

SE-I COURSE PROJECT (PHASE 1 COVER SHEET)

Discussions Scheduled for Week 10 (more details will be announced later).

- Print 1 copy of this cover sheet and attach it to a printed copy of the documentation (SRS, ... etc.). You must also submit softcopies of all your documents (as PDFs); details will be announced later.
- Please write all your names in Arabic.
- Please make sure that your students' IDs are correct.
- Handwritten Signatures for the attendance of all team members should be filled in before the discussion.
- Please attend the discussion on time (*announced separately*), late teams will lose 5 grades.

Project Name: A Video Sharing Application [inspired by YouTube]

Team Information

	ID	Full Name
1	202001028	هبة اسماعيل محمد عدلي
2	202000472	ضحي حسانين محمد الصاوي
3	202000577	علياء زغلول السيد
4	202000534	عبدالرحمن محمد يوسف زهران
5	202000096	ادهم احمد عيد محمد
6	202000475	ضياء احمد عبدالمنعم

Grading Criteria:

10 Items	Grade	Notes
1. Functional Requirements	1	
2. Non-Functional Requirements	1	
3. Use-Case Diagram(s) including general use-cases for the system, and the detailed use-cases description	2	
4. System Architecture – including applied Architectural Pattern(s)	1	
5. Activity Diagrams	2	
6. Database Specification (ERD, Tables)	2	
7. Class Diagram (Interfaces, Classes, Relations) – An initial version based on the requirements and Use-Case/Activity diagrams.	2	
8. Object Diagrams (Including object diagrams that illustrate the preconditions and the post-conditions of selected functions)	1	
9. Package Diagram(s)	1	
10. Sequence Diagrams including System Sequence Diagrams (SSDs)	2	

Project 21

A Video Sharing Application [inspired by YouTube]

The application includes functions such as (1) Users can search for and watch videos, (2) A user can create a personal channel, (3) A user can upload videos to his/her channel, (4) A user can Like, Comment on, and share other videos. (5) Users can subscribe/follow other channels and users. (6) Users can create playlists to organize videos and group the videos together.

Functional & non-functional requirements:

-user & system requirements-

Functional:

1.	Users shall be able to login the system 1.1 user will enter his name and password. 1.2 If the username and password is true system work start home page. 1.3 If the username and password is not true system displays error message.
2.	visitor shall be able to create an account 2.1 user requests system to create a new account 2.2 system check if the account is not exist 2.3 if account is already existing system displays message error 2.4 if account in not exist system will create new account and add it in DB.
3.	Users shall be able to upload videos in the system when they login 3.1 user requests system to create a new account 3.2 system check if the video exceed 250MB. 3.3 if the video exceed 250MB, system displays – can't upload this video- 3.4 if video not exceed 250MB, system will upload this video and add it in DB.
4.	Users can add comment to videos 4.1 when user wants to add comment, he will just write this content in comment's place and click on -Add comment- button. 4.1 system will add this comment and display it
5.	Users or visitors shall be able to search video and watch it 5.1 user/visitor can search by write the title of video which he wants to watch it in search bar. 5.2 system will display list of videos which related with their searching. 5.3 users/visitors select specific video and click on it to watch
7.	Users can like / dislike videos 7.1 when user wants to like/dislike any video, he clicks on like icon 7.2 system will increase the number of likes/dislike on this video.

8.	<p>Users can create a channel</p> <p>8.1 when user wants to create a new channel ,he click on button create channel</p> <p>8.2 system check if user is under 18 years or not.</p> <p>8.3 if user is under 18 years ,system will display can't create this channel</p> <p>8.4 if user older than 18, system will check if user has 3 channel or not</p> <p>8.5 if user has 3 channel, system not allowed him to create a new one.</p> <p>8.6 if user' channel less than 3, he can create one by enter its name then system will create this channel.</p>
9.	<p>Users can create playlist to organize videos</p> <p>9.1 user can create a new playlist by click on create playlist button</p> <p>9.2 user enter name of this playlist</p> <p>9.3 system will create this playlist, then user can add videos in it.</p>
10.	<p>The user can report the video that violates the ethics</p> <p>10.1 when user watch video, if this video violates the ethics, user can report on this video.</p> <p>10.2 if number of reports on this video exceed 1000 report, system will remove this video.</p>
11.	<p>User can add videos to watch later list</p> <p>11.1 when user watch video and he want to add it in watch later list, he just click on -add to watch later- icon.</p> <p>11.2 system will add this video in watch later list.</p>
12.	<p>The user must be able to see his profile</p> <p>12.1 The system must display profile of the user.</p> <p>12.2 The user must be able to see his profile and edit it.</p>
13.	<p>The user must be able to edit the profile.</p> <p>13.1 The system must save the new information.</p>
14.	<p>The user must be able to change his password.</p> <p>14.1 The system must save the new password after user changed it.</p>
15.	<p>The user can see the previously opened videos from the history.</p> <p>15.1 system display the list of videos in history section of website.</p> <p>15.2 user can see all videos from this list.</p>

16.	<p>User can share video.</p> <p>16.1 when user click on share button, system displays all App which user can share this video on it</p> <p>16.2 user select the App, he wants to share video on it.</p>
17.	<p>User get notified of new uploads from the subscribed channel.</p> <p>17.1 if channel which user subscribe with it uploads a new video/ post , system will notified user</p>
18.	<p>user can delete the video</p> <p>18.1 user will select video which he want to delete it.</p> <p>18.2 then system will delete it from DB.</p> <p>18.3 The system after that shall display a message that -deleted successfully-</p>
19.	<p>user can delete the comment.</p> <p>19.1 user will select comment which he want to delete it.</p> <p>19.2 then system will delete it from DB.</p> <p>19.3 The system after that shall display a message that -deleted successfully-</p>
20.	<p>user can unsubscribe the channel</p> <p>20.1 user can unsubscribe by click on unsubscribe icon.</p> <p>20.2 number of subscribers in this channel will decrease one.</p>
21.	<p>user can add title on video.</p> <p>21.1 when user upload a new video, he can add title of this video.</p>
22.	<p>User can add a thumbnail to the video.</p> <p>22.1 user can add a thumbnail image to be the background of the video that the user will upload.</p>
23.	<p>user can filter the video.</p> <p>23.1 when system shows to user all videos resulting from his searching, user can filter these based on date of upload.</p>
24.	<p>user can delete his account</p> <p>24.1 system delete this account from DB.</p> <p>24.2 System display a message that -deleted successfully-.</p>

25.	User can add Ads on his video. 25.1 after user select video to upload, he can add Ads on this video but he can't add more than 5 Ads.
26	user can complete watch the video where he stopped in last time. 26.1 The system keeps the last time the user stopped watching the video 26.2 if the user goes back to watching the video again, the system opens for him from the time he stopped the last time
27	User can send a feedback to the system. 27.1 user can send a feedback, then system will save it in DB. 27.2 user can see all the feedback of others
28	User can make a live on YouTube. 28.1 user can make a live and share it with others. 28.2 all other users can also participate in this live.
29	user can delete his channel 29.1 user select which channel he want to delete it. 29.2 System delete this channel 29.3 System display a message that -deleted successfully-.
30	The user can earn from the channel. 29.1 If the number of subscribers in the channel reaches 1000 subscribers and the number of hours is 4000 hours, the system will start calculating the profits for the user 29.2 For each Ad, the user will earn \$10

Non-functions:

performance requirements [BR]

1.	each request shall be processed within 3 seconds
2.	A system must support 10000 users at the same time

Usability Requirements [AR]

3.	The system shall be easy to use on the first attempt by a member of the public without training
-----------	---

Safety and security requirements [SSR]

4.	The database should be secured from SQL injection to prevent leak or loss information
5.	Hashing technology should be used to handle secure login for users
6.	The visitor can only watch the video and search for it.
7.	the system not allowed user to create account until the user creates a strong password.
8.	After a certain number of login attempts, a security system may lock an account to protect a user's information from potential hackers. To unlock their account, a user can typically call the company to verify their identity and set a new password.
9.	The system shall ensure that only authorized users are able to gain access.

Reliability Requirements [RR]

10.	The program should be reliable and provide catching of exceptions so that unintended results do not occur such system crashes or data validation failures.
------------	--

Cultural REQs.

11.	The language used in the interface should be formal and polite
-----	--

Look-and-Feel REQs.

12.	The system shall use only two colors.
-----	---------------------------------------

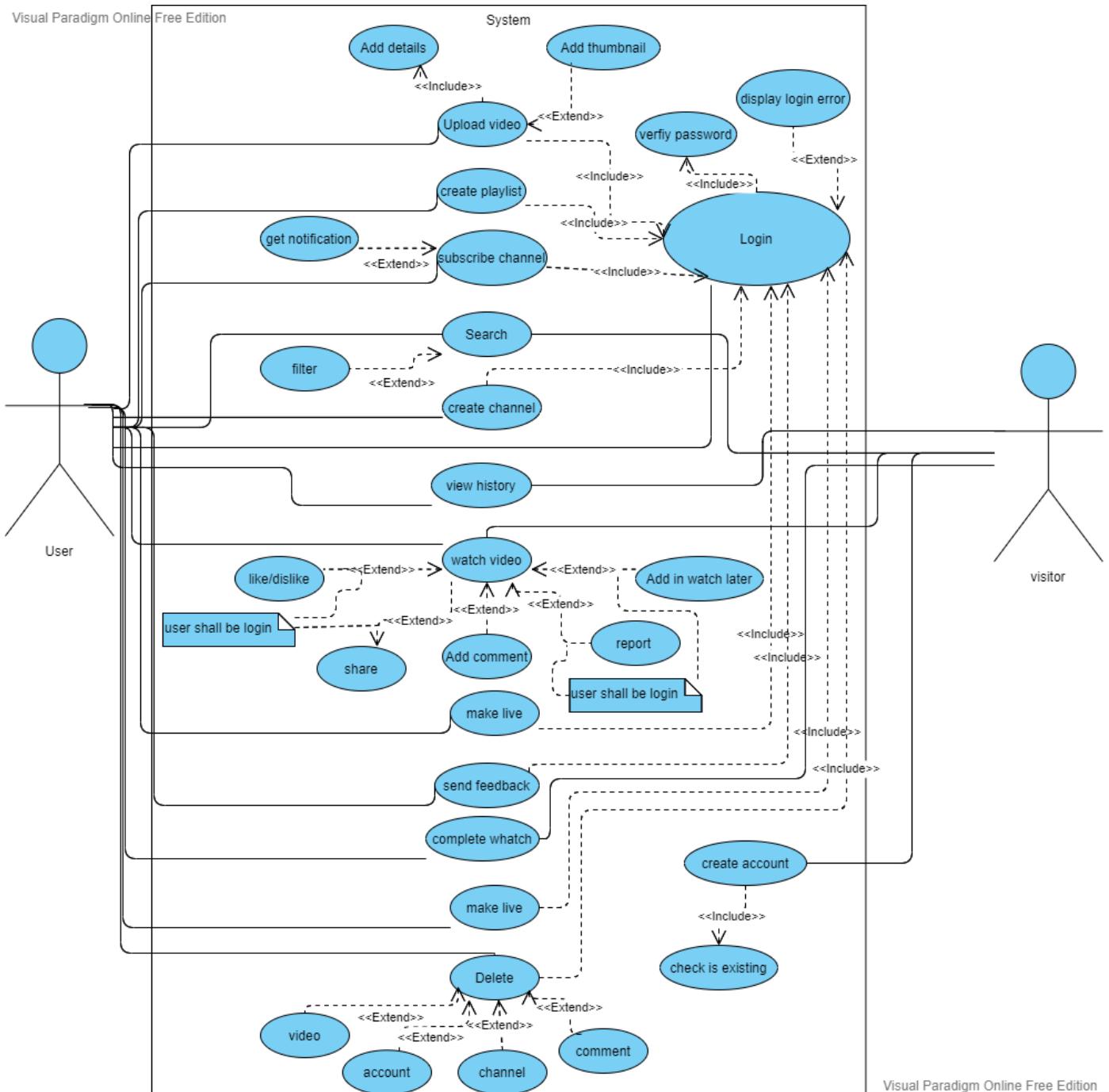
Legal REQs.

13.	If user's content violates the Copyrights, system will remove the content and send him an email to let him know.
14.	if number of reports on video exceed 1000 report, system will remove this video.

15.	The size of the video , which will be uploaded by the user shall not exceed 250MB.
16.	The user can add videos on his channel only, but he can't do that in the channels of other users.
18.	The user cannot create more than three channels
19.	if user gets 3 strikes within 90 days, his channel will be terminated
20.	The system can be run on all browsers
21.	The product shall be used in variable lighting conditions.
22.	The product shall conserve battery life.
23.	The number of ads on the video is not allowed to exceed 5 ads
24.	The user earns \$10 for each ad that appears while the video is playing

25.	While playing the video, only suggested videos appear below this video
26.	if user want to earn from YouTube, he should achieve 1000 subscribers in his channel and 4000 hours watching.

Use Case Diagram:



Use case description:

Create channel:

YouTube System/Create channel/SRS/1.1.0
Function: create a new channel
Description
User can create a new channel, and upload videos on it. Then others can subscribe on his channel.
Inputs: name of this channel. Source: name from user. Outputs: a new channel. Destination: User.
Main successful scenario : When user wants to create a new channel, he shall enter name of this channel, then system will check if he has 3 channels or not, if he has 3 channel, system not allowed him to create a new one, If not system will create a new channel.
Requirements
User has account.
Pre-condition
The user shall log in first, and hasn't more than 3 channels.
Post-condition
User has a new channel.

Upload Video:

YouTube System/Upload video/SRS/1.1.0
Function: Upload a new video.
Description
User can upload video on his channel or his account, and other can like/dislike this video. User can also see the number of video views.
Inputs: selected video.
Source: From the user's device.
Outputs: video uploaded.
Destination: video upload on system.
Main successful scenario : When user wants to upload video, he shall select the video from his device ,the system checks if video exceed 250MB or not. If video exceeds 250MB system can't upload it. If not system upload this video on user's channel successfully.
Requirements
User has account.
Pre-condition
The user shall log in first.
Post-condition
User has a new video on his channel.

Subscribe in channel:

YouTube System/ Subscribe in channel /SRS/1.1.0
Function: subscribe in channel
Description
User can subscribe in channel , then if channel upload a new video or post system will notified user what is the new.
Inputs: click on subscribe button.
Source: From user.
Outputs: Notification that the user has subscribed to the channel.
Destination: user.
Main successful scenario : User clicks the subscribe button in a channel, the system will allow the user to subscribe to this channel and notify him of any new uploaded video or post in this channel.
Requirements
User has account.
Pre-condition
The user shall log in first.
Post-condition
User subscribe in this channel successfully.

Create playlist:

YouTube System/ Create playlist /SRS/1.1.0
Function: create playlist.
Description
User can create a new playlist, then he can name it, and add videos in it.
Inputs: name of playlist. Source: From user. Outputs: new playlist is created. Destination: user.
Main successful scenario : When user wants to create a playlist, he shall name this list, then system will created it successfully, after that he can add the videos in this list.
Requirements
User has account.
Pre-condition
The user shall log in first.
Post-condition
User creates a playlist successfully.

Search for a video:

YouTube System/ Search for a video/SRS/1.1.0
Function: Search for a video
Description
User/visitor can search for a video, then system will display to him all related videos with his searching, then user can select any video to watch it.
Inputs: name of video which he wants to watch it. Source: user/visitor enters name of video in search bar. Outputs: all related videos with his searching Destination: user/visitor.
Main successful scenario : User will enter name of video in searching bar, then system will display to him all videos related with his searching, then user can filter videos based on date of uploaded.
Requirements
None.
Pre-condition
None.
Post-condition
System display all videos which related with his searching.

Watch video:

YouTube System/ watch video /SRS/1.1.0
Function: watch video.
Description
After user selects a video to watch, system will play it for user to watch it.
Inputs: click on icon to watch video.
Source: From user.
Outputs: video will play.
Destination: display to user.
Main successful scenario :
When user select video to watch , system will play this video to user can watch it, then system will increase number of views one, And if user (likes/dislikes, adds comment,.. etc.) All these reactions will be added and updated by the system in DB.
Requirements
None.
Pre-condition
None.
Post-condition
User watch video.

User can earn from videos:

YouTube System/ earn from videos /SRS/1.1.0
Function: earn from videos
Description
User can earn money from every video he upload that reaches the target on his channel .
Inputs: video name , his channel name , no. of subscribers , no. of hours watching
Source: From user.
Outputs: calculates the money he earned from that video
Destination: display to user.
Main successful scenario : When a user has a channel and achieves the target of 4000 hour of watching videos and gets 1000 subscribers, after that system allows the user to earn from the channel. if one of those requirements wasn't achieved, user can't earn from his channel.
Requirements
User has a channel User reached a 1000 subscribers User reached on his channel as a total 4000 hour of time watch The system checked his channel by system monetization policies
Pre-condition
The user shall log in first Have a channel Reach the all requirements
Post-condition
Calculates by system policies how much he will earn on each video

Delete a channel:

YouTube System/ Delete a channel /SRS/1.1.0
Function: delete user's channel from system.
Description
User can delete his channel with all videos that he uploaded it before
Inputs: his channel name
Source: From user.
Outputs: delete channel from the system
Destination: user.
Main successful scenario : When the user want to delete his channel , the system will check if his channel has videos so it will appear a confirm/warning message that all his videos will deleted too and if he refuse then nothing happens but if he confirm to this message it will delete first all the videos and then delete channel from the system.
Requirements
User has a channel
Pre-condition
The user shall log in first Have a channel Confirm to warning message
Post-condition
Delete his channel

Delete a video:

YouTube System/ Delete a video /SRS/1.1.0
Function: delete user's video from system.
Description
User can delete his video from his channel forever
Inputs: his video name
Source: From user.
Outputs: delete video from the system
Destination: user.
Main successful scenario : When the user want to delete a video from his account or his channel, he shall select which video he wants to deleted, then system deletes from DB successfully.
Requirements
User has a video on his channel/account.
Pre-condition
The user shall log in first Have a video Confirm to warning message
Post-condition
Delete his video

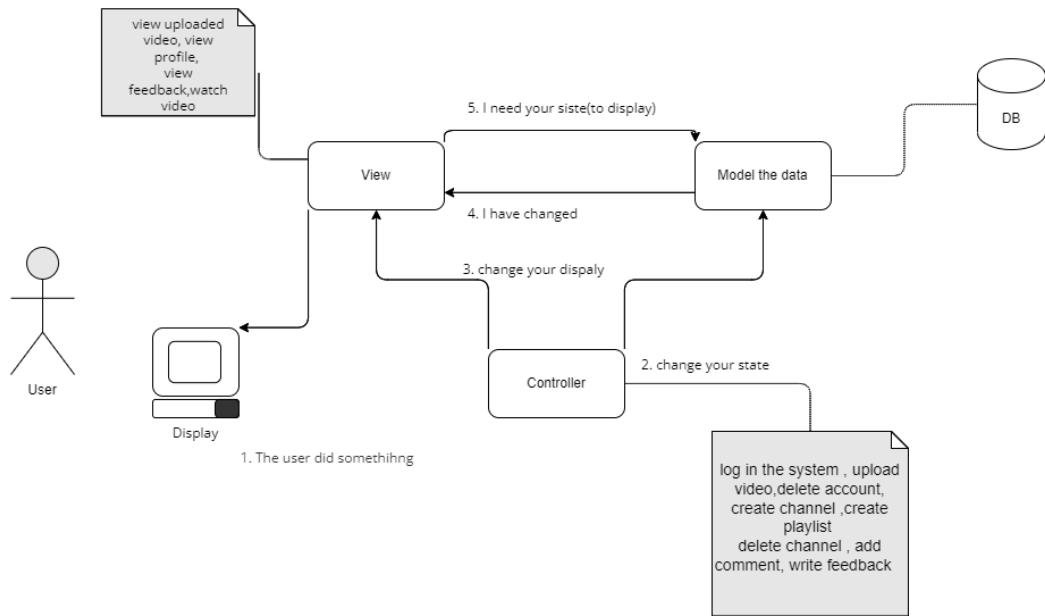
Delete an account:

YouTube System/ Delete an account/SRS/1.1.0
Function: delete user account from system
Description
User can delete his account from the system forever
Inputs: user name
Source: From user.
Outputs: delete an account
Destination: user.
Main successful scenario : When the user wants to delete his account , the system will show a warning message says that all your information will delete and you can't like the videos or share it without having an account if he confirm to this message his account will be deleted from the system database if he didn't confirm noting changes.
Requirements
User has an account
Pre-condition
The user shall log in first
Post-condition
Delete his account from the database system

System Architecture

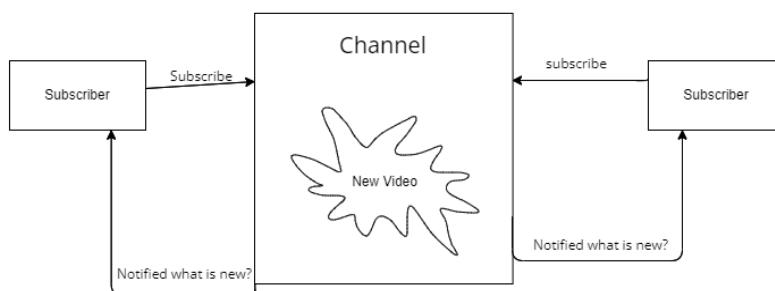
1)MVC -Model View Controller-

Visual Paradigm Online Free Edition

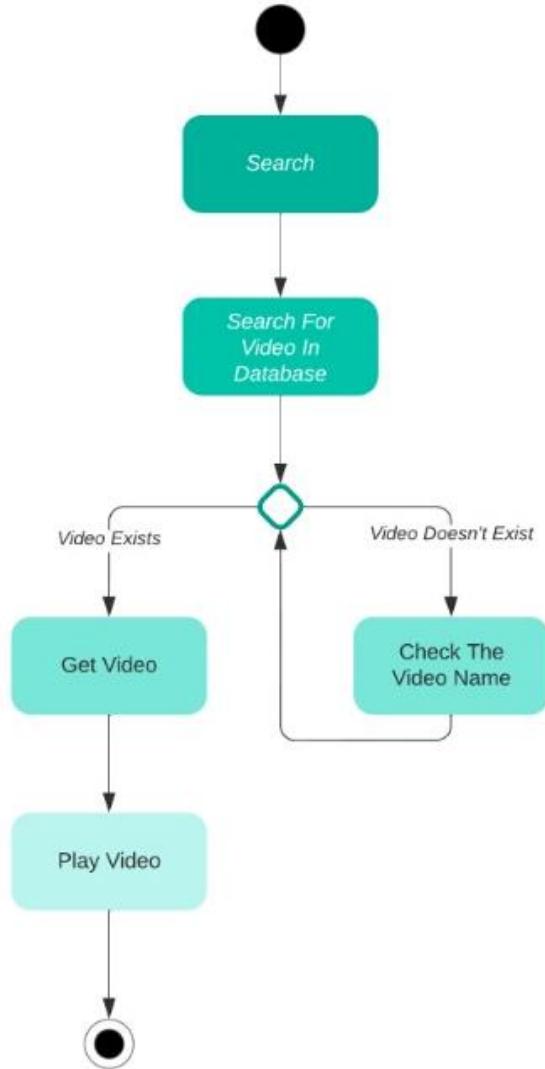


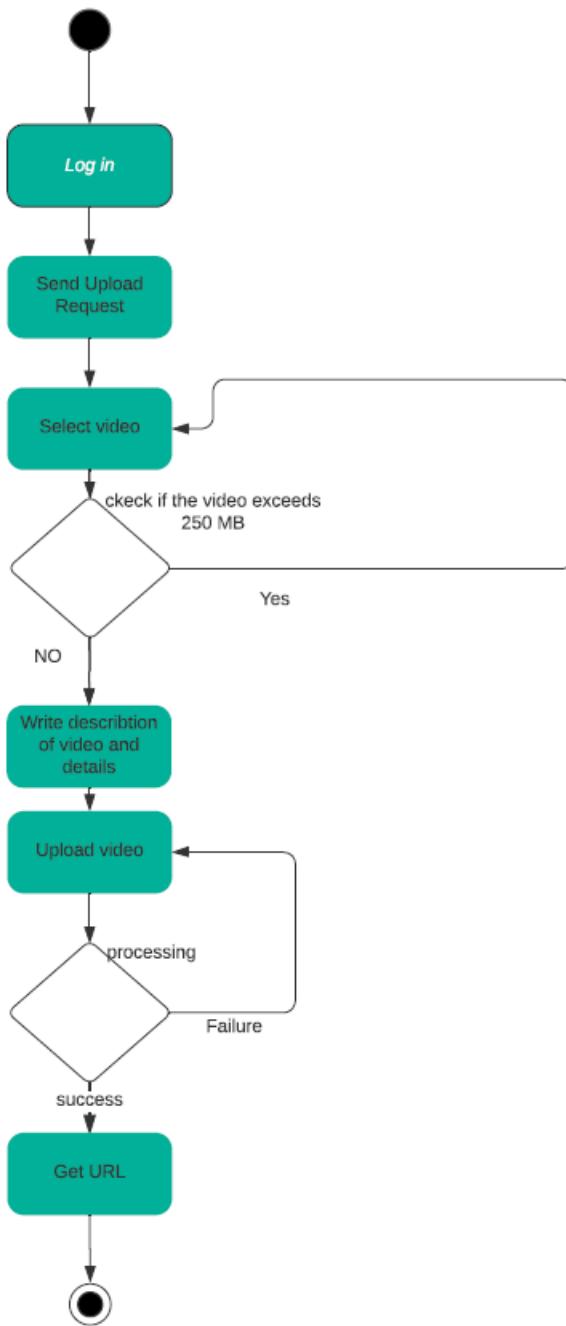
2) Notification / Implicit-invocation

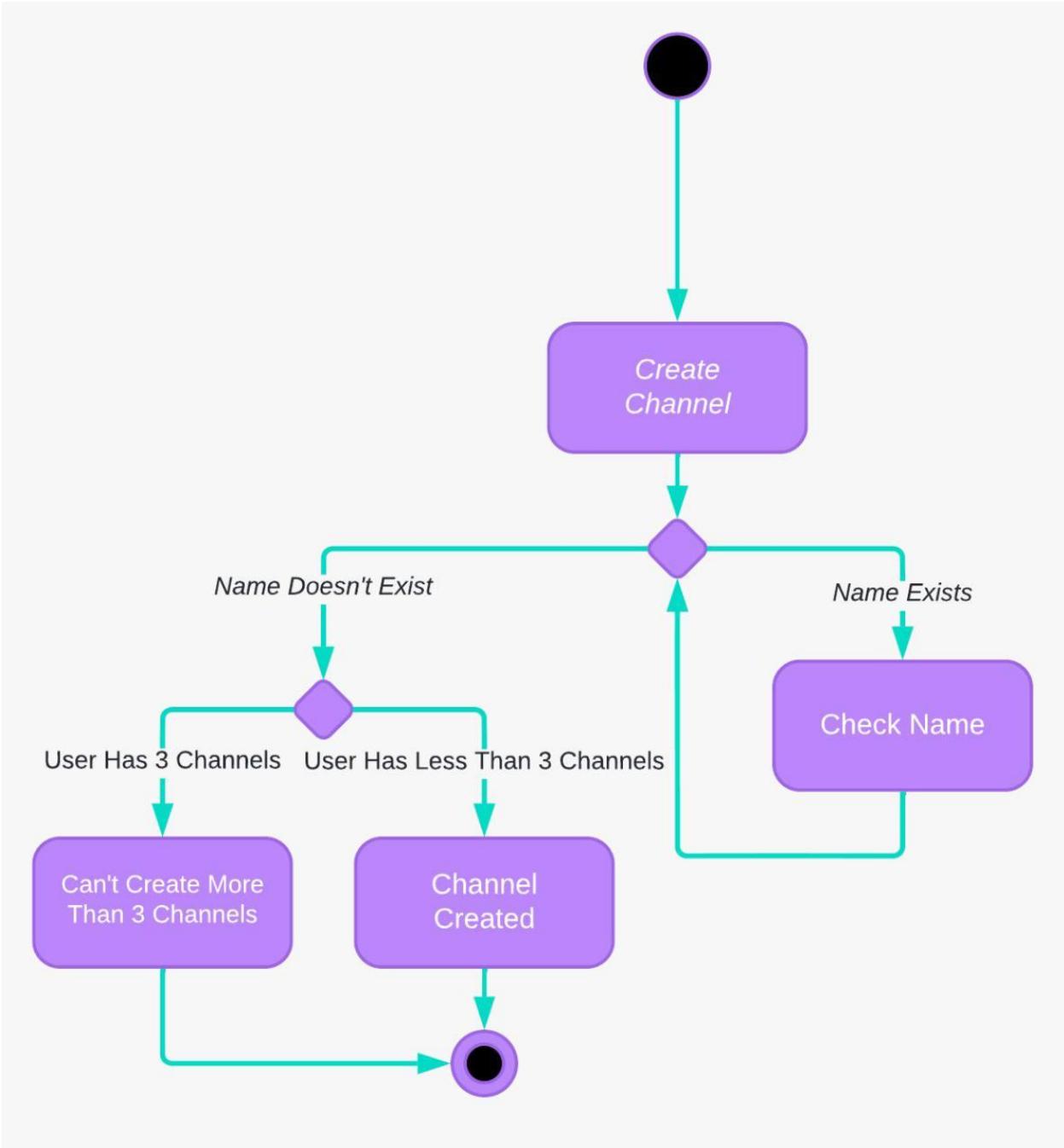
Users can register themselves with a channel to be kept notified whenever some particular event happens at the channel.

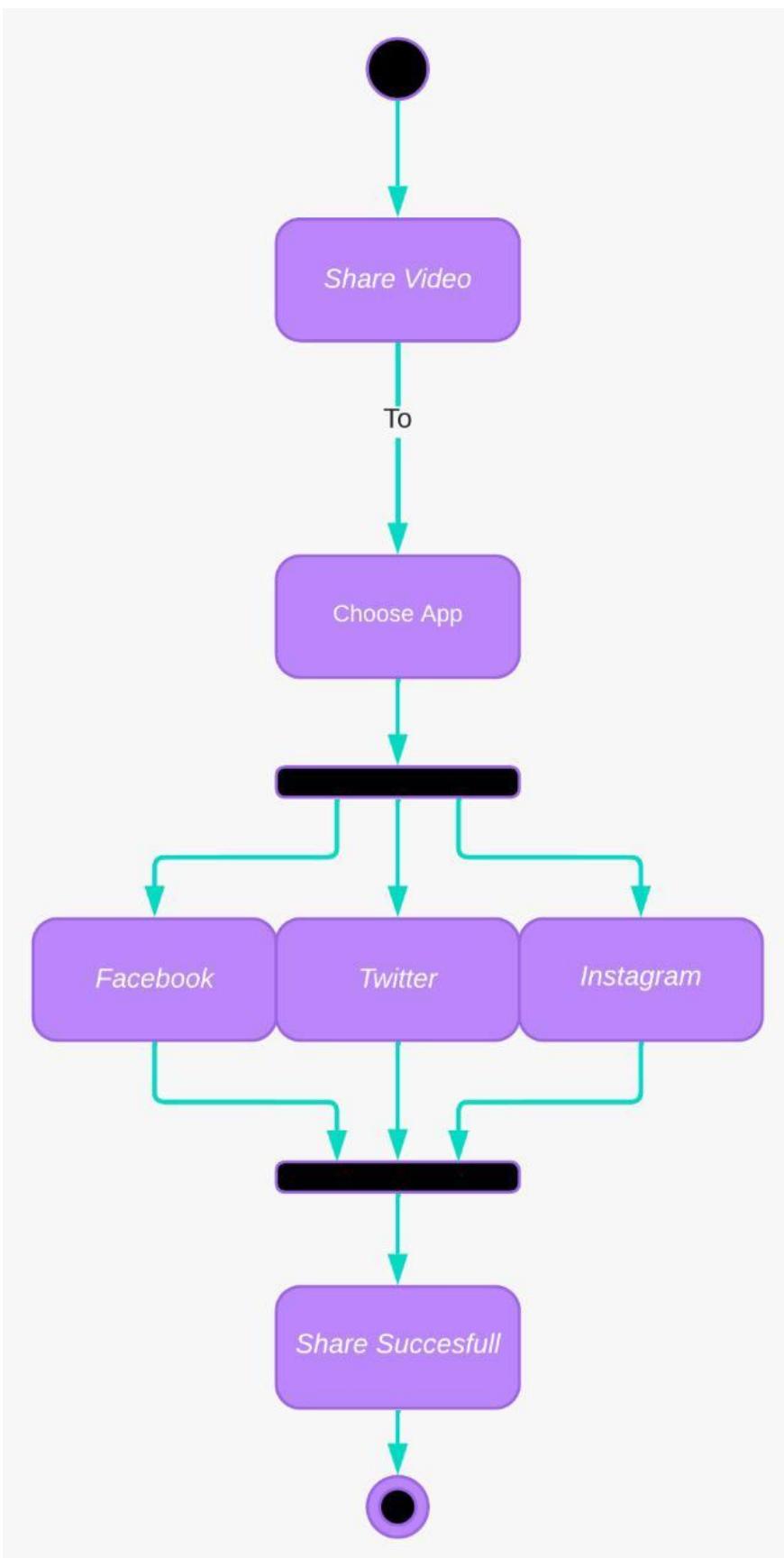


Activity Diagrams:



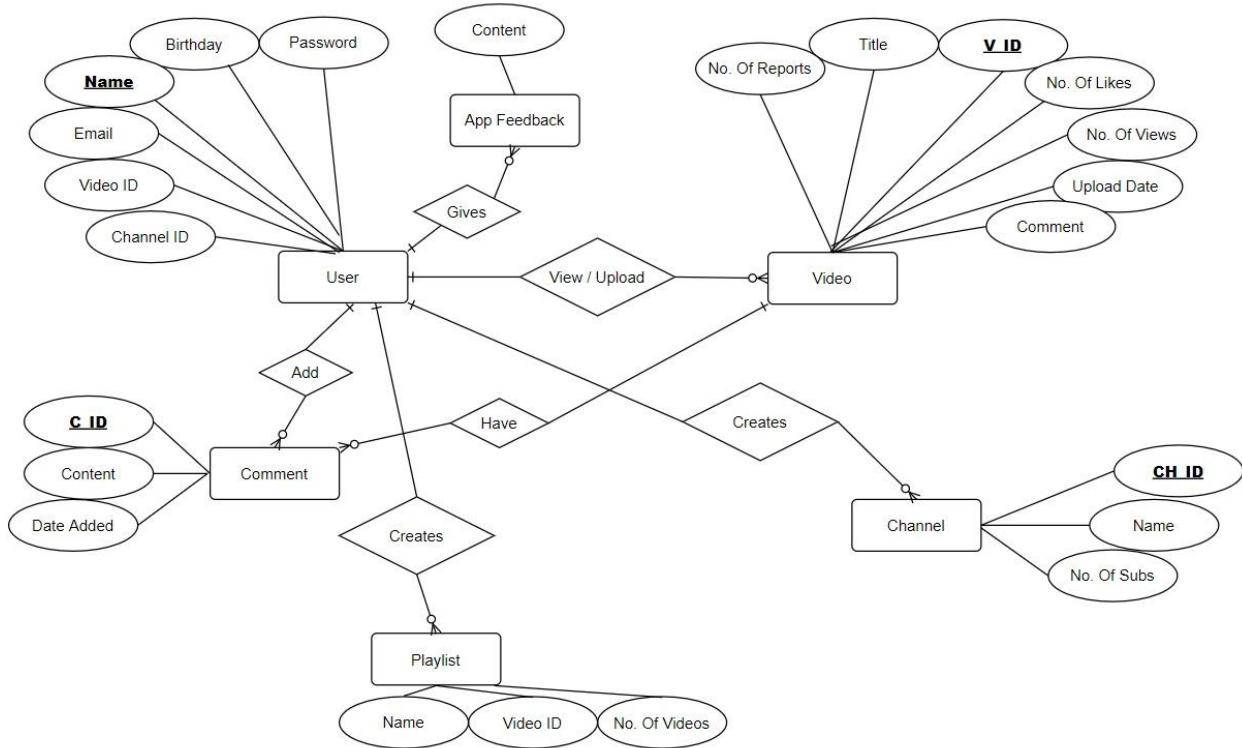




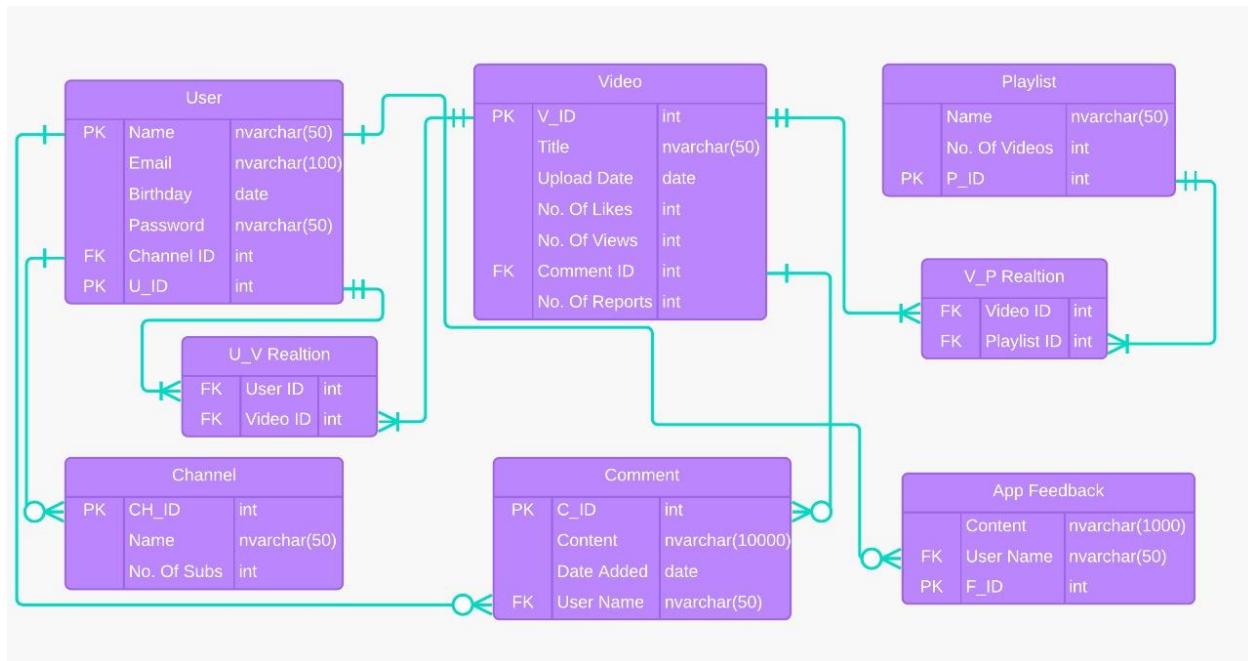


Database Specification (ERD, Tables)

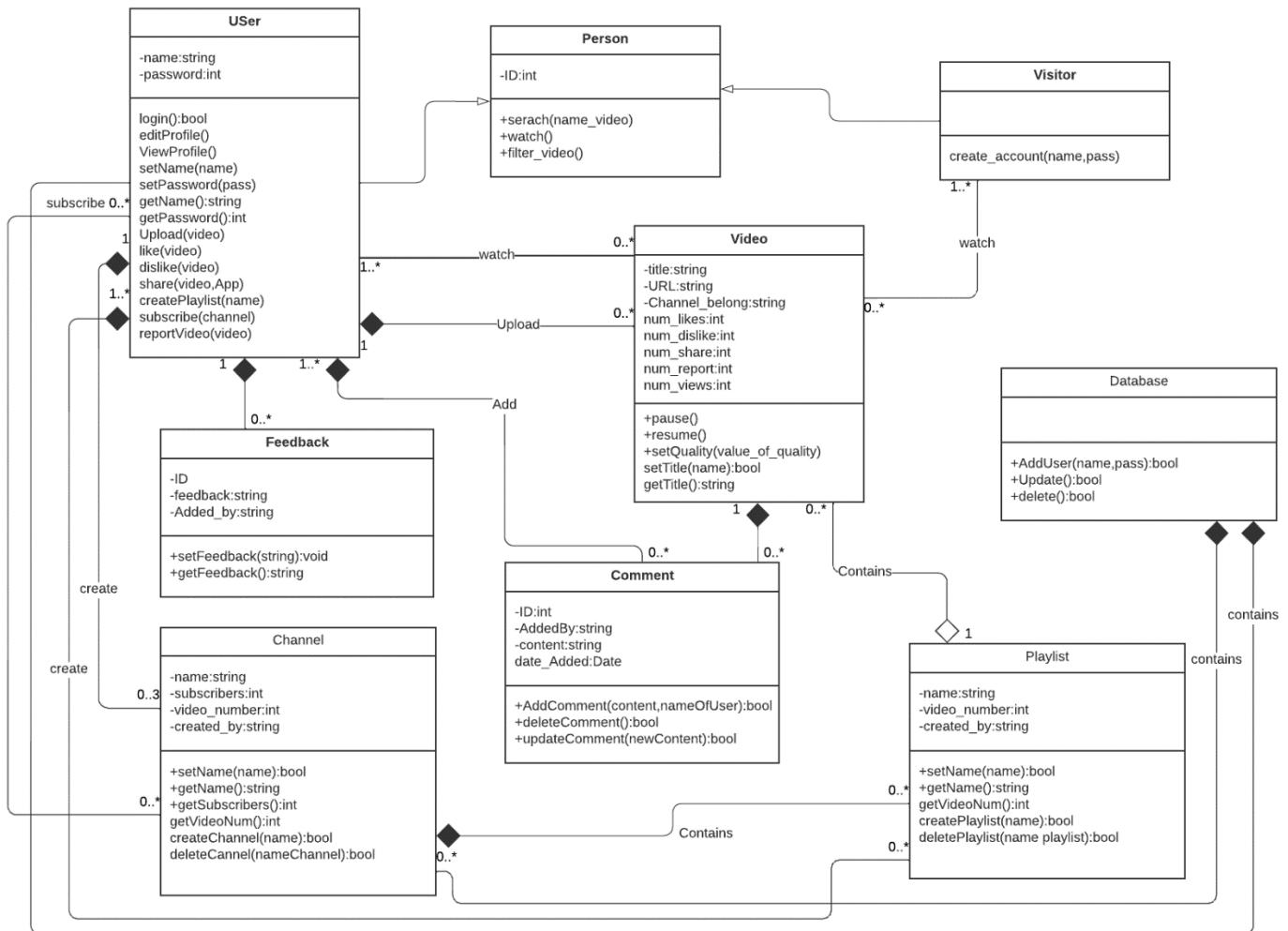
ERD



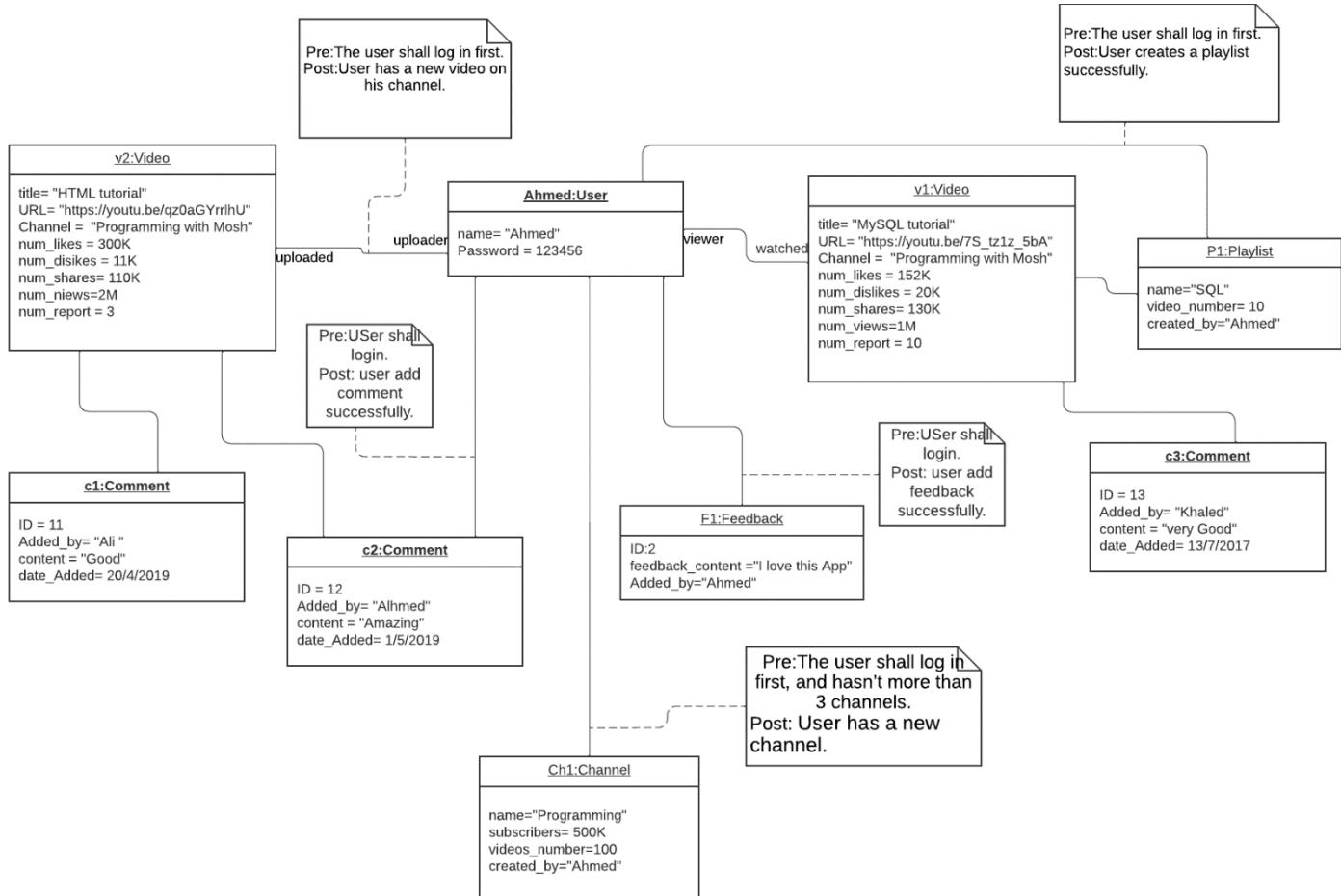
Tables



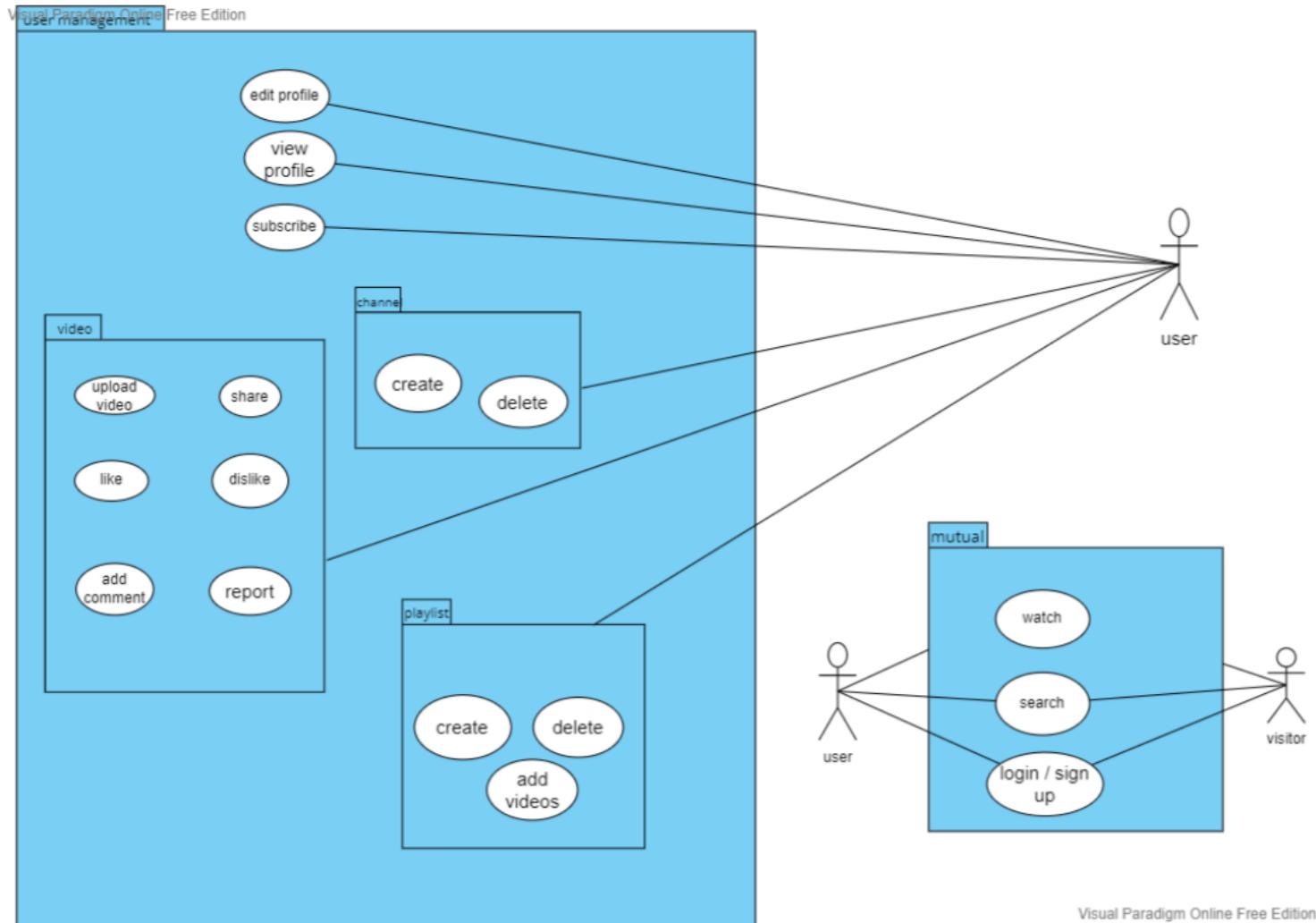
Class Diagram



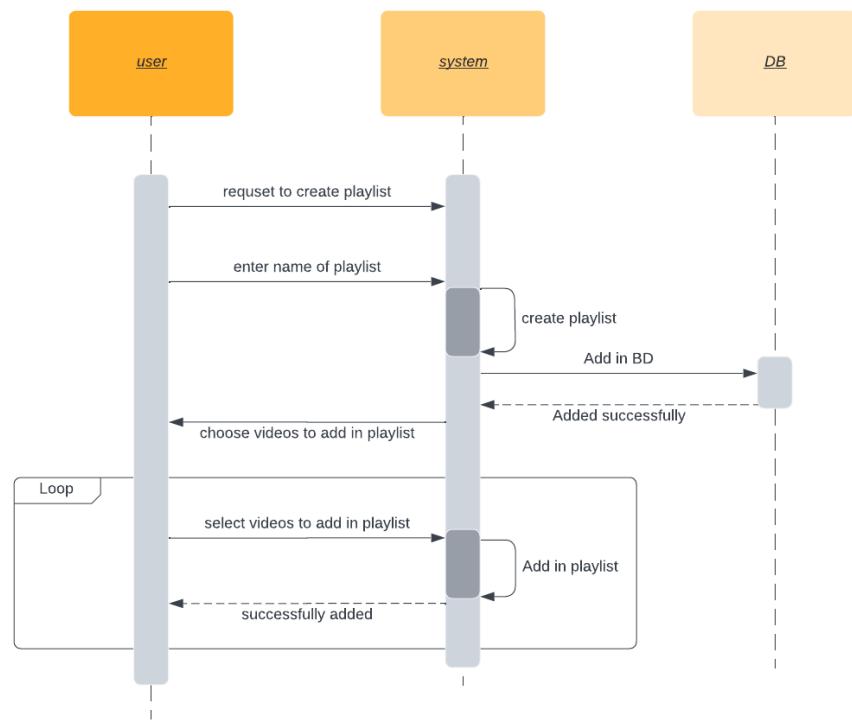
Object Diagrams

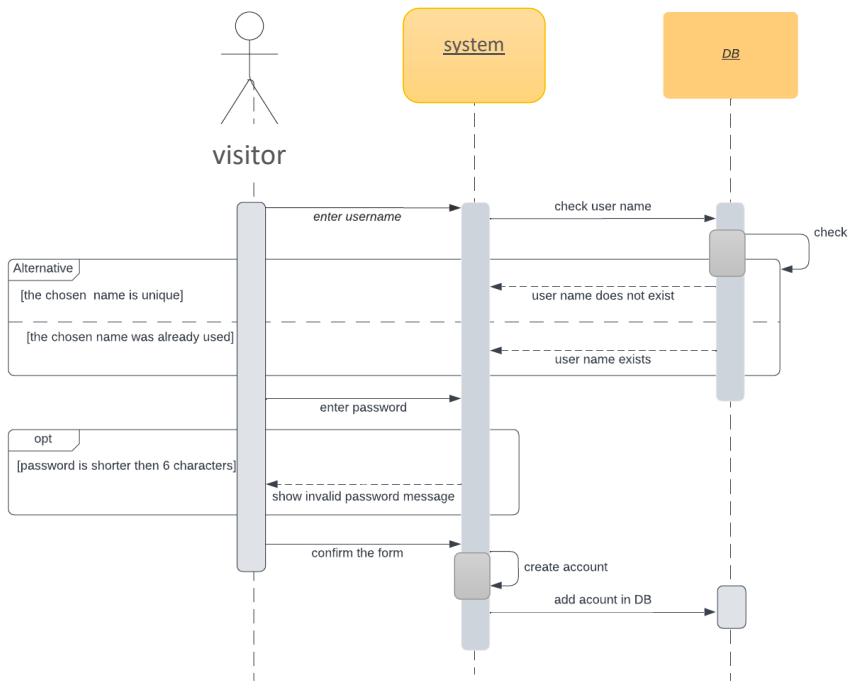


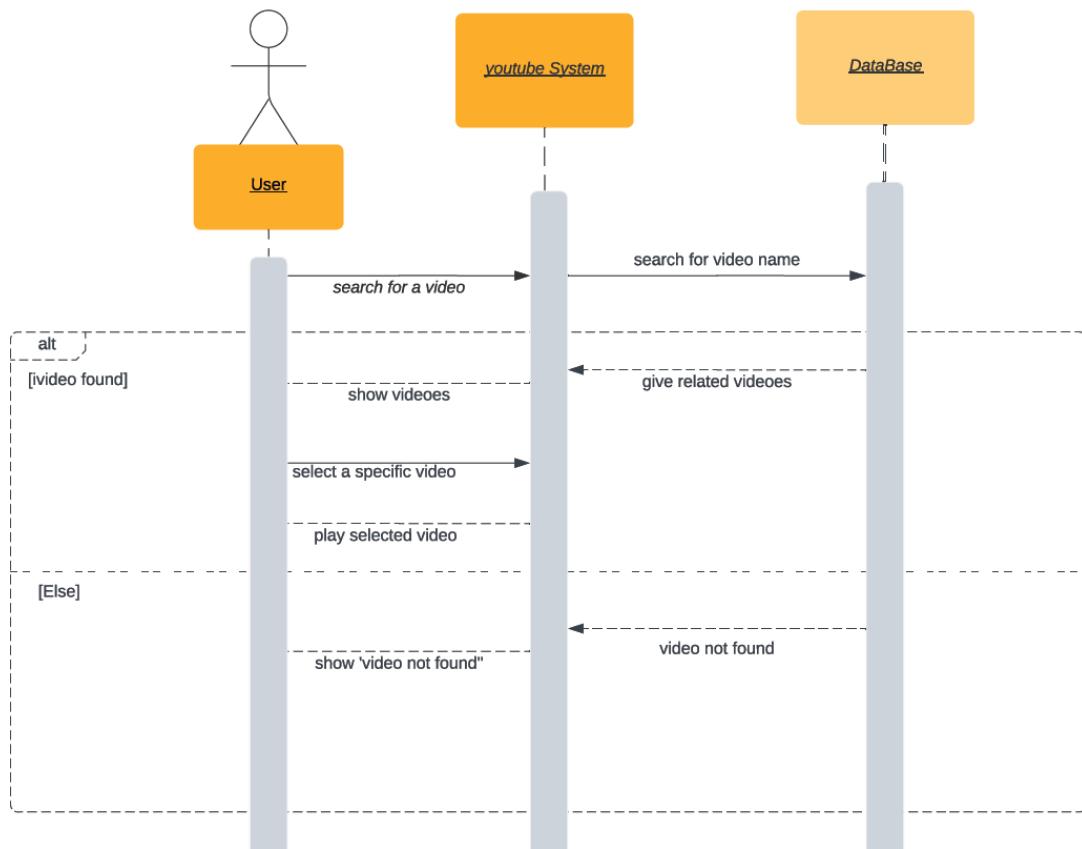
Package Diagram

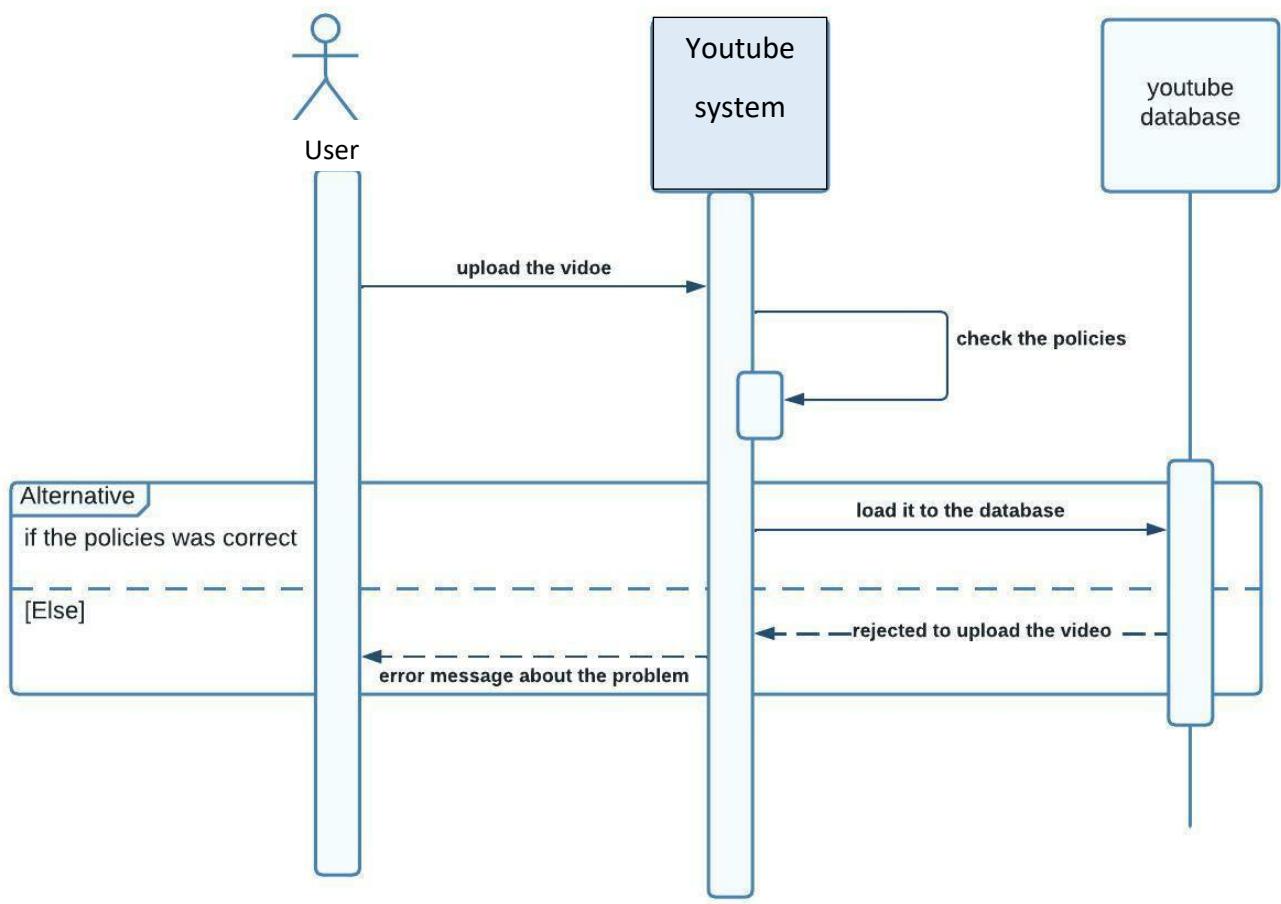


Sequence Diagrams

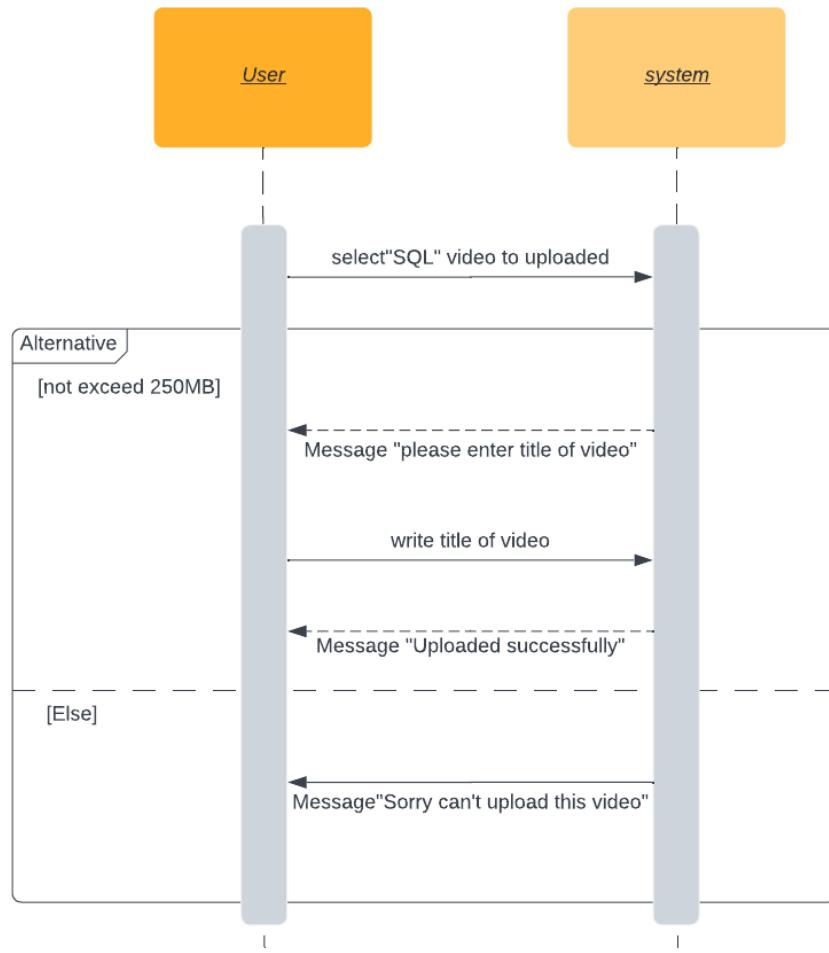


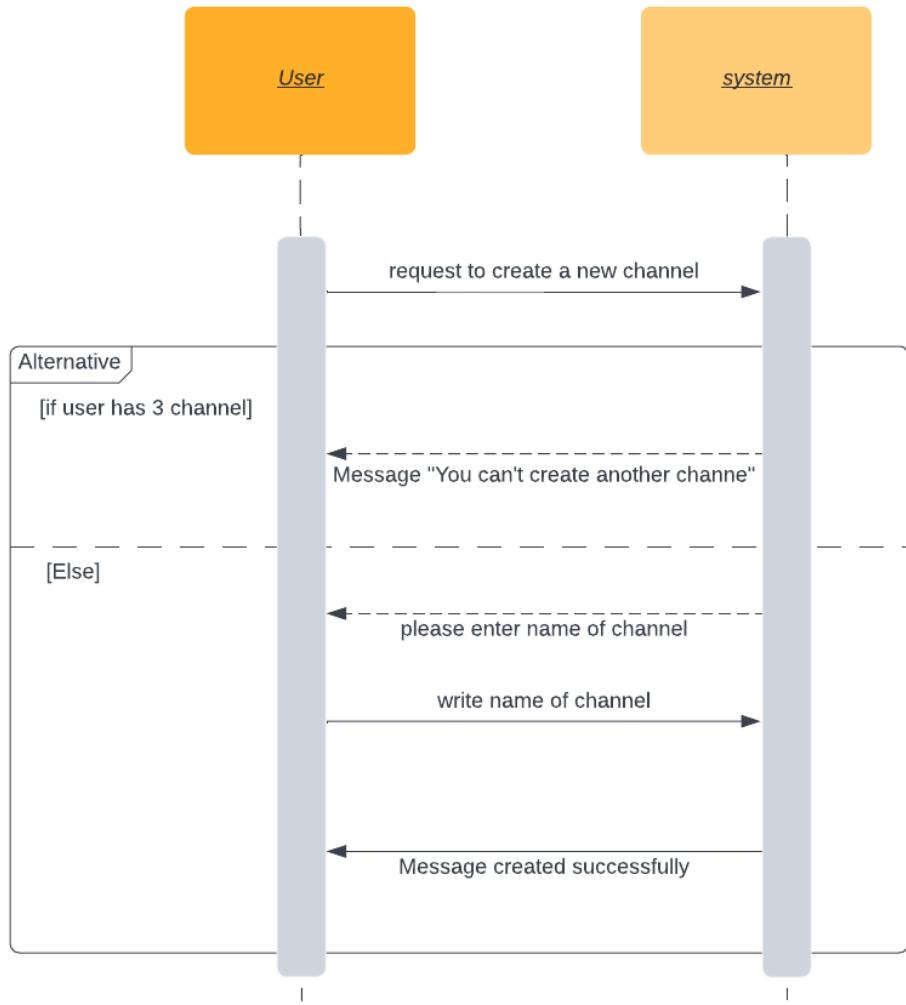






System Sequence Diagram(SSD)





PHASE 2

Design pattern description:

Singleton Pattern:

Context:

You only want to create one and only instance of that class (database) and provide a global point of access to it.

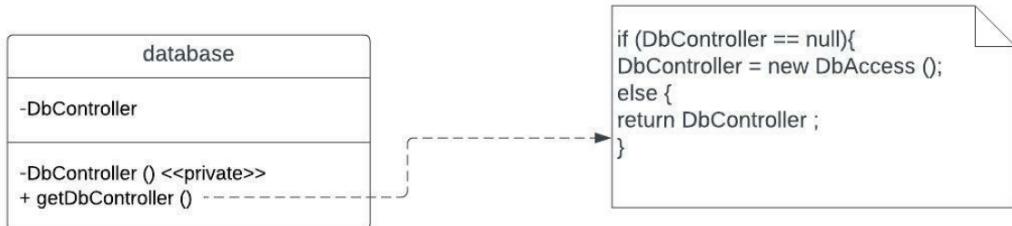
Problem:

Application needs one access to, and only one, instance of an object. Additionally, initialization and global access are necessary.

Forces:

You need to ensure that you encapsulate the class, “just-time initialization” or initialization on first use”

Solution:



Make the class of the single instance responsible for access and "initialization on first use". The single instance is a private static attribute. The accessor function is a public static method.

Design pattern description: Immutable Pattern

Context:

An immutable object is an object that has a state that never changes after creation.

Problem:

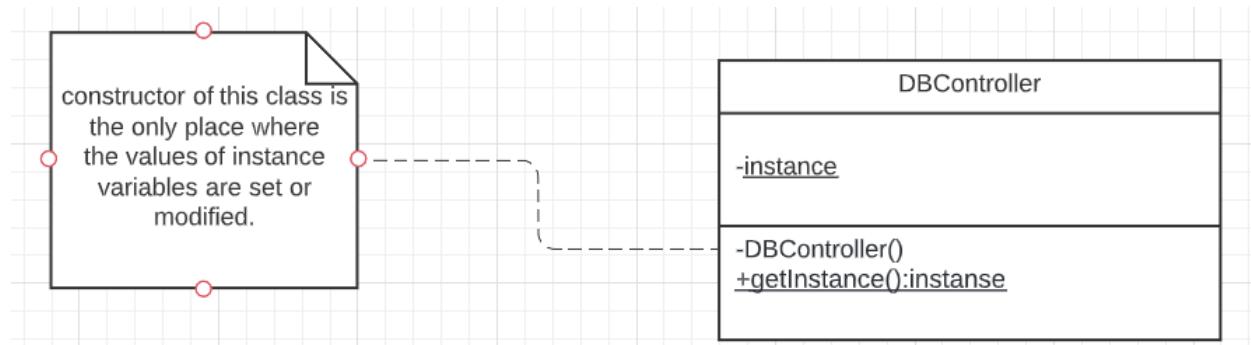
How can you effectively sure the state of object never change after creation.

Forces:

There must be no would allow ‘illegal’ modification of an immutable object.

solution:

Ensure that the constructor of the immutable class is the only place where the values of instance variables are set or modified.



Design pattern description:

Observer (Publish-Subscribe) Pattern

Context:

Mechanism to notify multiple objects about any events that happen to the object they're observing

Problem:

how to notify all users what the new about this channel who subscribe in it without coupled code, cost loosely

Forces:

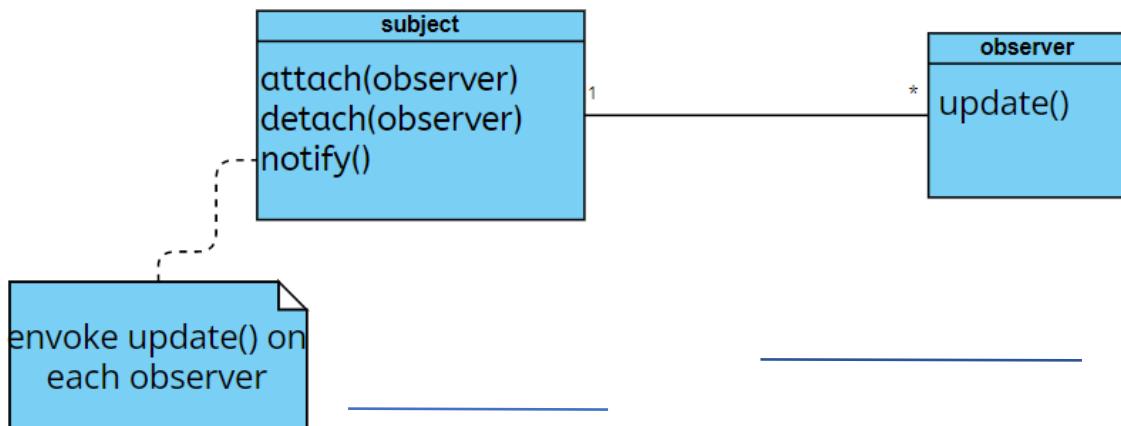
It should be ensured that when one object changes state, an open-ended number of dependent objects are updated automatically.

solution:

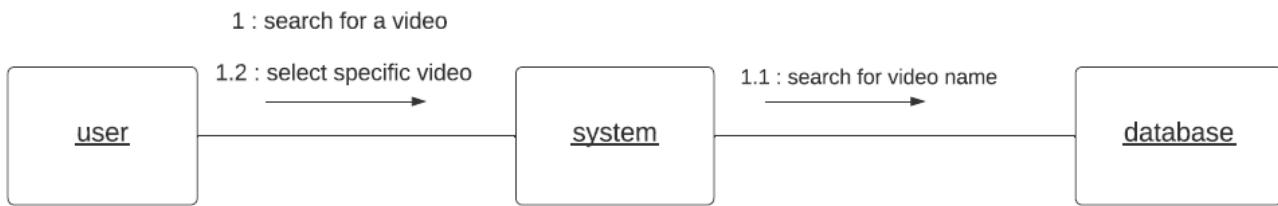
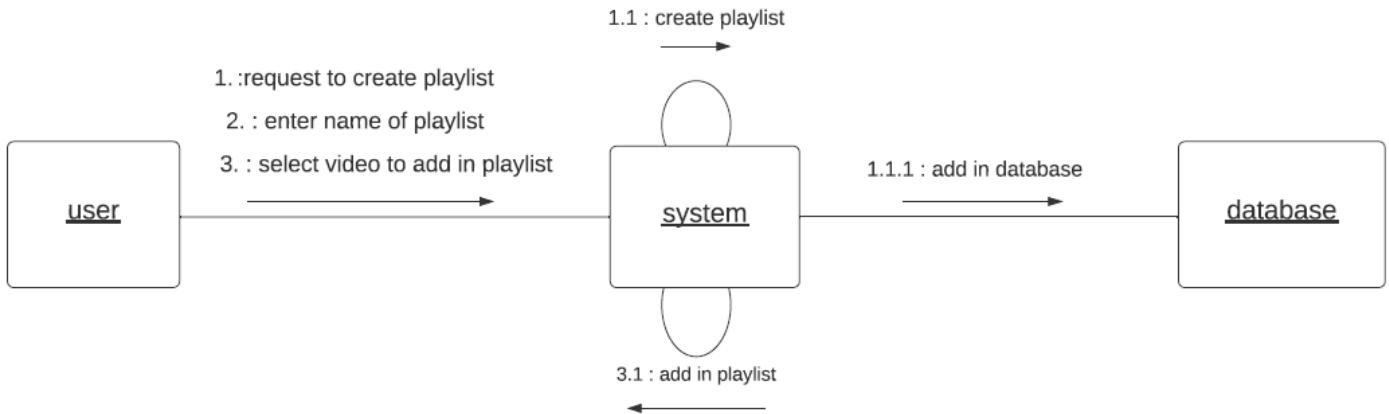
Define one Subject-channel- and many Observer -users- object.

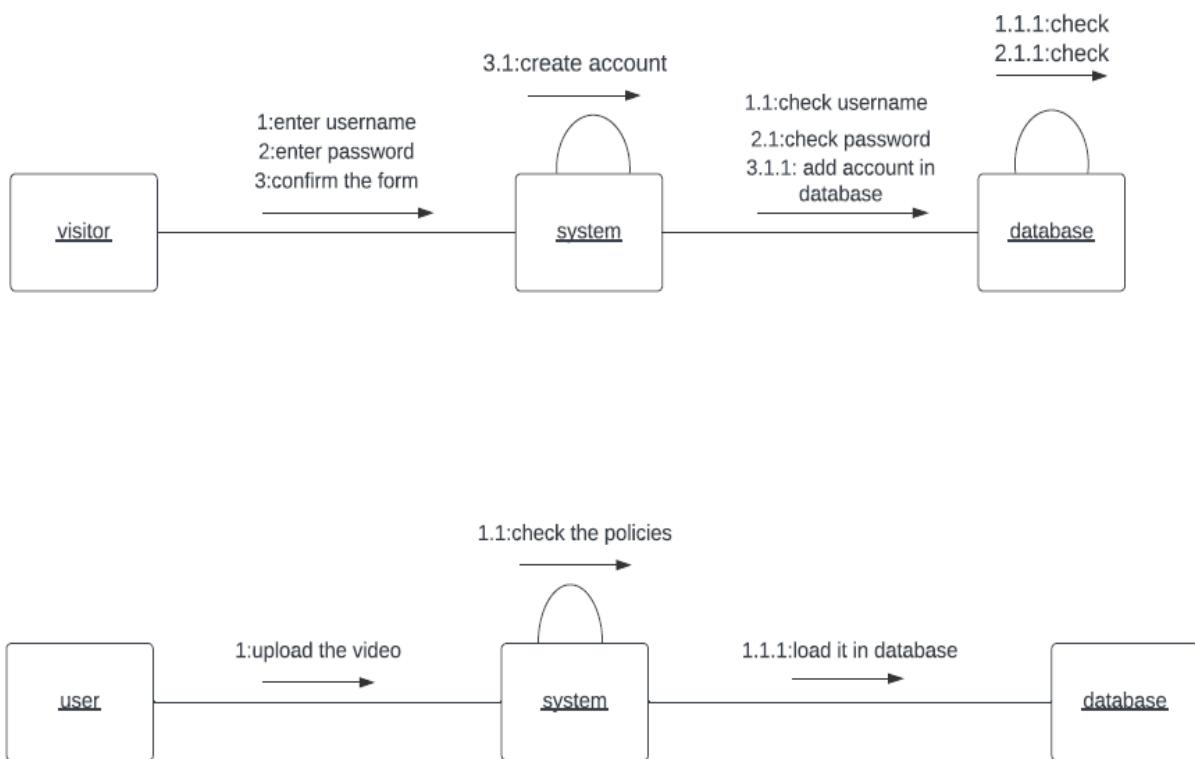
The observers register themselves with the subject,

Push Model : if the state of the subject changes, all users will be notified of details about this change



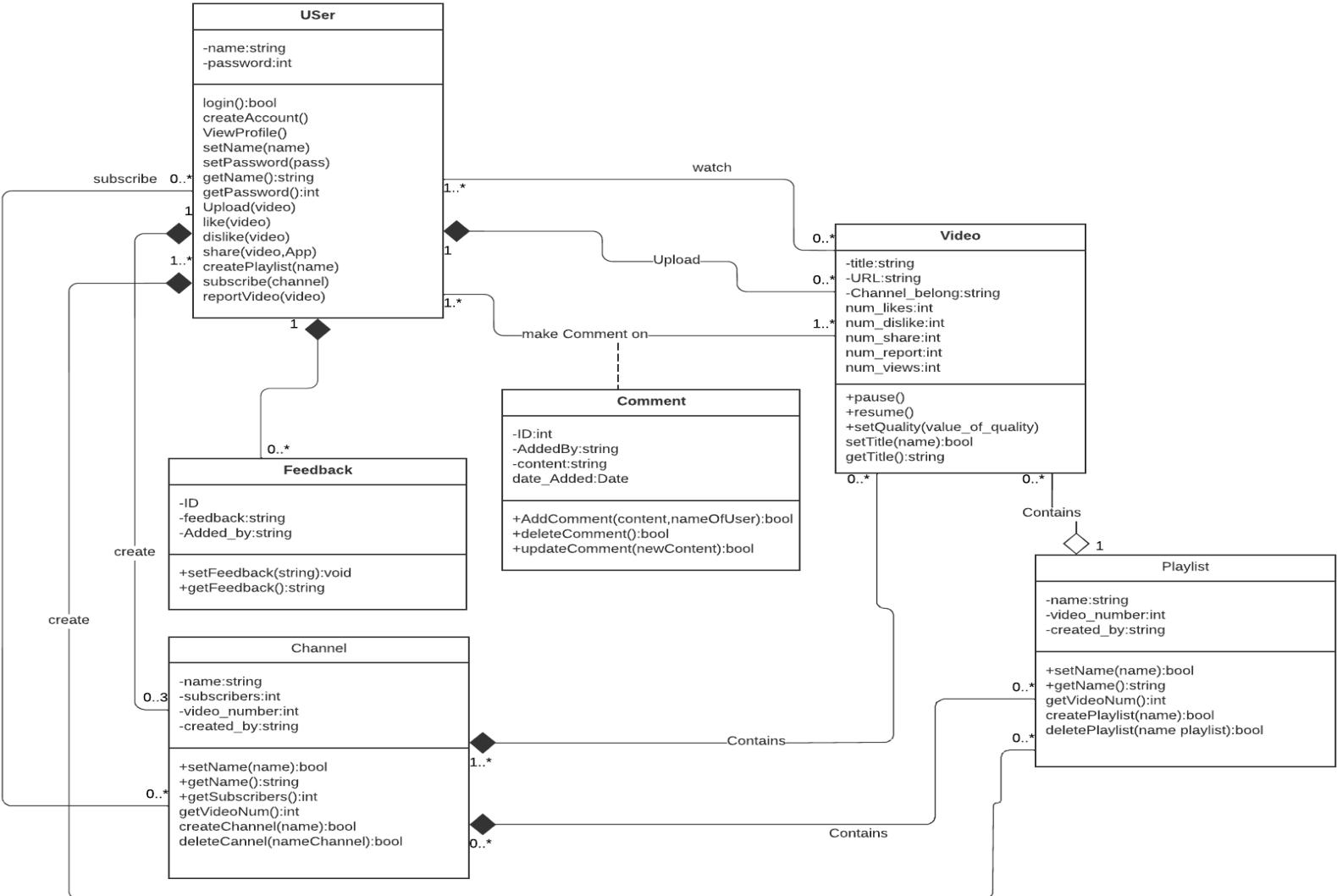
Collaboration/Communication Diagrams





Class Diagram:

An intermediate version based on the interaction diagrams.



A final version, after applying the design patterns and any other modifications.

