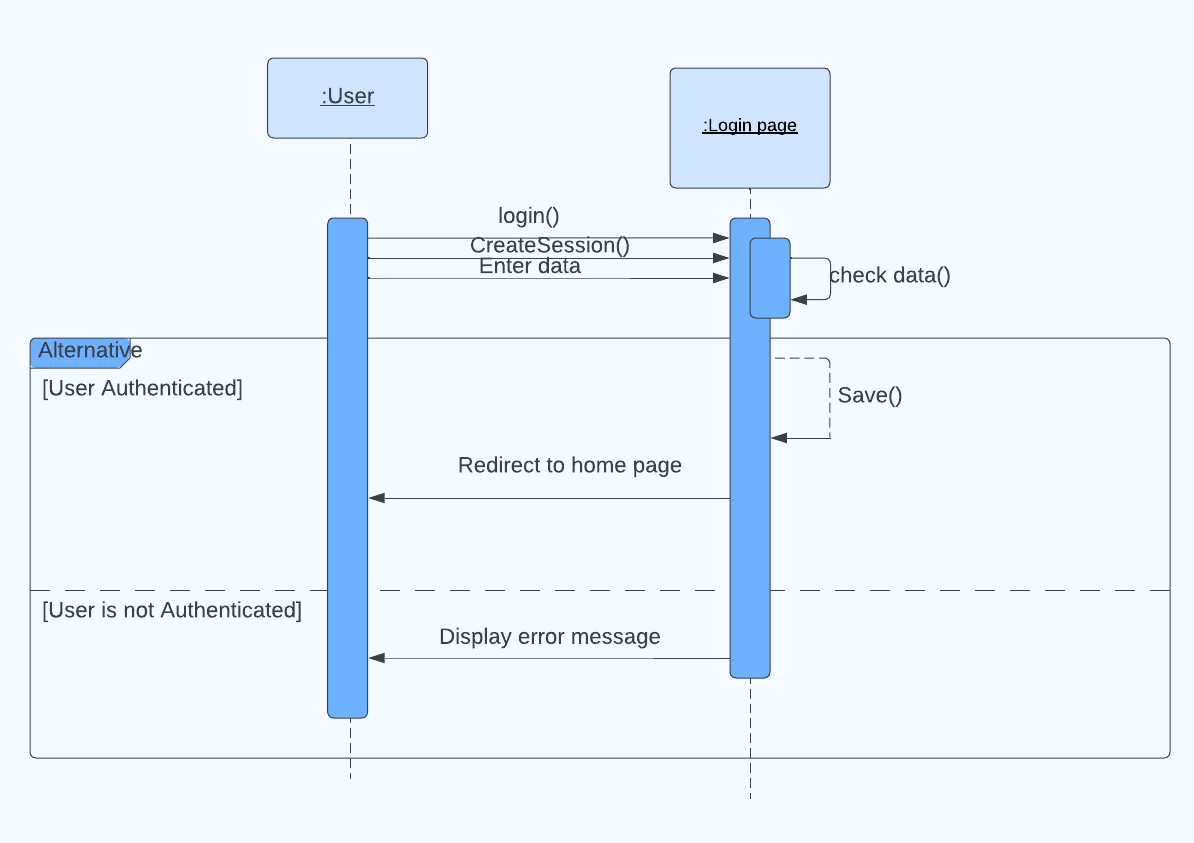
UpdatedObject Diagram

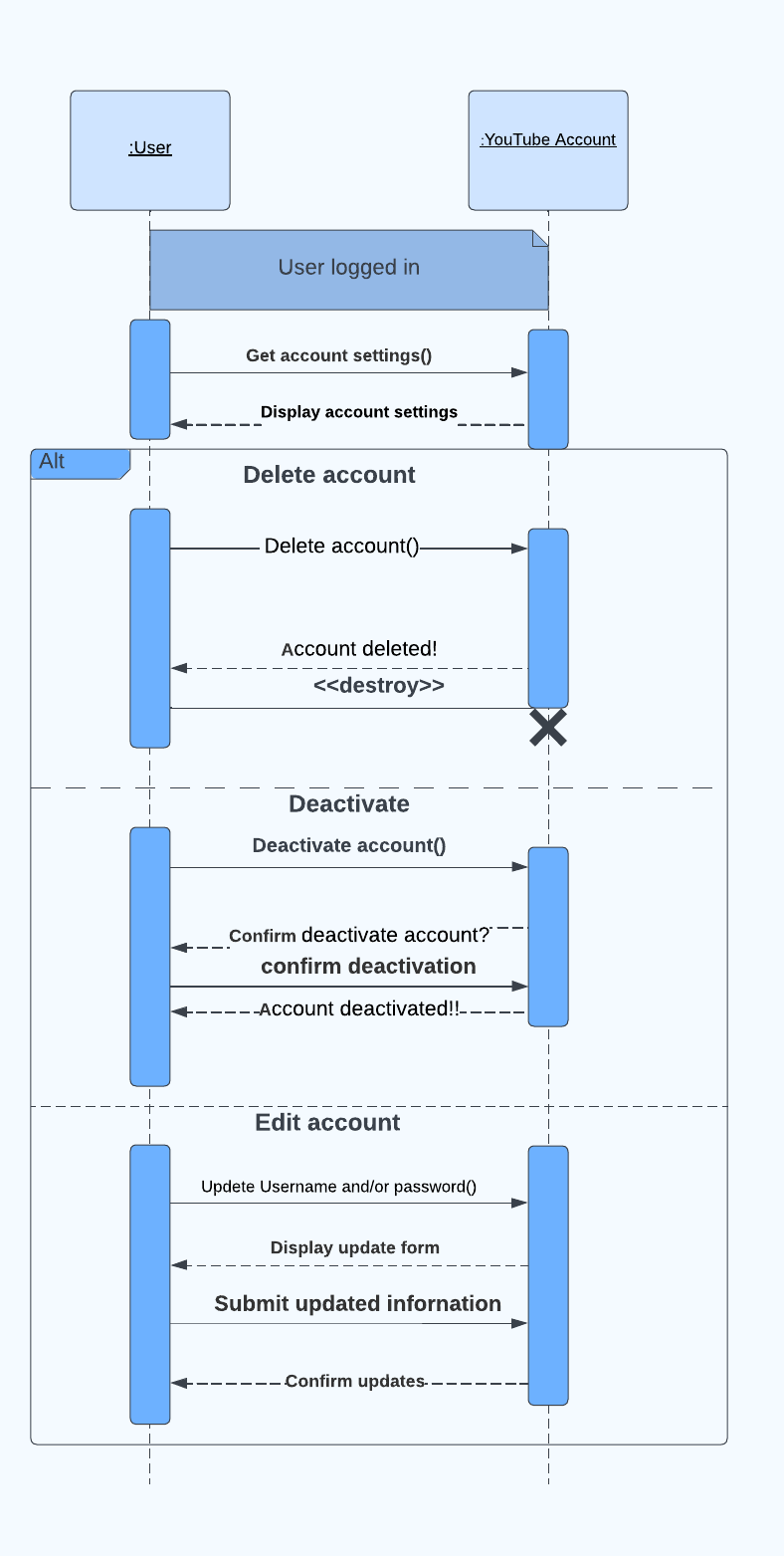


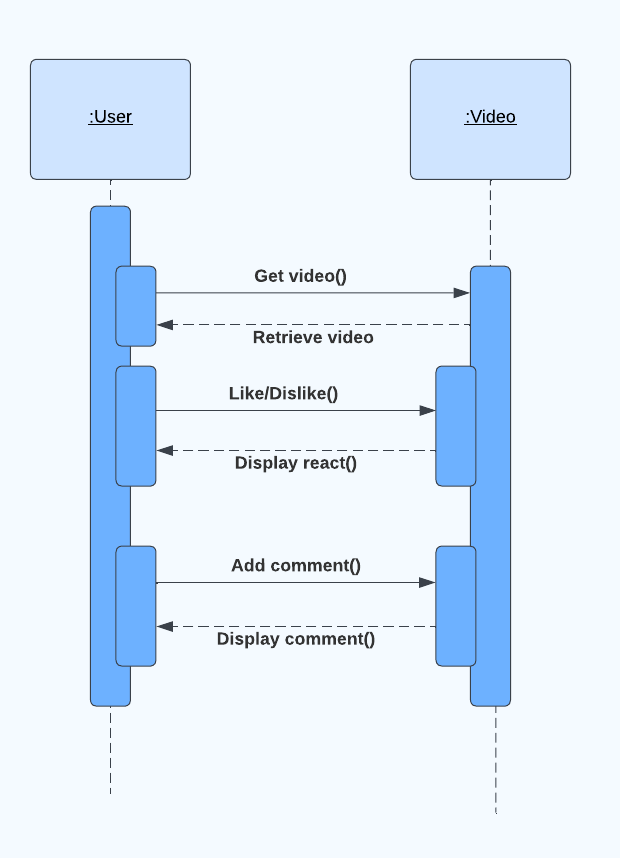
*PACKAGE DIAGRAM:*

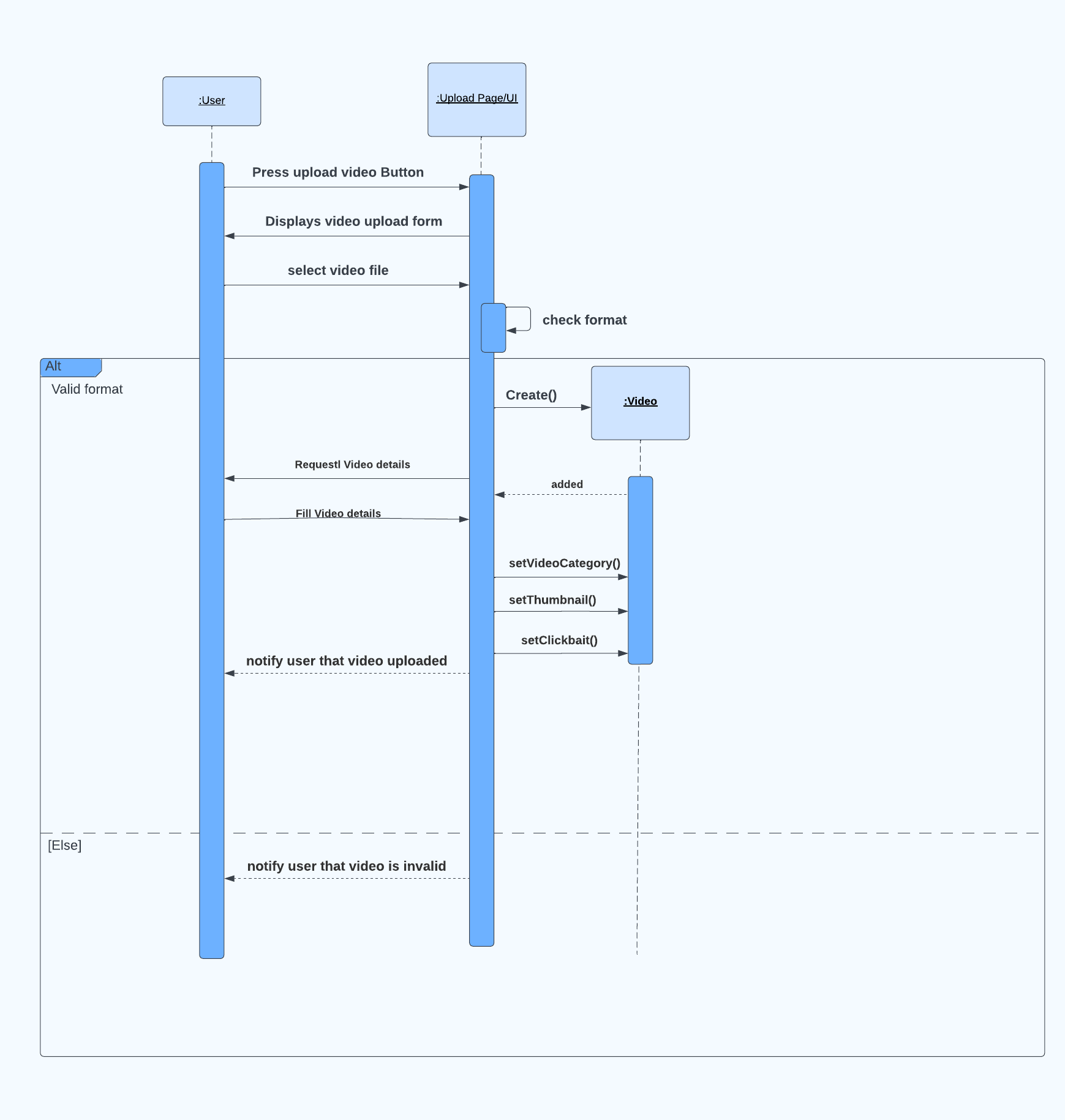


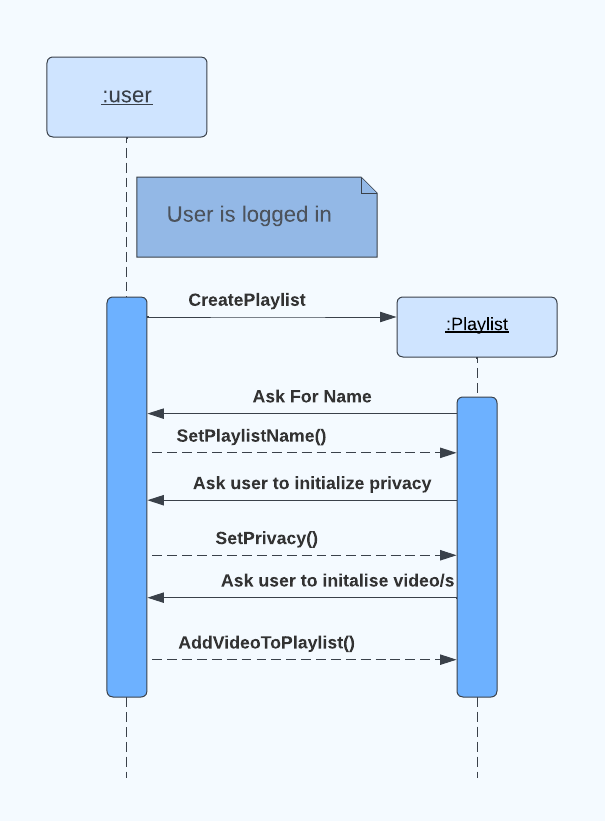
*SEQUENCE AND SYSTEM SEQUENCE DIAGRAM:*

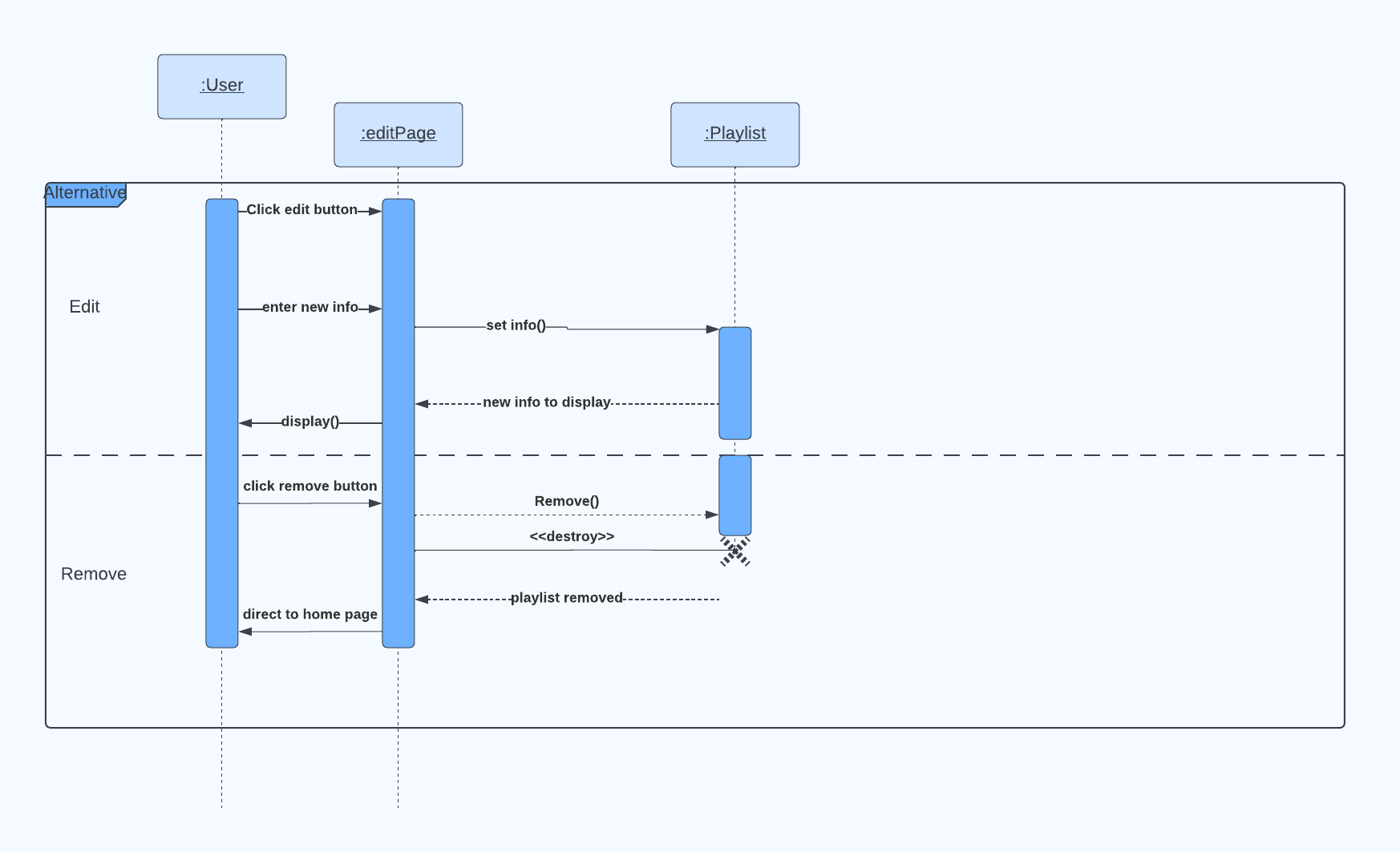


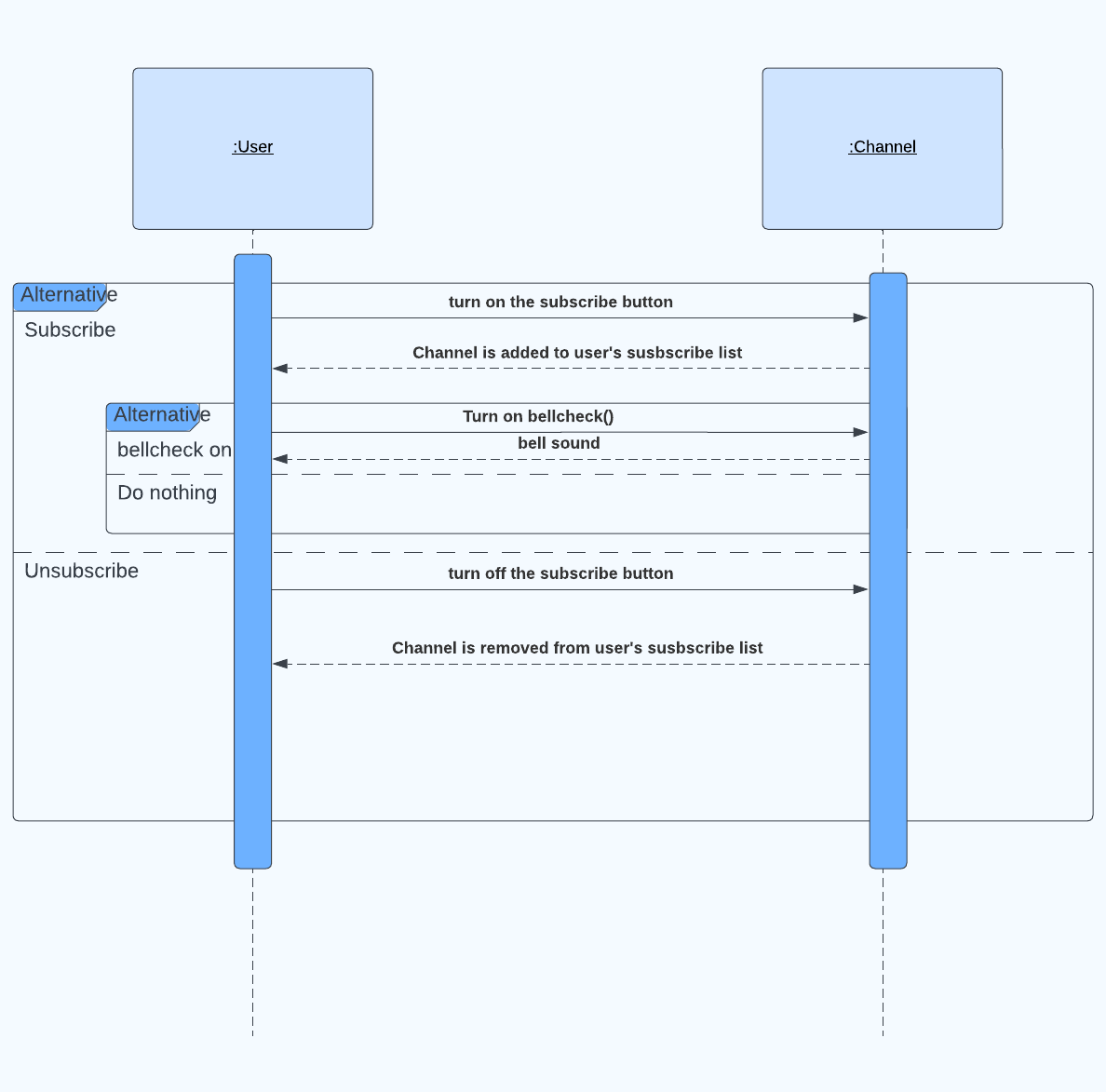


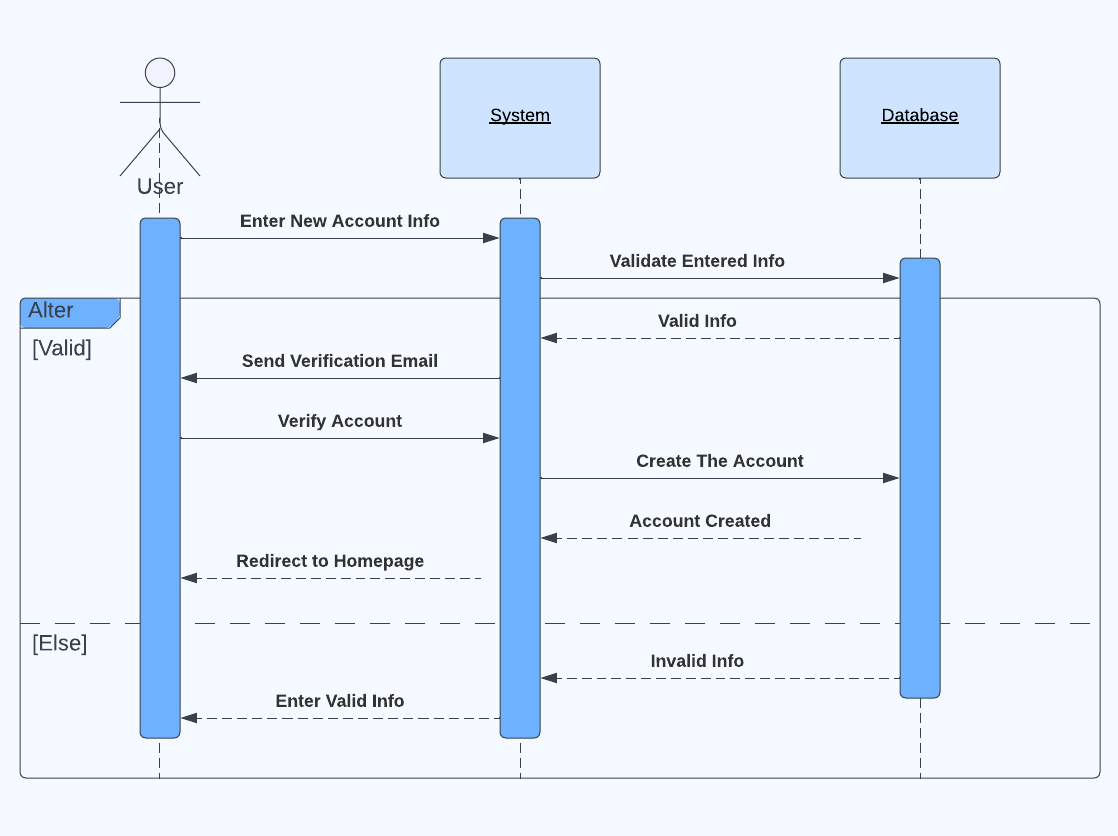










*SSD:*

