Video sharing application: Videosharetube

***Anh Tuan Pham**, 1423003, tap403@student.bham.ac.uk*

***Anant Khanna**, 1219348, axk249@student.bham.ac.uk*

***Darryl Lisk-Carew**, 1376625, drl425@student.bham.ac.uk*

***Khushbu Hussain**, 1622019, krh519@student.bham.ac.uk*

***Olamide Akindele**, 1341309, oxa309@student.bham.ac.uk*

***Talat Rani** 1473521, txr521@student.bham.ac.uk*

Group 1

University of Birmingham

Software Engineering II

Table of Contents

Assumptions on the Scope of the Videosharetube System

The assignment is about a video management application. The application name is Videosharetube, with the target customers of adult users. We will be focusing on the subsystem involving users personalising their experiences with the application by enabling profile customising as much as possible. The subsystem should have 2 main components: personal profile and video channels.

Firstly, the user will have a unique username, a secure password and a unique E-Mail address when they first sign up. The password should be 8 to 256 character long with at least 1 uppercase character, 1 lowercase character and 1 numerical character. The system should have a mechanism for validation of these criteria: the username should have not existed yet then and the password should be 8 to 256 characters long with at least 1 lower case character, 1 upper case character and 1 numerical character. In the personal profile, the users should be able to present themselves with their own profile pictures. The pictures can be uploaded by the users in maximum size of 2 MB. There should be a converter to convert the images uploaded to a certain image format. The uniformed format should be jpeg, and the available formats that can be upload should be JPG, JPEG and PNG. The users should also be able to choose different layouts to present themselves to the users. There should be different layout models provided for the users to choose from. It is to be possible for certain users to upload their own profile layouts. The layouts should also include uploading different pictures to look more artistic. The final part of the personal profile should be about the personal info of the users. The users should be able to present about themselves as much as possible. The content of this part should include their name, date of birth, occupation, education, belief, preferences and interest in different things and their personal achievement. The users should be also provide a short passage for them to choose to express themselves. For privacy and personal interest, the users should be able to choose which parts are to be made visible, and whom to be shown to if they are visible.

The user can access to their account by typing in their username and password. If they forget details, they can choose to reset password by using the E- Mail address. If they fail signing in for 3 times, the account will then be locked temporarily.

In the video channel component, the users should be able to firstly decorate their channels uniquely. This means they should be able to set a picture avatar to present their channels. The pictures, like in the personal profile, can be uploaded by the users in maximum size of 2 MB with a converter to uniform the format with the format restriction of being JPG, JPEG and PNG. The layout policy in this part should also be the same as the personal profile with provided models, the choices for importing compatible personal models and uploading different pictures for different parts in the channels. After the decorations, the most important aspect for the channel should be the possibility of videos uploading. The system should have a 2-level moderation mechanism. The first level is automatic and the second level is manual.

The users should be able to upload videos to their channels, as long as the videos pass the automatic moderator. If the videos fail the first level, it will be assed manually and would only be uploaded if they pass this second level. The criteria for checking in the moderation are appropriate contents for users and legally uploaded. The size should be limited at 2GB for normal users, and can be unlimited for users with premium memberships at the cost of £13.99 per month. There should be a converter to convert the format of all videos to mp4 and the formats that can be uploaded should be only MP4, FLV, MKV, AVI, 3GP and WMV. The upload speed should be fast and about equally with the download speed of the users' internet connection. After uploading the video, the users should be able to write descriptions for the videos for their personal gains. On the channels, the users can create playlists to group different videos together and write descriptions for these playlists. The users can also connect their channel to other channels as partners through invitations. Only when given permission from both sides, the channels can be connected. The users should also be able to choose the viewing policy of their videos and channels, only people are given permission can view their videos or channels. There should be pay out for the owners of the channel and videos in the form of credit per view. The moderations for the videos should also check for illegal action of increasing views for videos or channels.

The subsystem should always be available for almost perfect 24/7. There should be different back up of the subsystem in case the running one breaks down. The subsystem should be able to run in different language and on different platforms, machines for different users all over the world. Users' information and transactions' details should be secured implementing trustworthy third party software.

Functional and Non-Functional Requirements

Functional:

- Users can set their own username for the account.
- Users can set their own password for the account.
- Users can set their own E-Mail for the account.
- Profile page creation:
    - Users can upload their own profile picture.
        - The pictures should be converted to the uniform format.
    - Users can choose the layout for their profile page:
        - Users can upload custom layout or user pre-made ones.
        - Users can upload pictures in the layouts.
    - Users can have different information in their profile:
        - Users can have name on their profile.
        - Users can have date of birth on their profile.
        - Users can have education history on their profile.
        - Users can have occupation history on their profile.
        - Users can have personal achievements on their profile.
        - Users can have personal belief on their profile.
        - Users can have personal interest and preferences on their profile.
            - Users can have personal interest in sports on their profile.

            - Users can have personal interest in books on their profile.
            - Users can have personal interest in food on their profile.
            - Users can add additional personal interest on their profile.
        - Users can write additional information on a paragraph field.
    - Users can set privacy settings for the profile page.
        - Users can set who to show to.
        - Users can set what to show.
- User logging:
    - User can log in with their username and password.
    - User can choose to reset password with the E-Mail address.
    - The account will be locked if the user fail signing in thrice.
- Video channel management:
    - Users can set picture avatar for their channels.
        - The pictures should be converted to the uniform format.
    - Users can set layout for their channels.
        - Users can upload custom layout or user pre-made ones.
        - Users can upload pictures in the layouts.
    - Users can upload videos to the channels.
        - There should be a 2-level moderation mechanism for uploading videos
            - The first level of moderation is automatic. If the videos past this level, it is considered valid for uploading.
            - The second level of moderation is manual, only needed when the videos fail the first level.
        - The videos should be converted to the uniform format.
        - The size of the videos should be limited for normal users.
        - The size of the videos should be unlimited for premium users.
        - Users can write description for the videos.
    - Users can create playlists to group the videos together.
        - Users can write description for the playlists.
    - Users can connect their channels to others' channels.
        - The connections will need to have both channels' owners' permission.
        - Owners can give out invitation of connection.
    - Users can set privacy settings for the profile page.
        - Users can set who to show to.
        - Users can set what to show.
- There should be pay out mechanism for users in the form of credit per views.
    - There should be moderation to check for illegal action of increasing views.

Non-functional:

- Configurability:
    - Username should be unique.
    - E-Mail address should be unique.
    - Passwords should be secure:

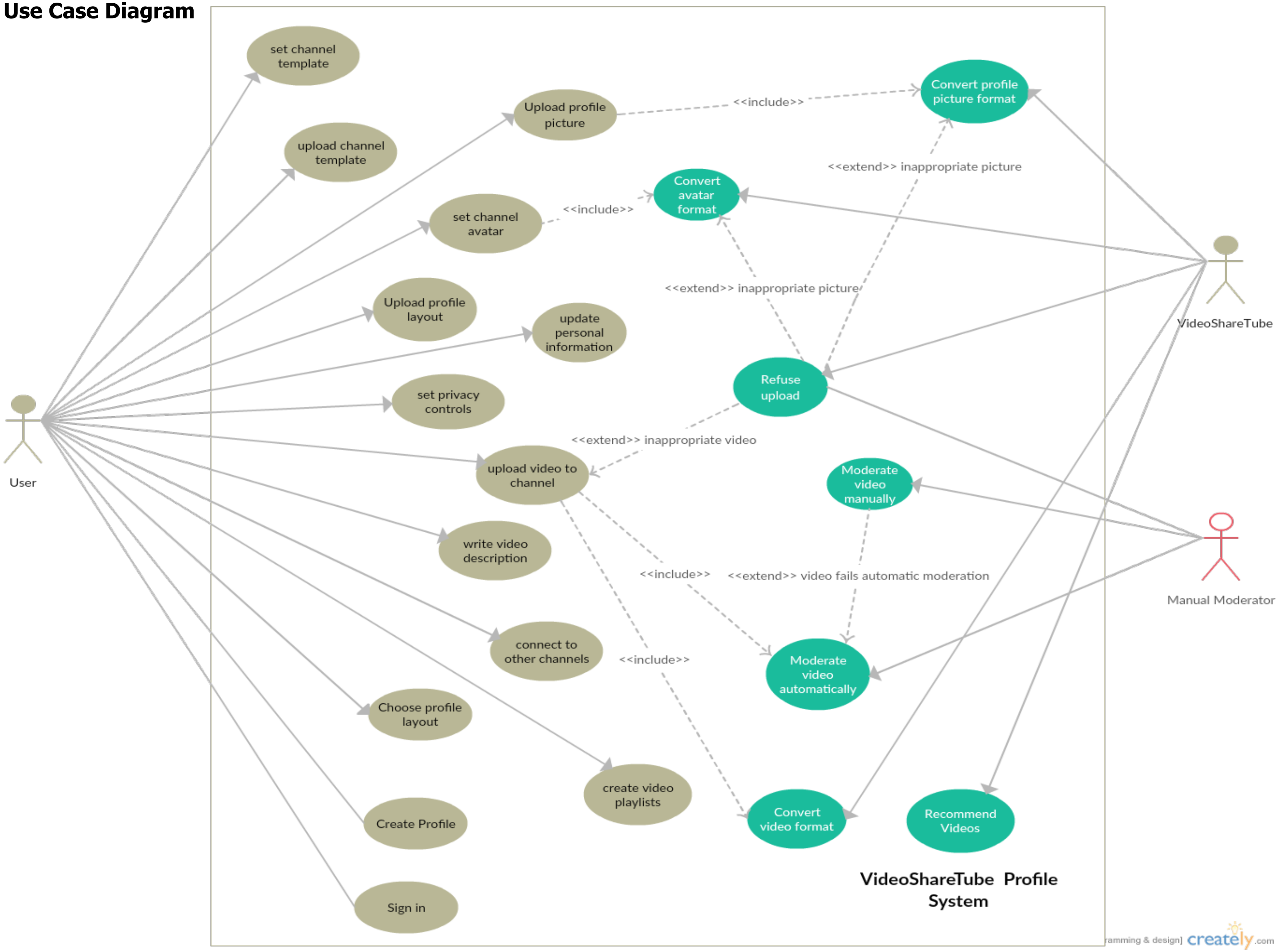
        - 8 to 256 characters long.
        - At least 1 uppercase character.
        - At least 1 lowercase character.
        - At least 1 numerical character.
    - Pictures:
        - Maximum size for profile and avatar pictures is 2MB.
        - The formats of the profile and avatar pictures to be uploaded are JPG, JPEG and PNG.
        - The converted format of the profile and avatar pictures is JPEG.
    - Videos:
        - Maximum size of normal users is 2GB.
        - The formats of the videos to be uploaded are MP4, FLV, MKV, AVI, 3GP and WMV.
        - The converted format of the videos is MP4.
    - Premium membership cost £13.99 monthly.
- Availability:
    - The system should always be online for almost perfect 24/7.
    - The system should have back up database in case of failure of the main data station.
- Compatibility:
    - The system should be run in different languages.
    - The system should be able to run in different platform: Web browsers, iOS and Android.
- Performances:
    - Users should be able to access their account instantly.
    - Users should be able to update their account instantly.
    - Users should be able to upload their videos at unlimited speed.
    - The conversion speed of the pictures should be instantly.
    - The conversion speed of the videos should be at least 10Mb/s.
    - The automatic moderation of the videos should be 1 second for every 1GB of the size of the videos.

- Security:
    - Users’ information should be secure using encryption software from trustworthy third party.
    - Users’ transactions’ information should be secure using software from trustworthy third party.
    - There should be moderations checking for illegal increase of views like checking IP of users.

UML Documentation

The following pages document the analysis and design of the Videosharetube system. The diagrams in order are: Use Case diagram, Activity diagram, Class Analysis (Noun-Verb Analysis, First-Cut Class diagram and Class diagram), Object diagram, Sequence diagram and State diagram, Component diagram and Deployment diagram. The section concludes with Architectural Tradeoffs Analysis Method (ATAM) evaluation.

Use Case Diagram



Use Case Documentation

Post a video

Pre-condition

1. The user has a profile set-up on the application.
2. The user has a video less than 2MB in size and in MP4, FLV, MKV, AVI, 3GP or WMV format.

Flow of events

1. The use case starts when the user attempts to post a video on the application.
2. The system checks the video to see if it passes the moderation criteria.
3. If the video fails, the automatic moderation the video is checked by manual moderation checks.
4. If the video fails, the manual moderation checks it is not posted and a message is displayed to the user explaining this.
5. If the video passes the manual or automatic moderation, the video is posted on the application and is available for viewing by the user or browsers of the application.

Post-condition

1. The application has updated the number of videos it has displayed by the user.
2. The application updates the user display to include the new video.

Scenario 1

User Amanda uploads a video on fitness to her station that she intends to share with everyone. It is less than 2MB in size and in mp4 format. The video is moderated and passes. The video is displayed accordingly.

Scenario 2

User Graham tries to post a video on addiction to the website but is unable to do so as it fails the moderator checks. The video is less than 2MB in size and in MP4 format.

Enter personal details

Pre-conditions

1. The user has the application running on their device.

2. The application is running in the language of the user's choice (Spanish).

Flow of events

1. The use case starts when the user enters details into the text boxes on the screen.
2. The system checks each field for a valid entry. For example, the user must enter a valid password that meets a minimum level of security.
3. If the user clicks to save the details they have entered and the password is not long enough, for example, the text in the field is removed and a message pops up for the user to read informing the user that the password entered is not long enough and that they should enter another password.
4. Appropriate messages are displayed for the user until all the fields are completed with valid entries.
5. When the user clicks to save the personal details and all field entries are valid, the system updates by adding a new user to the system. The number of users of the system increases by one.

Post-conditions

1. The system displays a new user profile screen (or other screen).
2. The application updates to increase the number of users by one.

Scenario 1

User Jason accesses the application for the first time. Jason clicks the appropriate button on the screen to join the application as a new user. Jason enters his personal details into the form on-screen. All Jason's field entries are valid and so Jason Bloggs is created as a new user of Videosharetube.

Scenario 2

User Barbara accesses the application to update her password. She already has an account on the application. Barbara navigates to the personal details section and inputs a new password. The password is not secure enough (does not contain at least one uppercase letter) so her password is not updated. A message is displayed on the screen showing that her password is 50% secure and asks Barbara to input another password.

Upload Profile Picture

Pre-conditions

1. User has a profile set up.

Flow of events

1. The use case starts when the user attempts to upload a photograph to the application as their profile picture.
2. The application checks that the image is in a correct format (less than 2MB in size and in JPG, JPEG or PNG format).
3. If the picture is in a correct format the application continues to the next step: that is, to check that the photograph is appropriate.
4. If the picture is not in a correct format, the system does not load the picture onto the user's profile page, instead it displays a message stating that the file is in an invalid format and states the suitable formats. This continues to occur until the user uploads a photograph in a correct format.
5. If the picture is in the correct format, the picture is checked by the system administrator to ensure that the picture is appropriate.
6. If the picture is inappropriate the system does not upload the image and presents a message to the user to change the photograph.
7. If the picture is appropriate the application displays the picture on the profile screen.

Post-conditions

1. The application is updated to display a new photograph on the user's profile page.

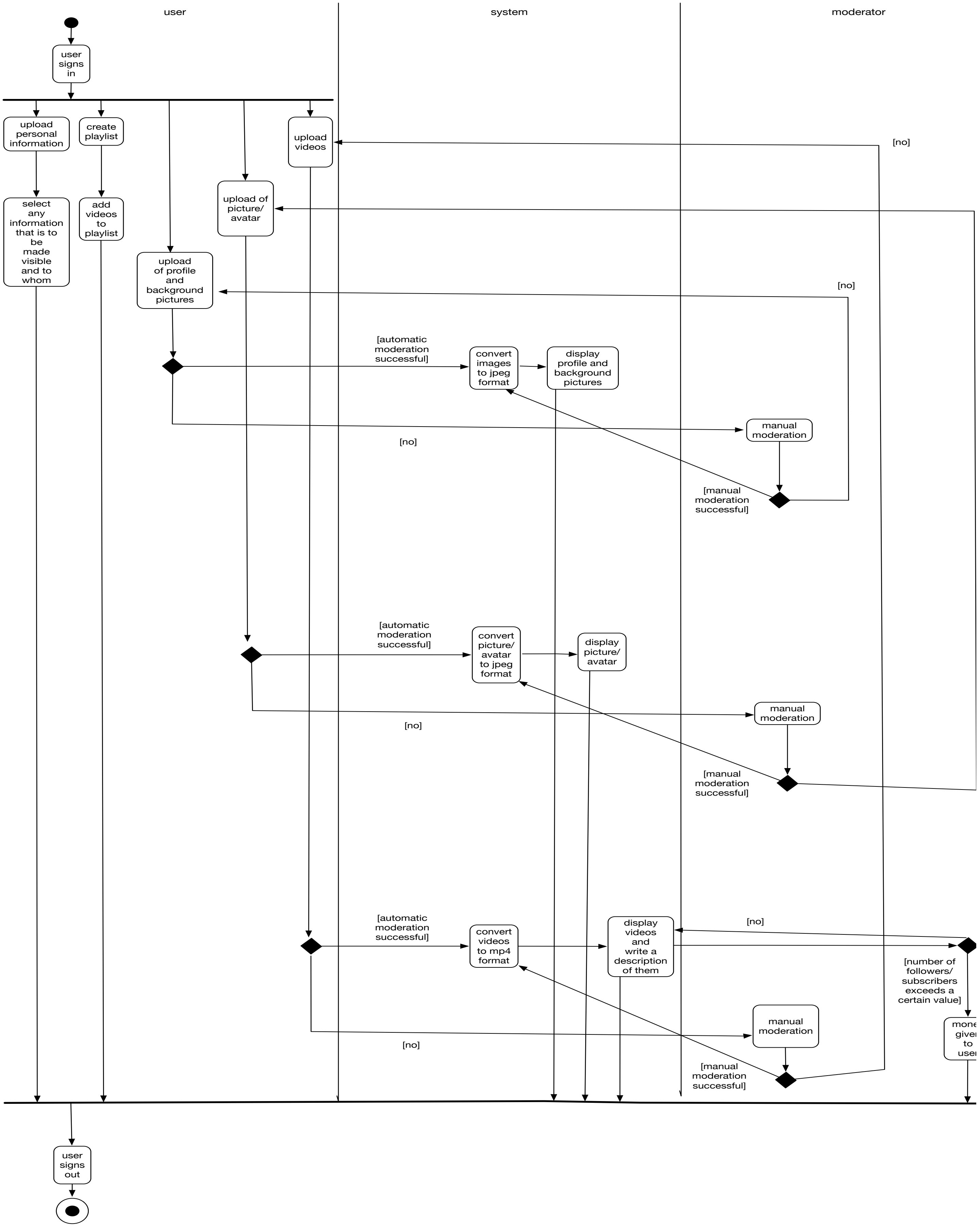
Scenario 1

User James uploads a photograph to his station' profile page. The photograph is checked by the system to be in a suitable format. The photo is then checked by the administrator and passes the check. The photograph is displayed on the user's station.

Scenario 2

User Ben uploads a photograph to his station's profile page. The photograph is checked by the application and it is in a suitable format. The photograph is then checked by the administrator of the site and fails the inspection. The photograph is not displayed on the user's station and the profile page remains in its original state.

Activity Diagram


user
system
moderator
user signs in
upload personal information
create playlist
upload videos
[no]
select any information that is to be made visible and to whom
add videos to playlist
upload of picture/ avatar
upload of profile and background pictures
[no]
[automatic moderation successful]
convert images to jpeg format
display profile and background pictures
manual moderation
[no]
[manual moderation successful]
[automatic moderation successful]
convert picture/ avatar to jpeg format
display picture/ avatar
manual moderation
[no]
[manual moderation successful]
[automatic moderation successful]
convert videos to mp4 format
display videos and write a description of them
[no]
[number of followers/ subscribers exceeds a certain value]
mone give to use
manual moderation
[no]
[manual moderation successful]
user signs out

Noun-Verb Analysis

~~Customers~~

~~Subsystem~~

Users

~~Profile~~

~~Personal profile~~

~~Video channels~~

~~Username~~

~~Password~~

~~Character~~

~~Mechanism~~

~~Profile pictures~~

~~Converter~~

~~Images~~

~~Image format~~

Layout

~~Layout models~~

Personal info

~~Name~~

~~Date of birth~~

~~Occupation~~

~~Education~~

~~Belief~~

~~Preferences~~

~~Interest~~

~~Personal achievement~~

~~Short passage~~

~~Avatar~~

Moderation mechanism

~~Videos~~

~~Automatic moderator~~

~~Appropriate contents~~

~~Size~~

~~Premium membership~~

~~Upload speed~~

~~Download speed~~

~~Descriptions~~

Playlists

~~Donation~~

~~Direct debit~~

~~Paypal~~

~~Invitations~~

~~Permission~~

~~Viewing policy~~

~~Pay out~~

~~Illegal action~~

~~Views~~

~~Back up~~

~~Language~~

~~Platforms~~

~~Machines~~

~~Transactions~~

~~Software~~

~~Format~~

List of Candidate Classes

User

Personal Info

Playlist

Videos

Video Channel

Picture

Converter

Layout

Verbs and verb phrases

Present themselves

Convert

Choose

Upload

Present about themselves

Give different video recommendations

Decorate

Set

Importing

Assessed

Checking

Convert

Write

Connect

Given permission

Run

Implementing

Potential Operations for classes

User

SetUsername

SetPassword

CreatePersonalProfile

CreateVideoChannel

Personal Info

setName

setDOB

setOccupation

setEducation

setBeliefs

setPreferences

setInterests

setPersonalAchievements

setShortPassage

uploadAvatar

convertAvatar

setAccessibilitySetting

Playlist

setTitle

setDescription

uploadVideo

convertVideo

Video Channel

recommendVideos

uploadAvatar

convertAvatar

uploadVideo

convertVideo

setLayout

uploadLayout

setAccessibilitySettings

sendChannelLinkRequest

acceptChannelLinkRequest

Layout

chooseTemplate

setVideoChannelLayout

setPersonalInformationLayout

Videos

uploadVideo

convertVideo

setDescription

setTitle

addToPlaylist

Picture

uploadPicture

convertPicture

ConverterAndModeratorMechanism

convertPicture

convertVideo

moderateAutomatic

moderateManually

CRC Cards

User	
Responsibilities	Collaborators
Set up an account for user consisting of username, password and E-Mail Set up a personal profile Set up a video channel Showing whether it is premium or normal user Allow users to sign in Allow to lock account	Personal Information Video Channel

Video Channel	
Responsibilities	Collaborators
Allow users to upload videos Allow users to create a playlist Allow users to edit video channel information To change layout To connect to other video channels Allow others to subscribe to the channel	Video Playlist Picture Layout

ConverterAndModerationMechanism	
Responsibilities	Collaborators
To ensure suitable content is uploaded To check for video format compatibility To check for picture format compatibility To convert a video if it passes the moderating criteria To convert a picture	Video Picture

Playlist	
Responsibilities	Collaborators
To allow users to group together videos of their own choice To edit the playlist description	Video Video channel

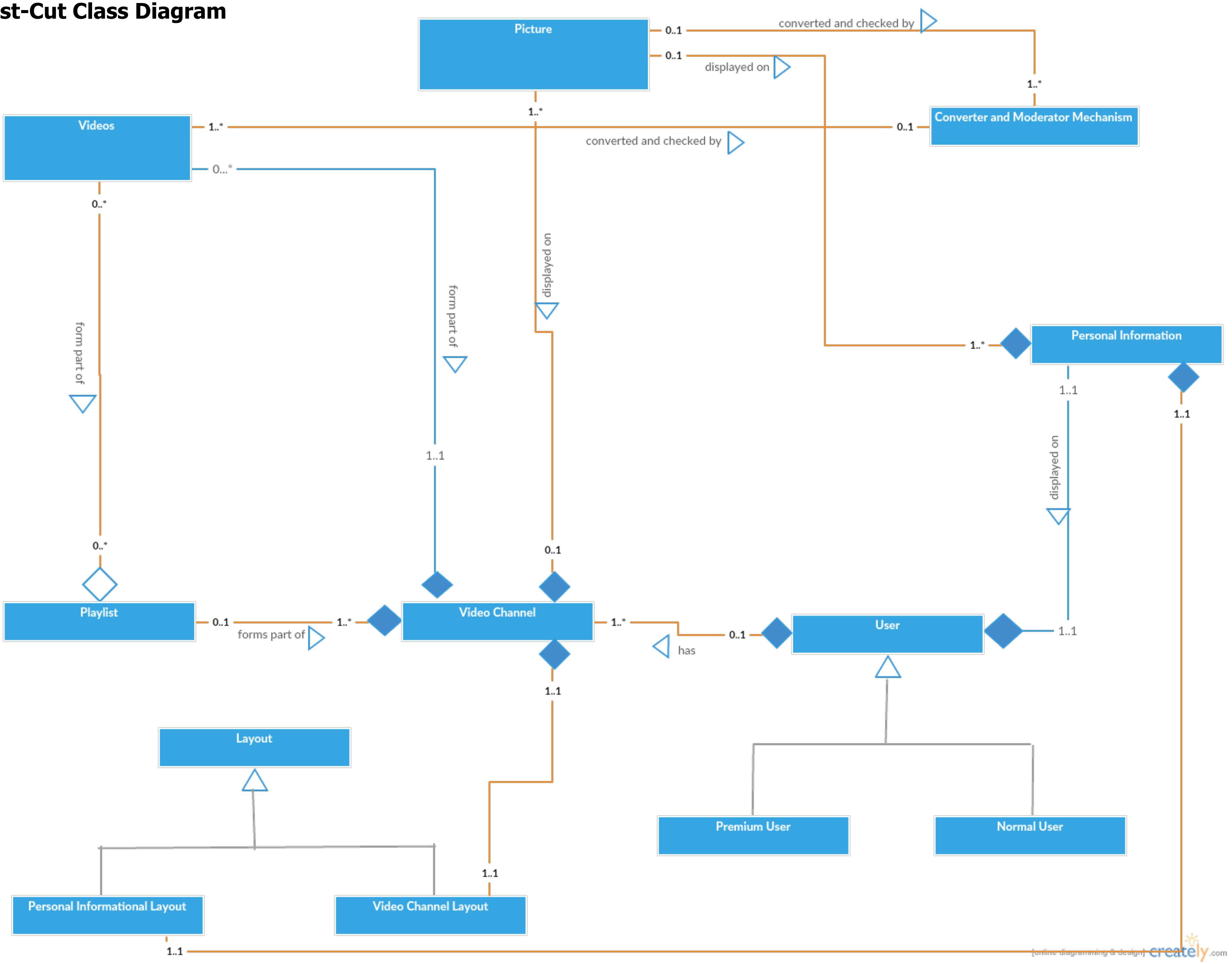
Video	
Responsibilities	Collaborators
To hold video object To edit video title To edit video description To show view count of the video	ConverterAndModerationMechanism

Personal Information	
Responsibilities	Collaborators
To allow users to add information about their name, age, occupation etc to their personal profile To change layout	Personal profile

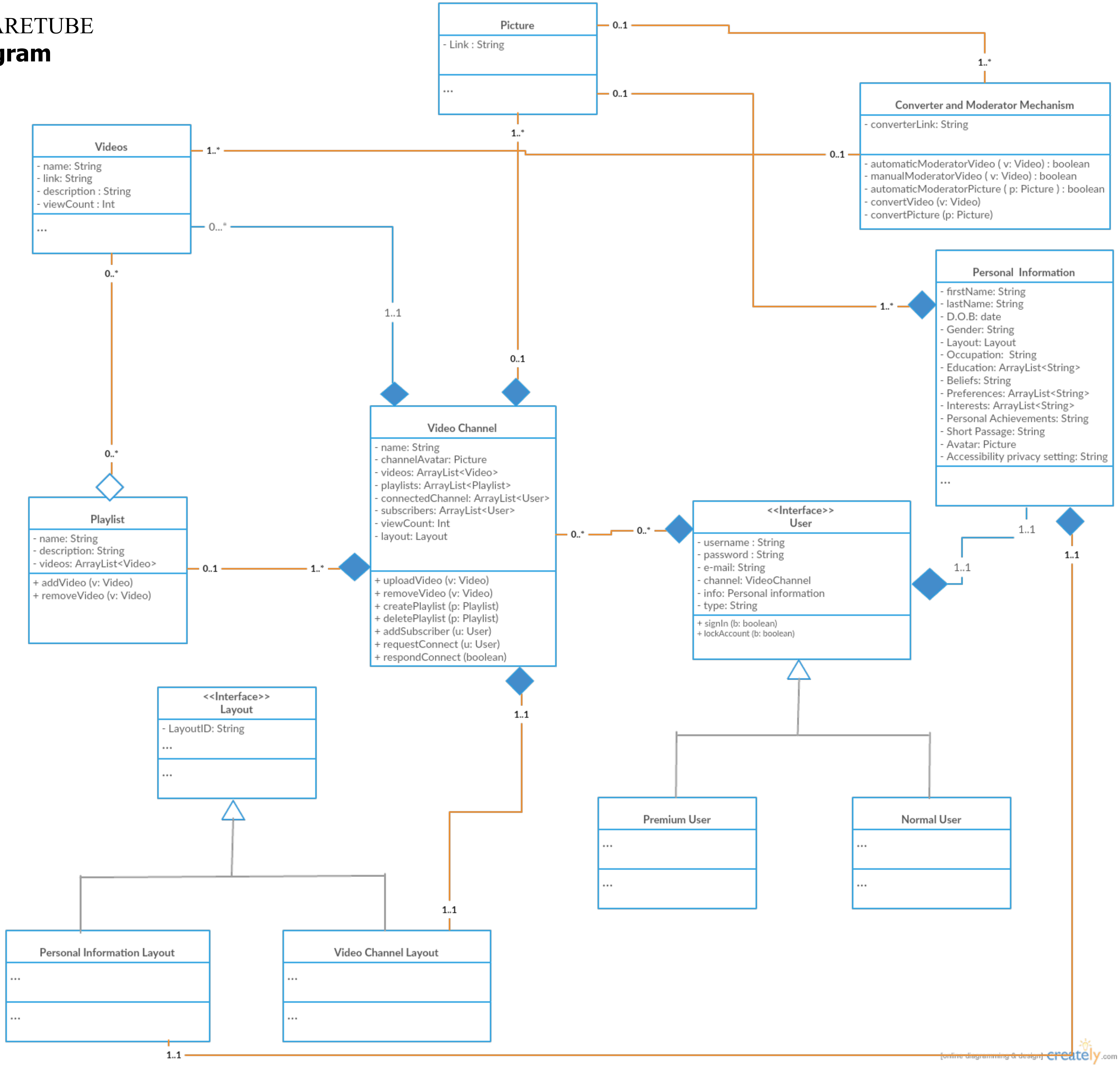
Layout	
Responsibilities	Collaborators
To provide layout templates To allow users to choose their personal information layout To allow users to choose their video channel layout	Video Channel Layout Personal Information Layout

Picture	
Responsibilities	Collaborators
To upload a picture	ConverterAndModerationMechanism

First-Cut Class Diagram



Class Diagram



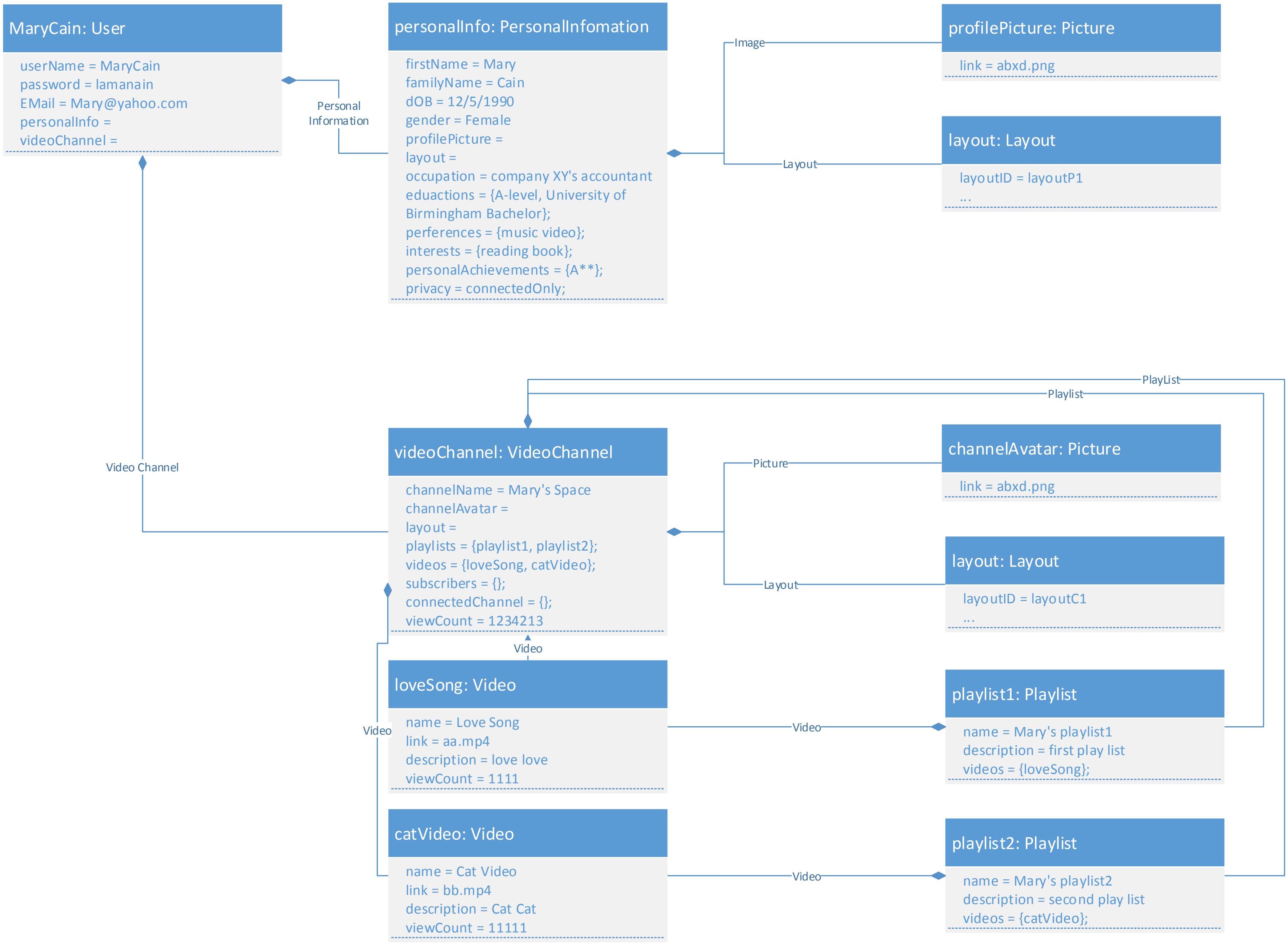
Notes on Class Diagram Analysis:

The first-cut class diagram shows only the names of classes and the relationships with the use multiplicity and directionality of relationship descriptors between classes.

The non-opaque diamond icons demonstrate aggregation between classes whereas the opaque diamond icons demonstrate stronger aggregation between the respective classes.

The final class diagram is a revised version includes attributes and operations where relevant; some classes such as ‘Picture’ and ‘Video’ do not require operations defined as this is uploaded directly by the user. Other classes such ‘Video Channel Layout’ inherit the interfaces (indicated by the arrow head) ‘Layout’ and ‘User’ therefore have no requirement for extra attributes or operations.

Object Diagram



MaryCain: User

userName = MaryCain
password = lamanain
EMail = Mary@yahoo.com
personalInfo =
videoChannel =

personalInfo: PersonalInfomation

firstName = Mary
familyName = Cain
dOB = 12/5/1990
gender = Female
profilePicture =
layout =
occupation = company XY's accountant
eduactions = {A-level, University of Birmingham Bachelor};
perferences = {music video};
interests = {reading book};
personalAchievements = {A**};
privacy = connectedOnly;

profilePicture: Picture

link = abxd.png

layout: Layout

layoutID = layoutP1
...

videoChannel: VideoChannel

channelName = Mary's Space
channelAvatar =
layout =
playlists = {playlist1, playlist2};
videos = {loveSong, catVideo};
subscribers = {};
connectedChannel = {};
viewCount = 1234213

channelAvatar: Picture

link = abxd.png

layout: Layout

layoutID = layoutC1
...

loveSong: Video

name = Love Song
link = aa.mp4
description = love love
viewCount = 1111

catVideo: Video

name = Cat Video
link = bb.mp4
description = Cat Cat
viewCount = 11111

playlist1: Playlist

name = Mary's playlist1
description = first play list
videos = {loveSong};

playlist2: Playlist

name = Mary's playlist2
description = second play list
videos = {catVideo};

Notes for Object Diagram:

The scenario for the object diagram is about the user account. A user account will first have a user object. The user object is a User class.

The user object will be having a user name attribute, a password attribute, a personal information object and a video channel object. The personal information object is of PersonalInformation class, the video channel object is of VideoChannel class.

In the personal information object, there will be attributes about different things, a profile picture and a layout object. The profile picture is Image class. The layout object is of Layout class.

In the video channel, there will be a channel name attribute, a list attribute of multiple playlist objects, a list attribute of multiple videos objects, a channel avatar object, a layout object, a list attribute of subscriber objects and a view attribute. The playlist object is of Playlist class. The video object is of Video class. The channel avatar object is of an Image class. The Layout object is of layout class. The subscriber object is of User class.

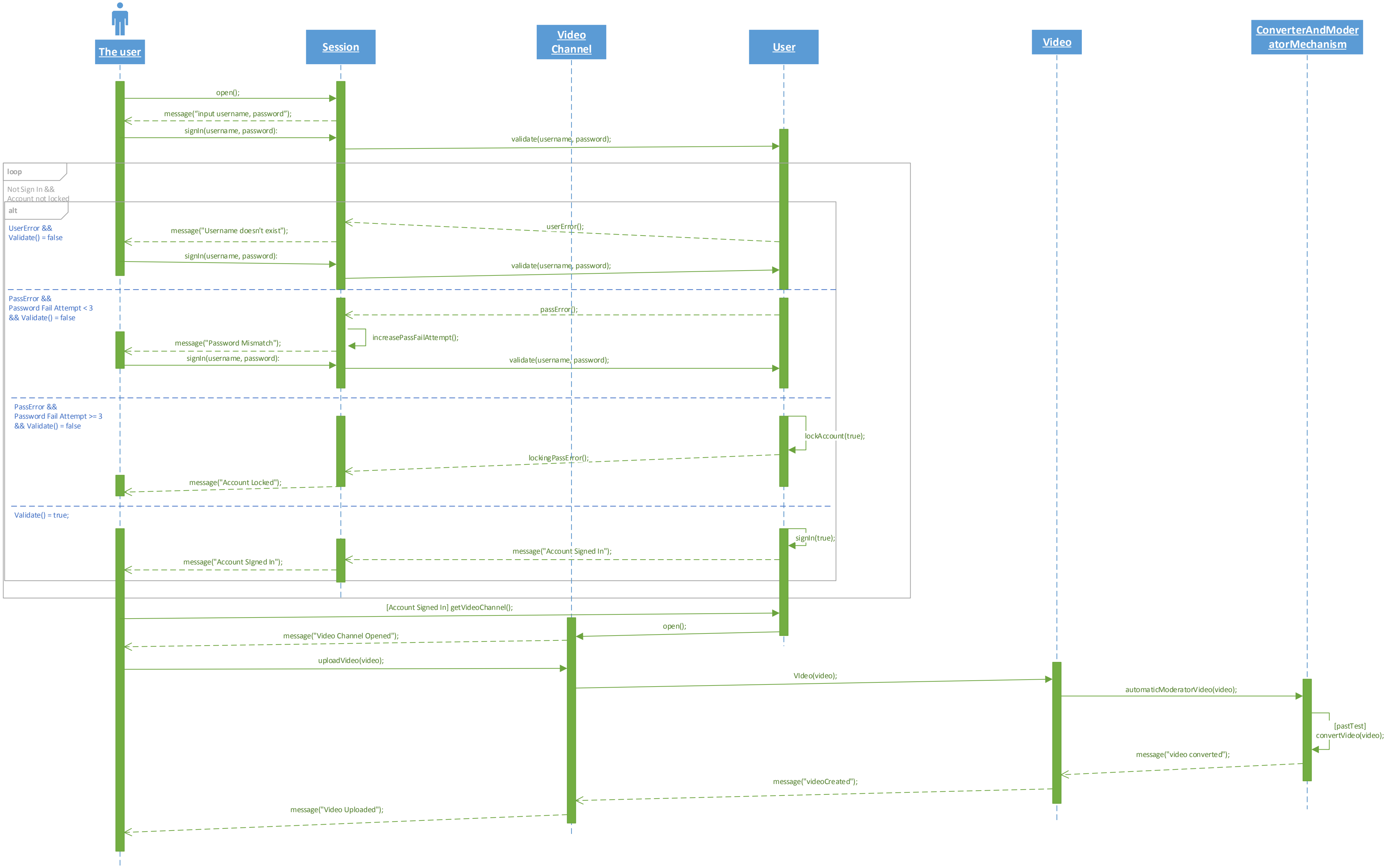
In every playlist object, there will be, a name attribute, a description attribute and a list attribute of multiple videos objects. The video objects should be the same object as in the video channel.

In every Image-class object, there will be a picture attribute.

In every Layout-class object, there will be a layout ID.

In every Video-class, there will be a name attribute, a description attribute and a view count attribute.

Sequence Diagram


The user
Session
Video Channel
User
Video
ConverterAndModeratorMechanism
open();
message("input username, password");
signIn(username, password):
validate(username, password);
loop
Not Sign In && Account not locked
alt
UserError && Validate() = false
userError();
message("Username doesn't exist");
signIn(username, password):
validate(username, password);
PassError && Password Fail Attempt < 3 && Validate() = false
passError();
increasePassFailAttempt();
message("Password Mismatch");
signIn(username, password):
validate(username, password);
PassError && Password Fail Attempt >= 3 && Validate() = false
lockAccount(true);
lockingPassError();
message("Account Locked");
Validate() = true;
signIn(true);
message("Account Signed In");
message("Account SIgned In");
[Account Signed In] getVideoChannel();
open();
message("Video Channel Opened");
uploadVideo(video);
VIdeo(video);
automaticModeratorVideo(video);
[pastTest] convertVideo(video);
message("video converted");
message("videoCreated");
message("Video Uploaded");

Notes for Sequence Diagram:

The scenario for the sequence diagram is the user trying to logging to the his (or her) account and then posting a video. The video should be valid and pass the automatic moderation test. When the user accesses the user object through a session, the session will display relevant fields to fills in with messages requiring the user to input username and password. After the user has finished filling in and send back the information, the session will ask the user object to validate the combination. While the user has not been able to log in yet, there are 4 cases that can happen.

+ The first case is when the username is wrong, if such thing happens, the user object will tell the session to notify the user such case and the user can try to sign in with different details.

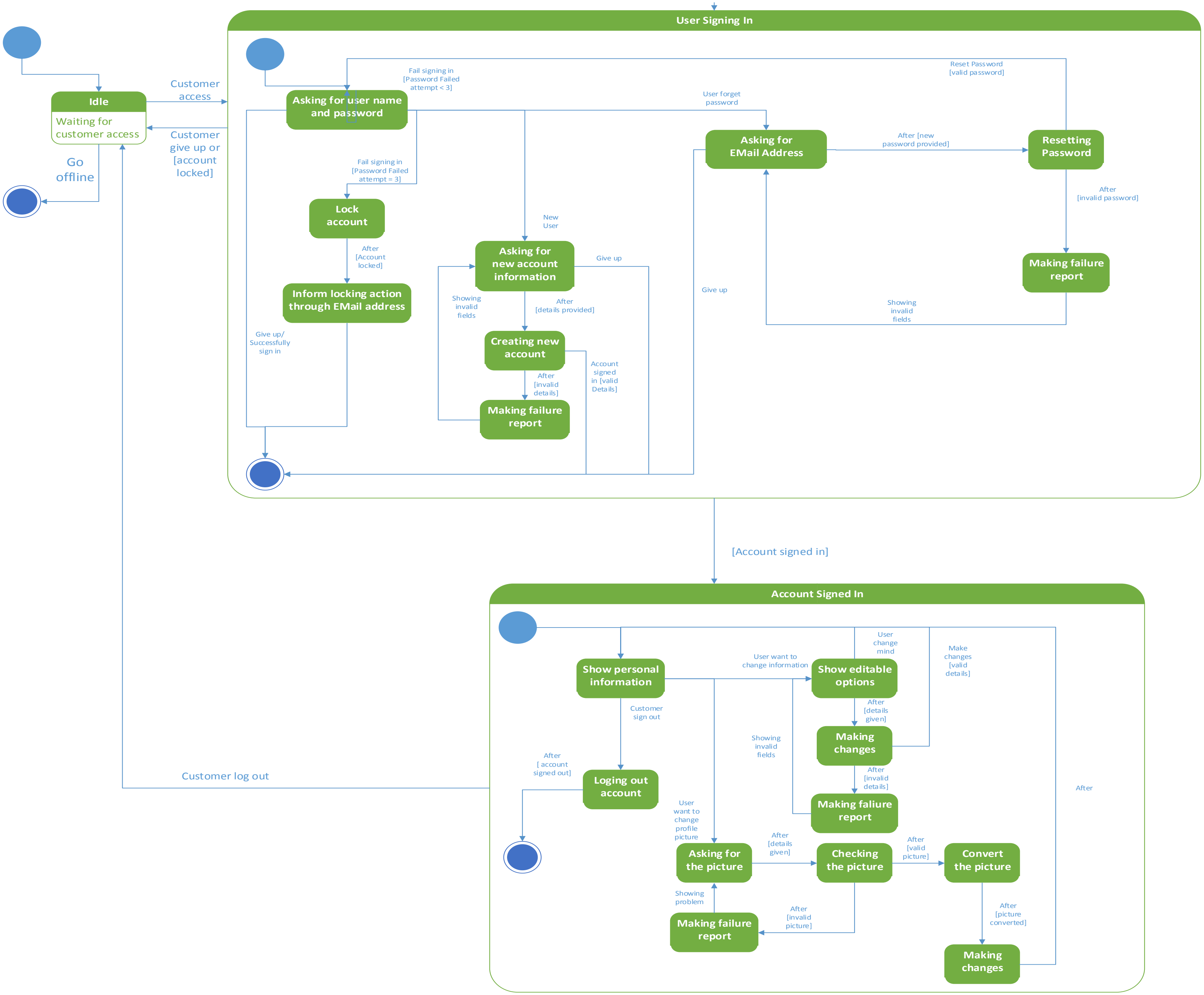
+ The second case is when the username is right but the password is wrong and the wrong password attempt is less than 3, then the user object will tell the session to notify the users about such case and the user can try to sign in with different details.

+ The third case is like the second case, however with 3 failed attempts; the user object then would lock the account and notify the user.

+ The final case is when the user successfully signs in, the user object would open a new session for the user and notify the user about the case.

If the user has successfully signed in, the user then can get access to the video channel page by asking the user object, the user object should contact the video channel object then. After that, the user can try to upload a video to the video channel. The video channel then would try to create a new Video object. The Video object will receive the video uploaded, ask the converter and moderator object to test the video. The video should then pass the test and get converted appropriately. The converter and moderator object then notify the video object. The video object then notifies the channel video object. Lastly, the user should be notified about this.

State Diagram



Notes for State Diagram:

The scenario for the state diagram is the user trying to logging to the his (or her) account and change information and there are three main big states.

The second state is idling state. When the system is in this state, it is waiting for the user to access. If a user access the system, the state changes to the third state. If the system goes offline, it will go to it final state.

The third state is a composite state about the user signing in. The second sub state will be asking for username and password. From this state, there are 5 ways to go:

+ If the user fail signing in and the password fail attempt is less than 3, the system will continue staying in the sub second state.

+ If the user fail signing in and the password fail attempt is 3, the system will lock the account and inform the user about it. After that, it goes to the final sub state.

+ If the user is new, the system will ask for new account details. After details are provided, the system will try to create the new account. If there are invalid details, the system will report the failure and ask the user to fill in again. If the details are all valid, the system will create the new account, sign the user in that new account and go to the final sub state.

+ If the user forgets the password, the system will ask for the E-Mail address. If the user does not want to provide the detail and give up, the system goes to the final sub state. After details are provided, the system will try to reset the password. If the detail is invalid, the system will report the failure and ask the user to fill in again. If the detail is valid, the system will reset the password, and go to the second sub state.

+ If the user does not want to try sign in and give up, the system will go to the final sub state.

After the third state, if the user gives up or the account is locked, the system will go back to the idling state. If the user successfully signs in, the system goes to the fourth state.

The fourth state is a composite state about the user changing account information or the user changing profile picture. The second sub state will be showing the account information. If the user signs out, the system will go to the logging out account state and then the final sub state.

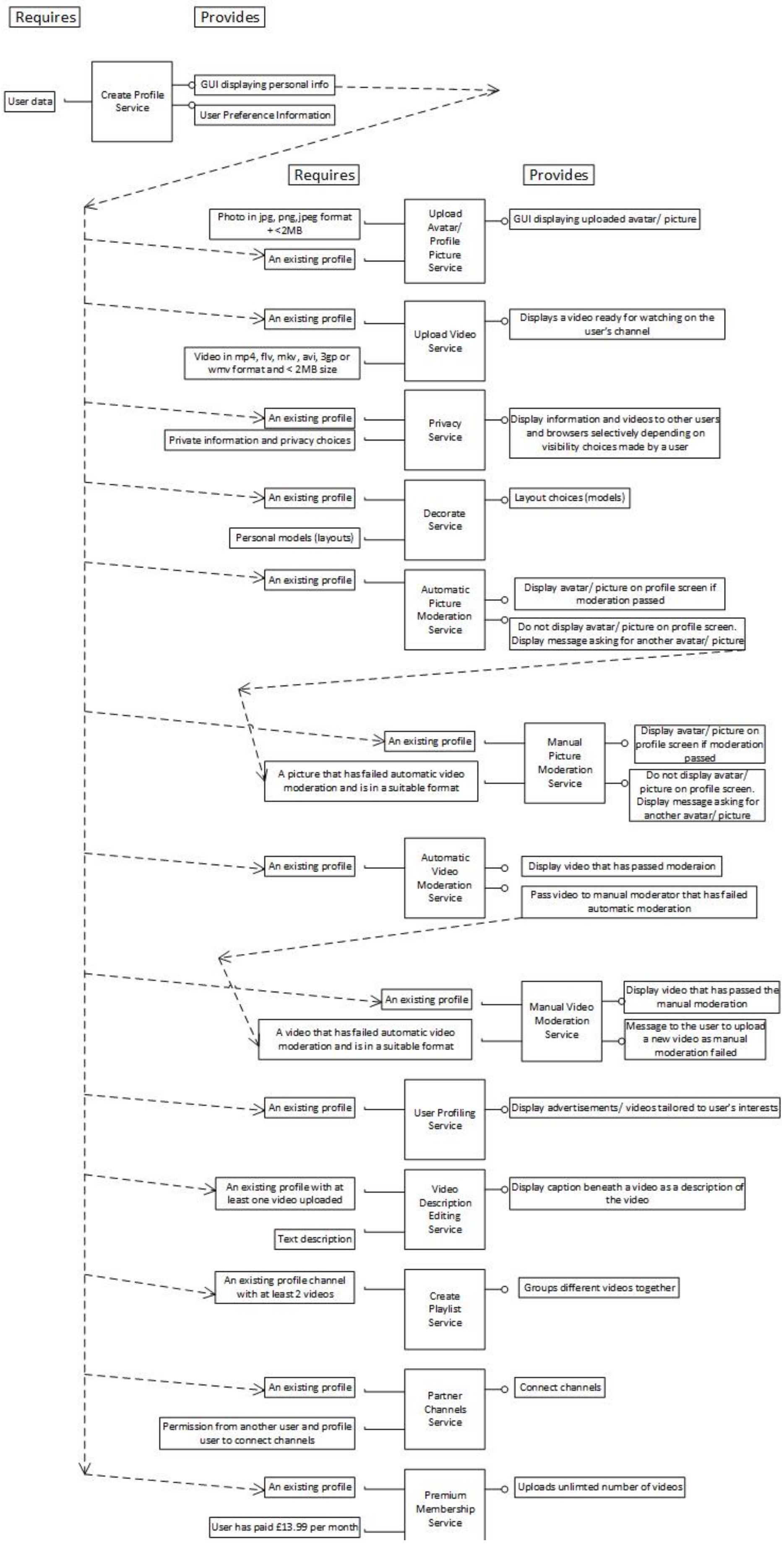
If the user wants to change the account information, the system will be showing all the editing options. After this, if the user changes his (or her) mind, the system goes back to the second sub state; if the user changes the details, the system will make changes. After the making change state, if the details are invalid, the system go back to showing the editing options with showing the invalid fields; if the details are valid, the system will change the details and go back to the second sub state.

If the user wants to change the profile picture, the system will ask for the picture. After the user upload the picture, the system will check for the validity of the picture. If the picture is invalid, the system will make failure report, tell the user and ask for the picture again. If the picture is valid, the system will convert the picture suitably and make change accordingly. After that, the system will go back to the second sub state.

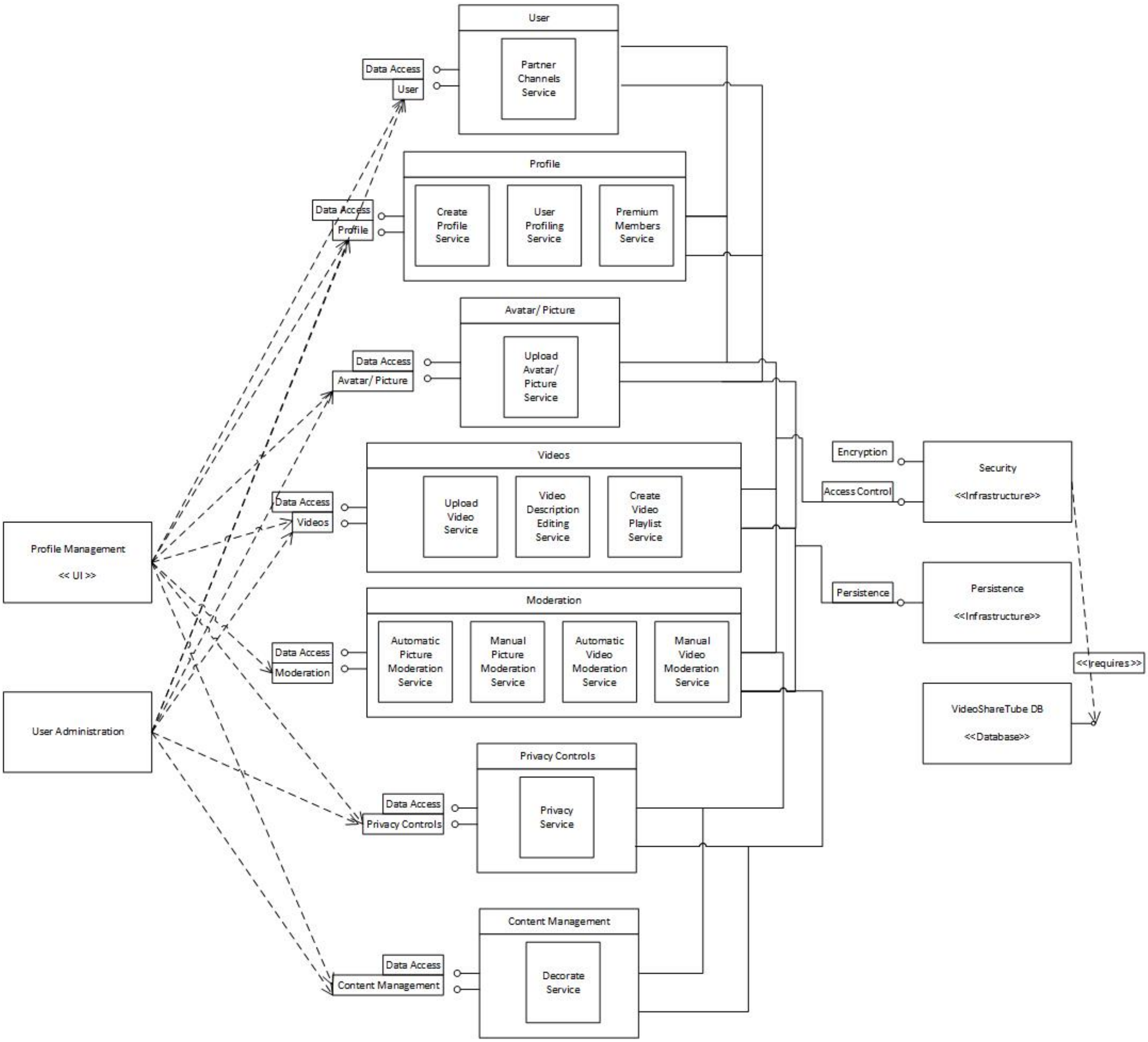
After the fourth state, the system will go back to the idling state.

Component Diagrams - Micro and Macro Architectures

Component Diagram for Micro-Services Architecture

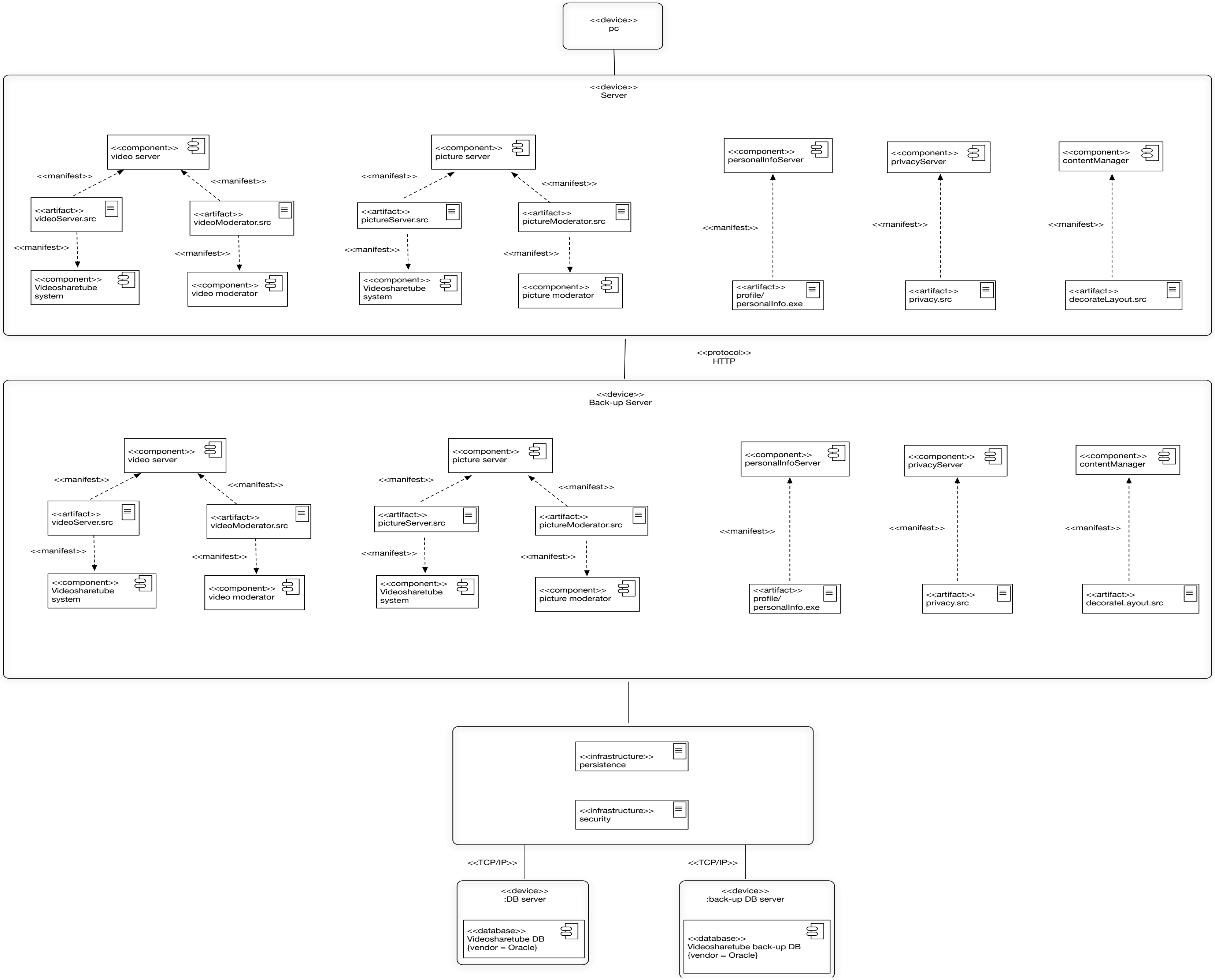

Requires
Provides
User data
Create Profile Service
GUI displaying personal info
User Preference Information
Requires
Provides
Photo in jpg, png, jpeg format + <2MB
An existing profile
Upload Avatar/ Profile Picture Service
GUI displaying uploaded avatar/ picture
An existing profile
Video in mp4, flv, mkv, avi, 3gp or wmv format and < 2MB size
Upload Video Service
Displays a video ready for watching on the user's channel
An existing profile
Private information and privacy choices
Privacy Service
Display information and videos to other users and browsers selectively depending on visibility choices made by a user
An existing profile
Personal models (layouts)
Decorate Service
Layout choices (models)
An existing profile
Automatic Picture Moderation Service
Display avatar/ picture on profile screen if moderation passed
Do not display avatar/ picture on profile screen. Display message asking for another avatar/ picture
An existing profile
A picture that has failed automatic video moderation and is in a suitable format
Manual Picture Moderation Service
Display avatar/ picture on profile screen if moderation passed
Do not display avatar/ picture on profile screen. Display message asking for another avatar/ picture
An existing profile
Automatic Video Moderation Service
Display video that has passed moderaion
Pass video to manual moderator that has failed automatic moderation
An existing profile
A video that has failed automatic video moderation and is in a suitable format
Manual Video Moderation Service
Display video that has passed the manual moderation
Message to the user to upload a new video as manual moderation failed
An existing profile
User Profiling Service
Display advertisements/ videos tailored to user's interests
An existing profile with at least one video uploaded
Text description
Video Description Editing Service
Display caption beneath a video as a description of the video
An existing profile channel with at least 2 videos
Create Playlist Service
Groups different videos together
An existing profile
Permission from another user and profile user to connect channels
Partner Channels Service
Connect channels
An existing profile
User has paid £13.99 per month
Premium Membership Service
Uploads unlimted number of videos

Component Diagram for Macro-Services Architecture

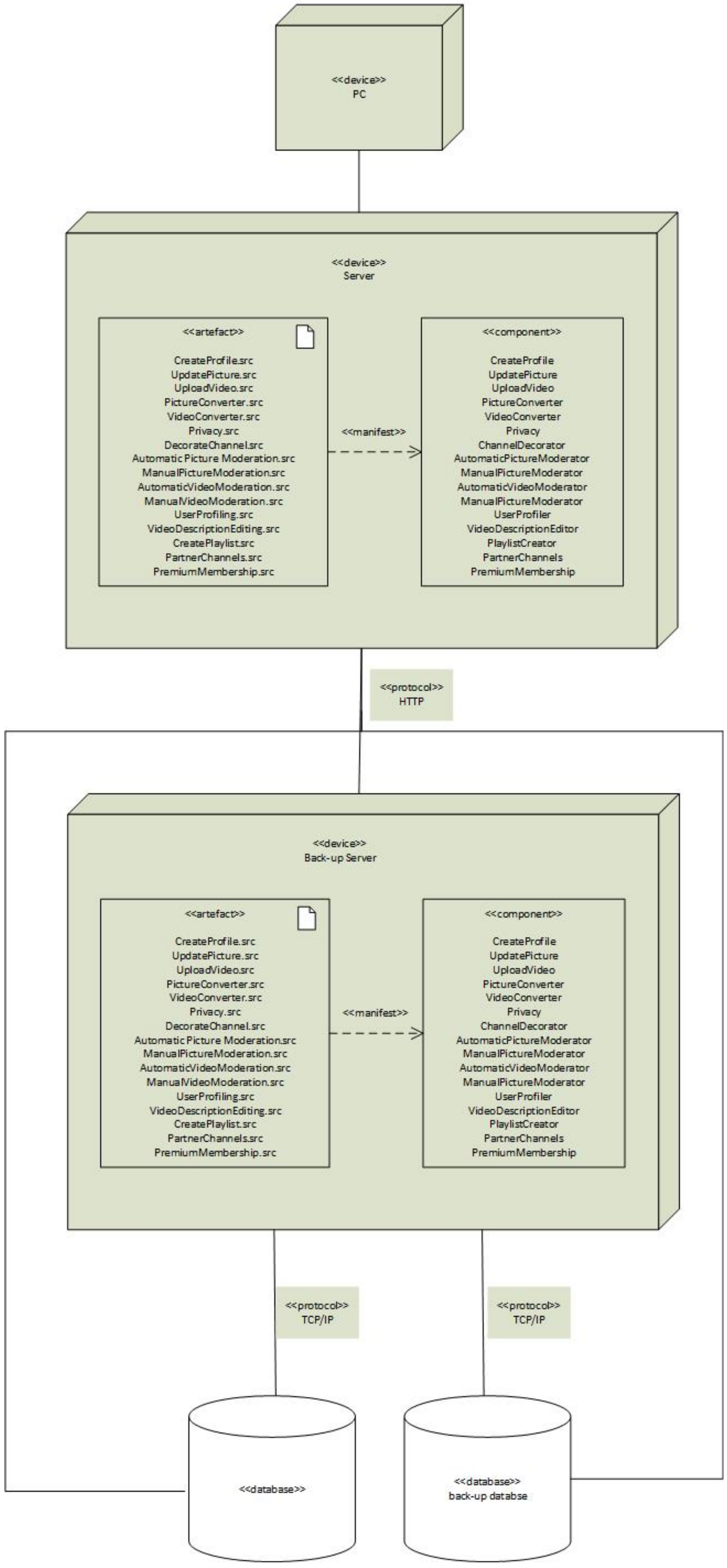


Deployment Diagrams – Micro and Macro Architectures

Deployment diagram with a microservices architecture



Deployment Diagram for Macro-Services Architecture



Appraisal of Designs for Two Architectures

Appraisal for Micro-services Architecture

Maintainability: This architecture was chosen as an important feature of Videosharetube is the ability to make changes to components as they arise. When components are divided into small services delivering a multitude of functions, the application is easy to maintain.

Availability: If something goes wrong in one component it does not affect the other components surrounding it and is therefore easier to fix. With increased availability there is however, decreased consistency.

Performance: Performance is reduced as there is a greater distance between components for interoperability.

Security: The architecture is layered with critical assets in the inner layers improving security. Security is easy to implement as a security breach in one component will only affect components within a group and no other surrounding components.

Safety: It is improved as safety-critical features exist in a small number of subsystems.
This architecture has a shared data model so sub-systems use the same data organisation so data interchange is efficient.

Scalability: It is more difficult to add new components to the system as individual component groups will need to be considered.

Modifiability: It is easier to deploy regular updates of the system than a macro-services architecture.
Micro-services support operating the system on multiple platforms. This enables the use of different languages for development allowing problems to be solved using the best language suitable. A disadvantage of this architecture is that it is important to get the services defined correctly when creating the system is started as it may be difficult to change this afterwards.

Fault tolerance: If the system fails in one area, a degree of the system will run even if it is a degraded version if other services around it fail.

Appraisal for Macro-services Architecture

Performance: This architecture was chosen as an important feature of Videosharetube is performance. Unlike the micro-services architecture, the macro-services architecture groups components together as a whole making interactions between individual components faster and more efficient as they are close together.

Maintainability: Maintainability is compromised as if there is a fault with one function, the rest of the components on the server are likely to be affected as they are interconnected.

Availability: If something goes wrong in one component it will affect the other components surrounding it and is therefore more difficult to fix. However, consistency is likely to be high.

Security: The architecture is layered with critical assets in the inner layers improving security. With all the components grouped together it is easier to make the system secure.

Safety: This is improved as safety-critical features exist in a small number of subsystems.
This architecture has a shared data model so sub-systems use the same data organisation so data interchange is efficient.

Scalability: The addition of new components is fairly simple with this architecture.

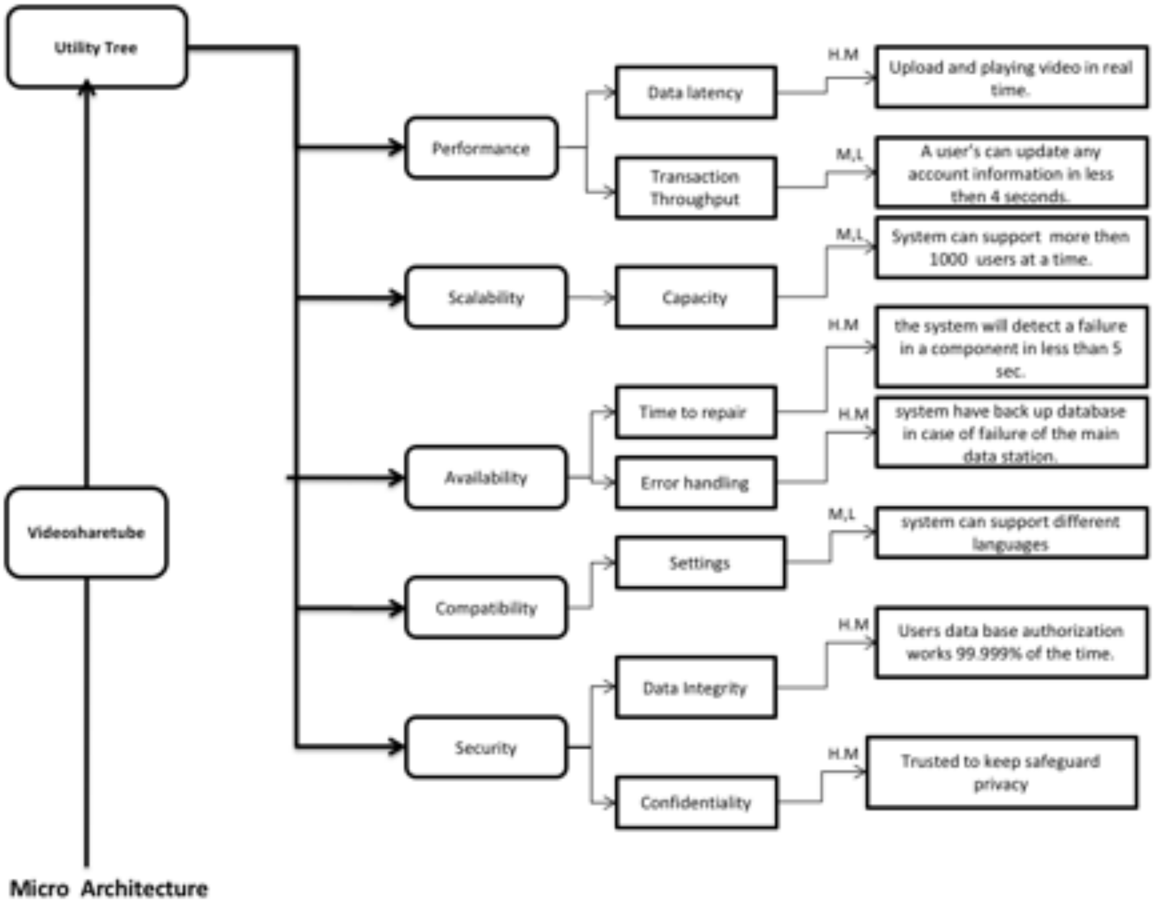
Modifiability: An advantage of macro-services architecture is that it is easier to change code as the services are in one place, whereas in a micro-services architecture if code needs to be changed associated services in a different place will need to be changed/ considered.

Fault tolerance: If the system fails in one area, the whole system may fail to run.

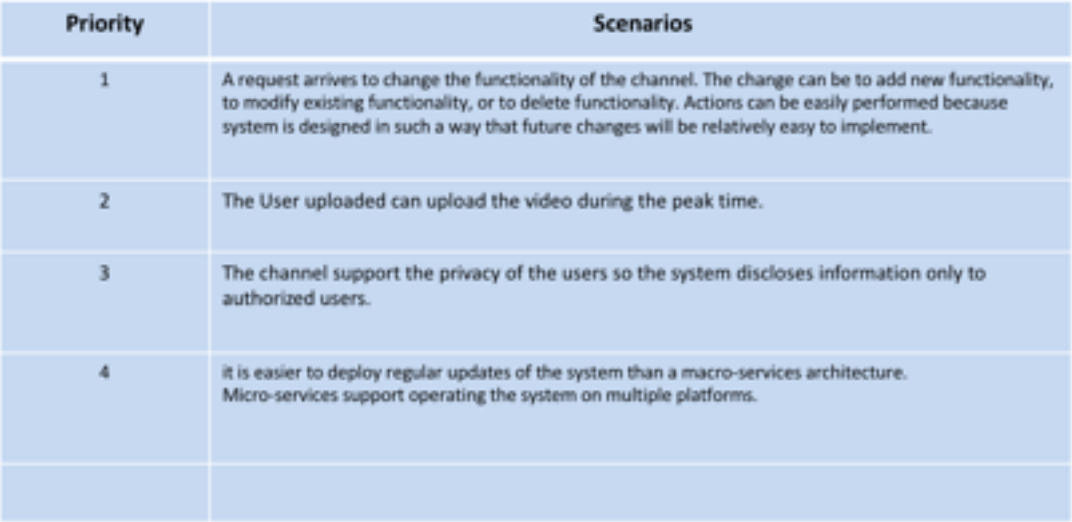
Architectural Tradeoffs Analysis Method (ATAM) Evaluation

The Videosharetube architectures represents the earliest design decisions. These decisions affect scalability, modifiability, security, real-time performance and compatibility of the application. They are the most critical to get right and the most difficult to change downstream in the development life cycle.

The right architecture can cover the way for successful system Videosharetube development. In this part our team is going to compare two architectures Micro and Macro so we can choose the better one for the project Videosharetube.

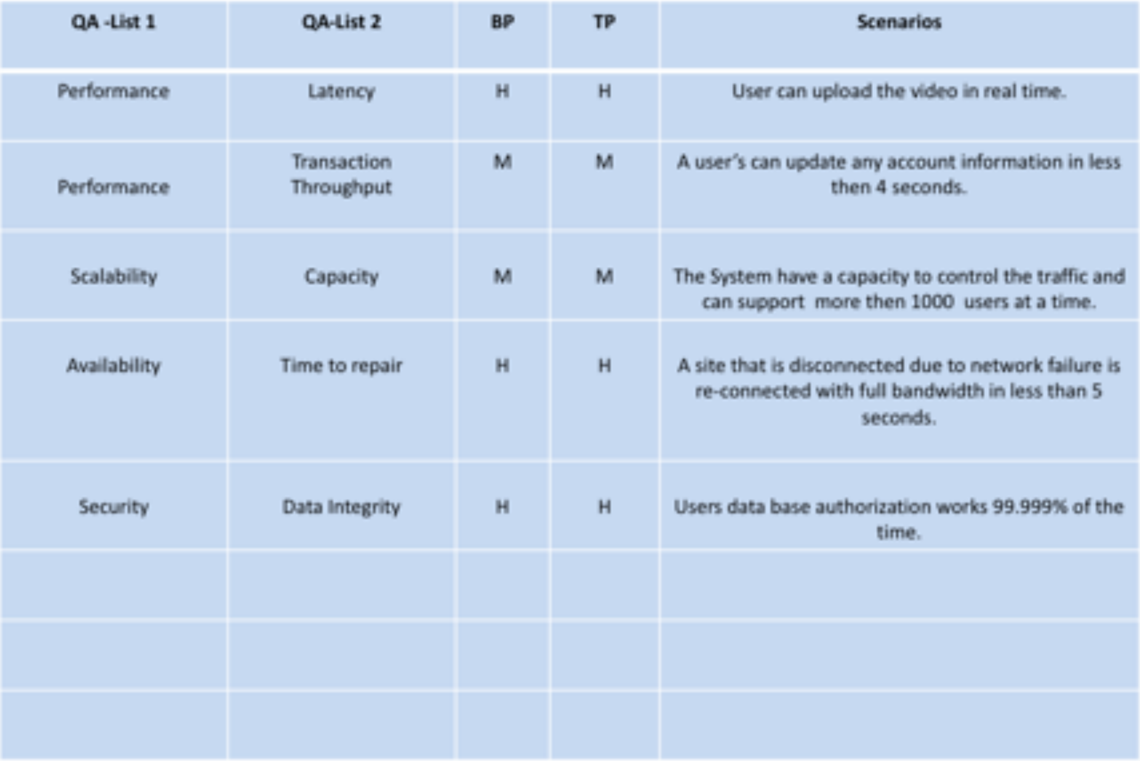

Utility Tree
Videosharetube
Micro Architecture
Performance
Data latency
H.M
Upload and playing video in real time.
Transaction Throughput
M,L
A user's can update any account information in less then 4 seconds.
Scalability
Capacity
M,L
System can support more then 1000 users at a time.
Availability
Time to repair
H.M
the system will detect a failure in a component in less than 5 sec.
Error handling
H.M
system have back up database in case of failure of the main data station.
Compatibility
Settings
M,L
system can support different languages
Security
Data Integrity
H.M
Users data base authorization works 99.999% of the time.
Confidentiality
H.M
Trusted to keep safeguard privacy

Scenarios for Macro Architecture

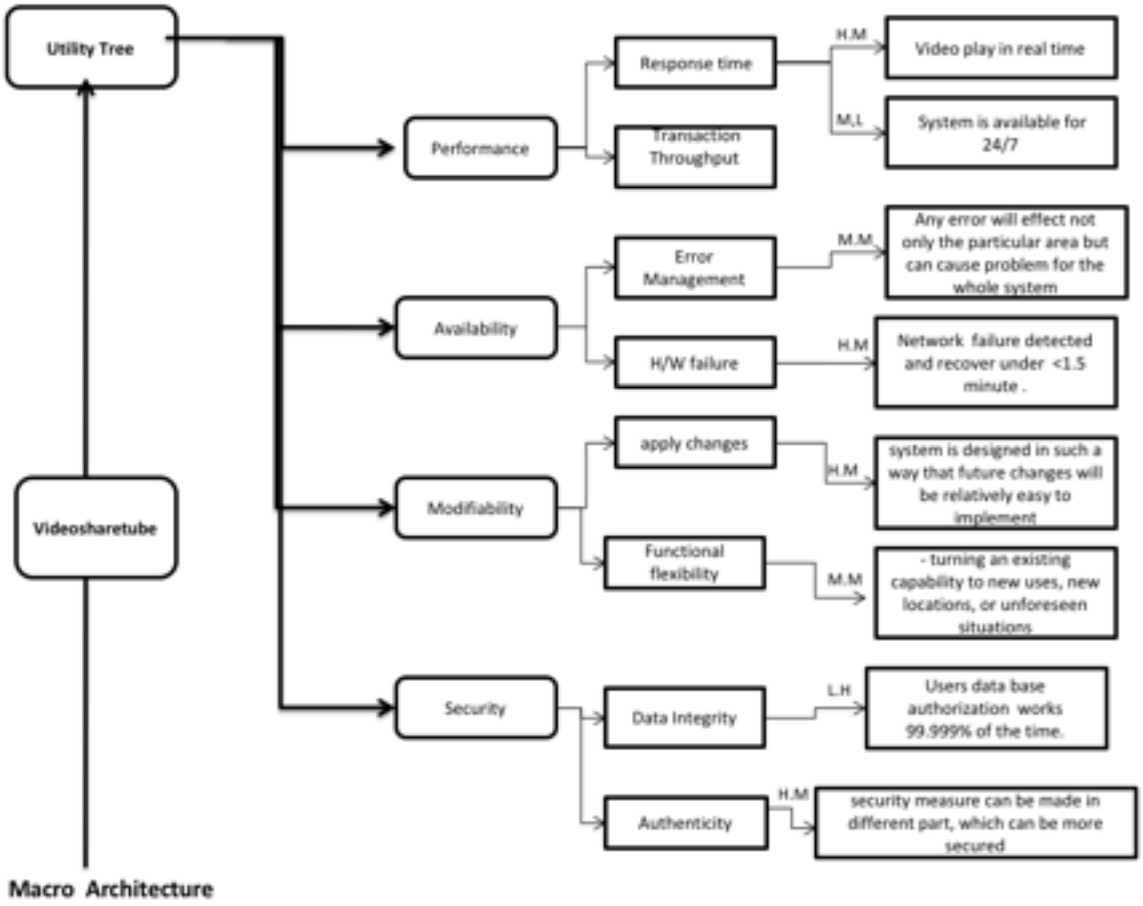


Priority	Scenarios
1	A request arrives to change the functionality of the channel. The change can be to add new functionality, to modify existing functionality, or to delete functionality. Actions can be easily performed because system is designed in such a way that future changes will be relatively easy to implement.
2	The User uploaded can upload the video during the peak time.
3	The channel support the privacy of the users so the system discloses information only to authorized users.
4	it is easier to deploy regular updates of the system than a macro-services architecture. Micro-services support operating the system on multiple platforms.

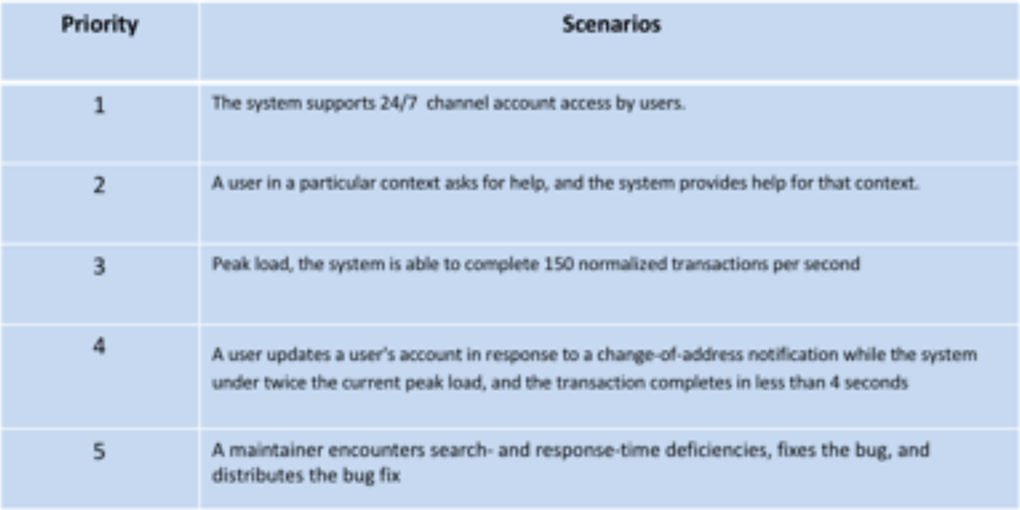
Quality Attribute Tree for Micro Architecture



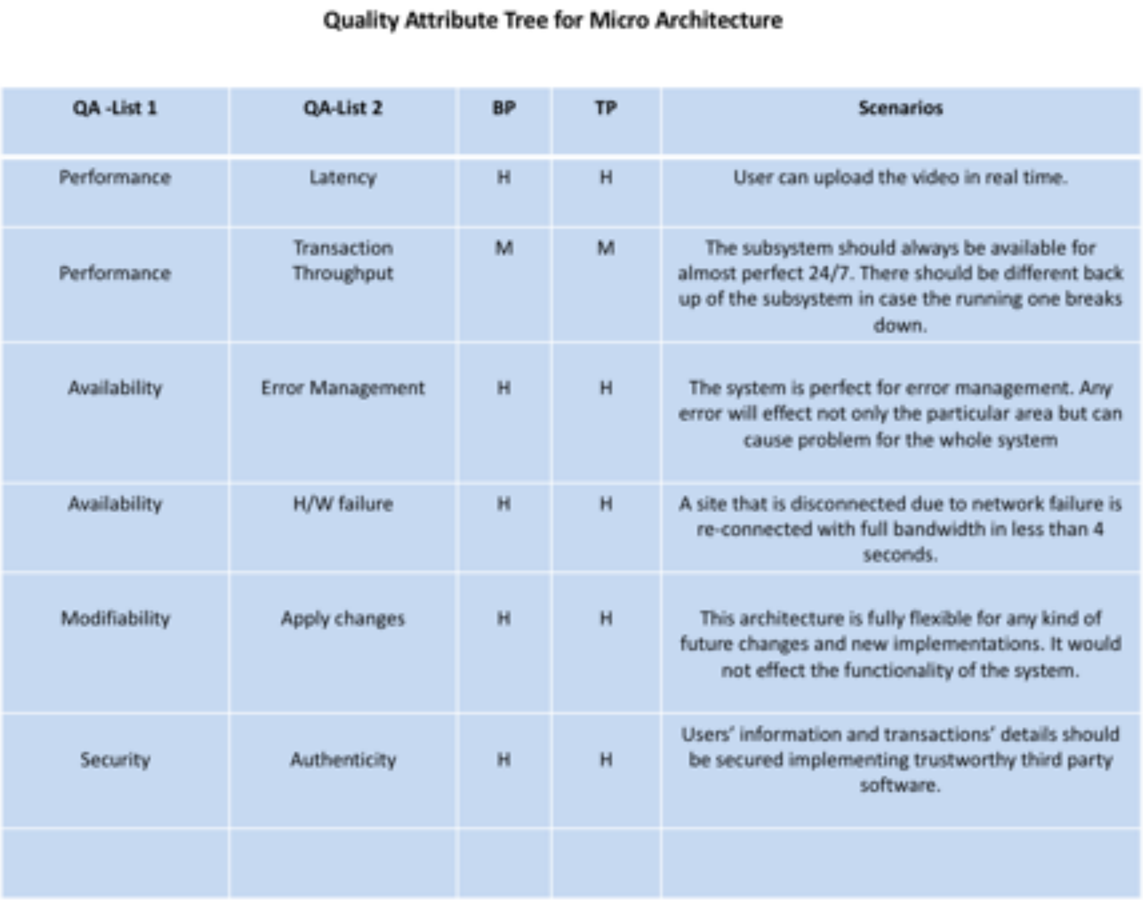
QA -List 1	QA-List 2	BP	TP	Scenarios
Performance	Latency	H	H	User can upload the video in real time.
Performance	Transaction Throughput	M	M	A user's can update any account information in less then 4 seconds.
Scalability	Capacity	M	M	The System have a capacity to control the traffic and can support more then 1000 users at a time.
Availability	Time to repair	H	H	A site that is disconnected due to network failure is re-connected with full bandwidth in less than 5 seconds.
Security	Data Integrity	H	H	Users data base authorization works 99.999% of the time.


Utility Tree
Videosharetube
Macro Architecture
Performance
Response time
Transaction Throughput
H.M
Video play in real time
M,L
System is available for 24/7
Availability
Error Management
M.M
Any error will effect not only the particular area but can cause problem for the whole system
H/W failure
H.M
Network failure detected and recover under <1.5 minute .
Modifiability
apply changes
H.M
system is designed in such a way that future changes will be relatively easy to implement
Functional flexibility
M.M
- turning an existing capability to new uses, new locations, or unforeseen situations
Security
Data Integrity
L.H
Users data base authorization works 99.999% of the time.
Authenticity
H.M
security measure can be made in different part, which can be more secured

Scenarios for Micro Architecture

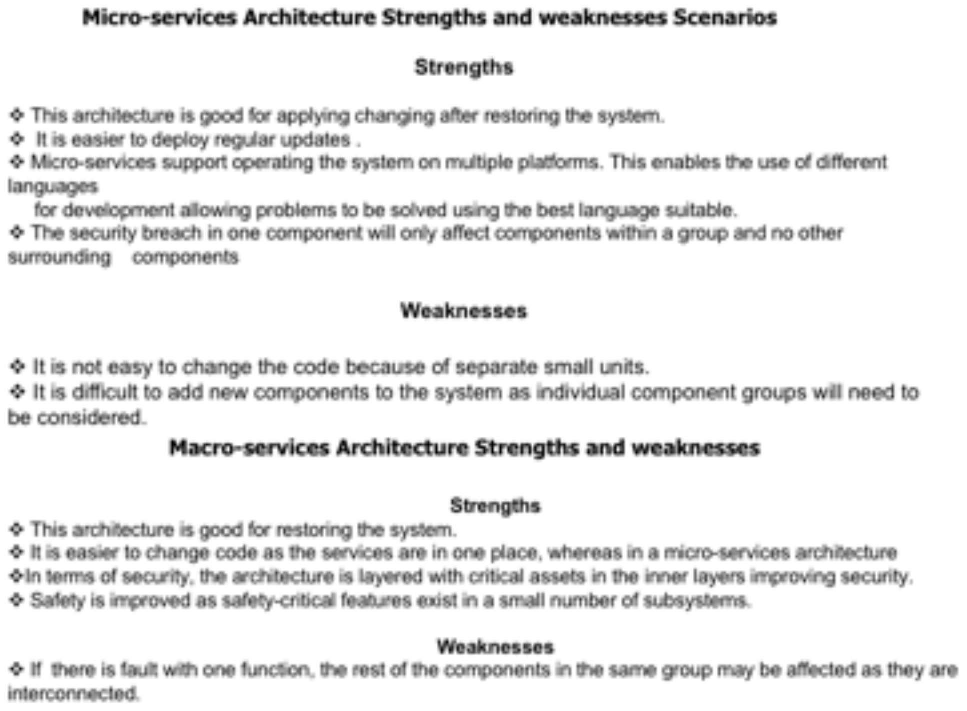


Priority	Scenarios
1	The system supports 24/7 channel account access by users.
2	A user in a particular context asks for help, and the system provides help for that context.
3	Peak load, the system is able to complete 150 normalized transactions per second
4	A user updates a user's account in response to a change-of-address notification while the system under twice the current peak load, and the transaction completes in less than 4 seconds
5	A maintainer encounters search- and response-time deficiencies, fixes the bug, and distributes the bug fix



Quality Attribute Tree for Micro Architecture

QA -List 1	QA-List 2	BP	TP	Scenarios
Performance	Latency	H	H	User can upload the video in real time.
Performance	Transaction Throughput	M	M	The subsystem should always be available for almost perfect 24/7. There should be different back up of the subsystem in case the running one breaks down.
Availability	Error Management	H	H	The system is perfect for error management. Any error will effect not only the particular area but can cause problem for the whole system
Availability	H/W failure	H	H	A site that is disconnected due to network failure is re-connected with full bandwidth in less than 4 seconds.
Modifiability	Apply changes	H	H	This architecture is fully flexible for any kind of future changes and new implementations. It would not effect the functionality of the system.
Security	Authenticity	H	H	Users' information and transactions' details should be secured implementing trustworthy third party software.



Micro-services Architecture Strengths and weaknesses Scenarios

Strengths

❖ This architecture is good for applying changing after restoring the system.
❖ It is easier to deploy regular updates .
❖ Micro-services support operating the system on multiple platforms. This enables the use of different languages
for development allowing problems to be solved using the best language suitable.
❖ The security breach in one component will only affect components within a group and no other surrounding components

Weaknesses

❖ It is not easy to change the code because of separate small units.
❖ It is difficult to add new components to the system as individual component groups will need to be considered.

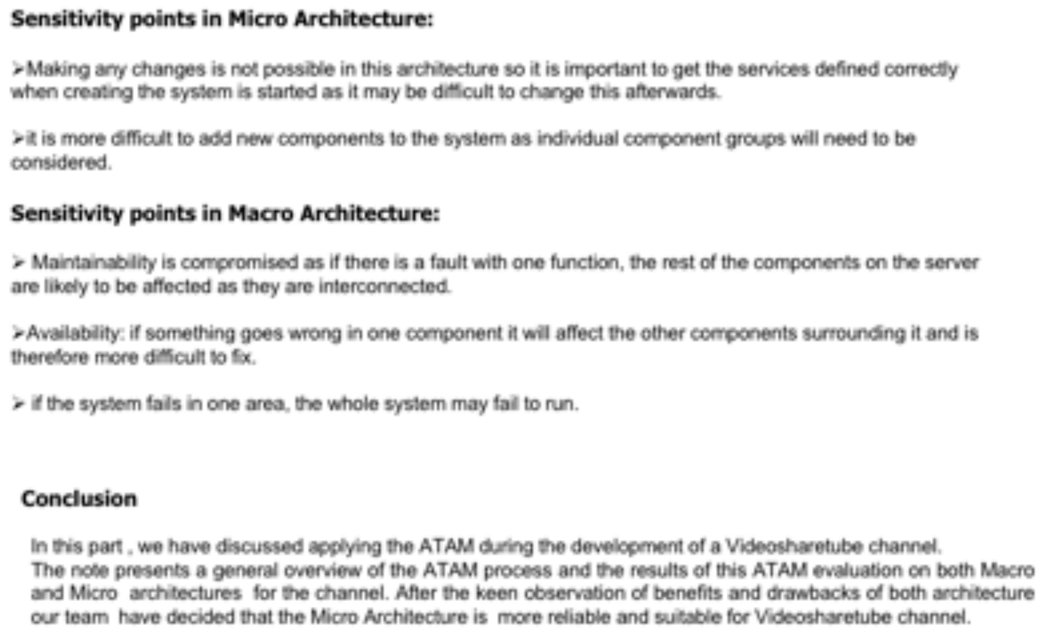
Macro-services Architecture Strengths and weaknesses

Strengths

❖ This architecture is good for restoring the system.
❖ It is easier to change code as the services are in one place, whereas in a micro-services architecture
❖In terms of security, the architecture is layered with critical assets in the inner layers improving security.
❖ Safety is improved as safety-critical features exist in a small number of subsystems.

Weaknesses

❖ If there is fault with one function, the rest of the components in the same group may be affected as they are interconnected.



Sensitivity points in Micro Architecture:

➢Making any changes is not possible in this architecture so it is important to get the services defined correctly when creating the system is started as it may be difficult to change this afterwards.

➢it is more difficult to add new components to the system as individual component groups will need to be considered.

Sensitivity points in Macro Architecture:

➢ Maintainability is compromised as if there is a fault with one function, the rest of the components on the server are likely to be affected as they are interconnected.

➢Availability: if something goes wrong in one component it will affect the other components surrounding it and is therefore more difficult to fix.

➢ if the system fails in one area, the whole system may fail to run.

Conclusion

In this part , we have discussed applying the ATAM during the development of a Videosharetube channel. The note presents a general overview of the ATAM process and the results of this ATAM evaluation on both Macro and Micro architectures for the channel. After the keen observation of benefits and drawbacks of both architecture our team have decided that the Micro Architecture is more reliable and suitable for Videosharetube channel.

Additional MSc Questions

Elaborate on the compliance requirements for monitoring and preventing digital addiction through gamification. Identify relevant trade-offs.

The digital addiction through gamification has become a very serious and common phenomenon in this day and age. Gamification is very useful tools for improving engagement of the user but at the same time, it has many physical and psychological side effects like sleeping disorder, stress, anxiety and depression. Videosharetube will help the users by developing the following system to monitor and prevent the digital addition.

Rewards

Videosharetube will use rewards (extra points, badges or avatars, etc.) as a motivation and behavioural changes technique. It is one of the best implementations of gamification. The users should be able to see the avatar displayed with the awareness of time usages or users will also get extra points and badges for staying under a time limit. There will be a complete guideline for the addictive aspects of the app so users will be completely aware of the positive and negative impacts and will be able to get benefits fully from the application app.

Set Up the time limit

The Videosharetube will monitor the activities of the users by keeping the session records. If the users will spend too long on a site, a warning message will appear on the screen. The user will be able to use the app maximum for 4 hours each session and will start getting reminders when they will reach 70% of their allotted time. The message or beep will appear on the screen to alert the user every 45 minutes until session will be over. The user should be able to set the time limit in customise settings and will be able to choose how many times a week they will be able to use the app and they will be able to select the specific days and time. The users will be able to set up reminder message for their phone so they will receive alerts.

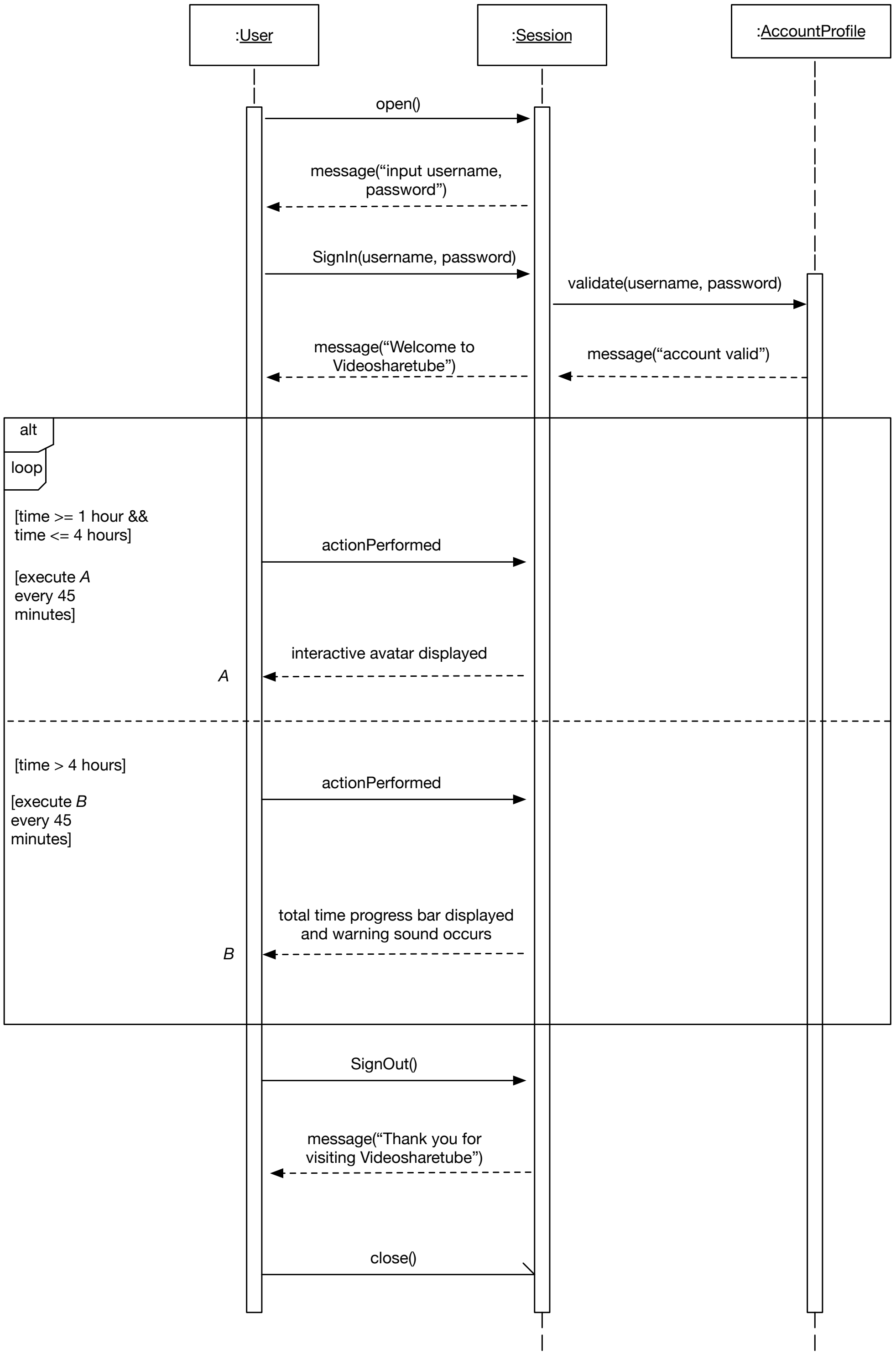
Discover addictive aspects of the app

In the help and support section, the users should be able to see the complete manual for the addictive aspects of the app and recommended a plan for addicted users.

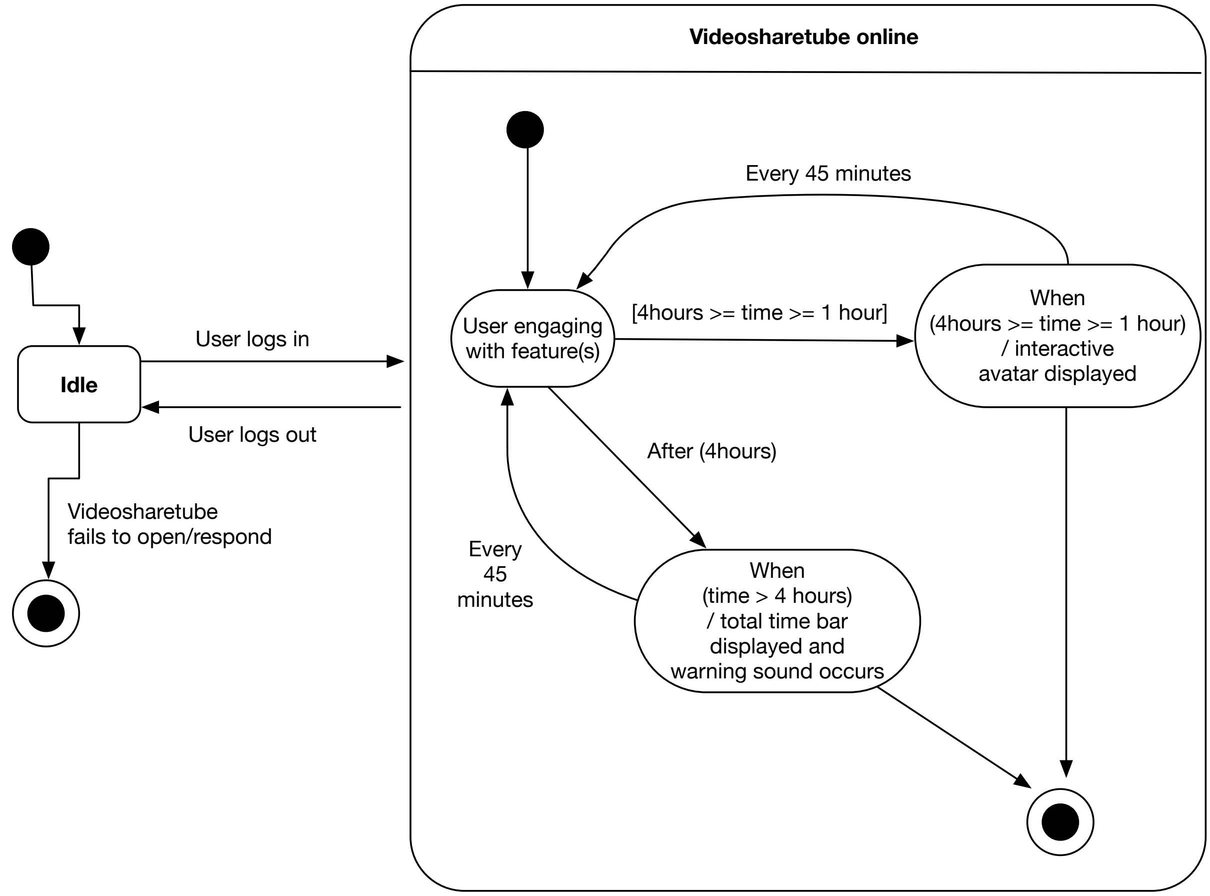
Draft a non-trivial sequence and state diagrams corresponding to the above scenario. Revisit your architecture – component diagram- and enrich it with your solution for monitoring Videosharetube digital addiction.

Below are edited versions of the sequence diagram and state diagram that deals with the scenario described above in effort to overcome the issue.

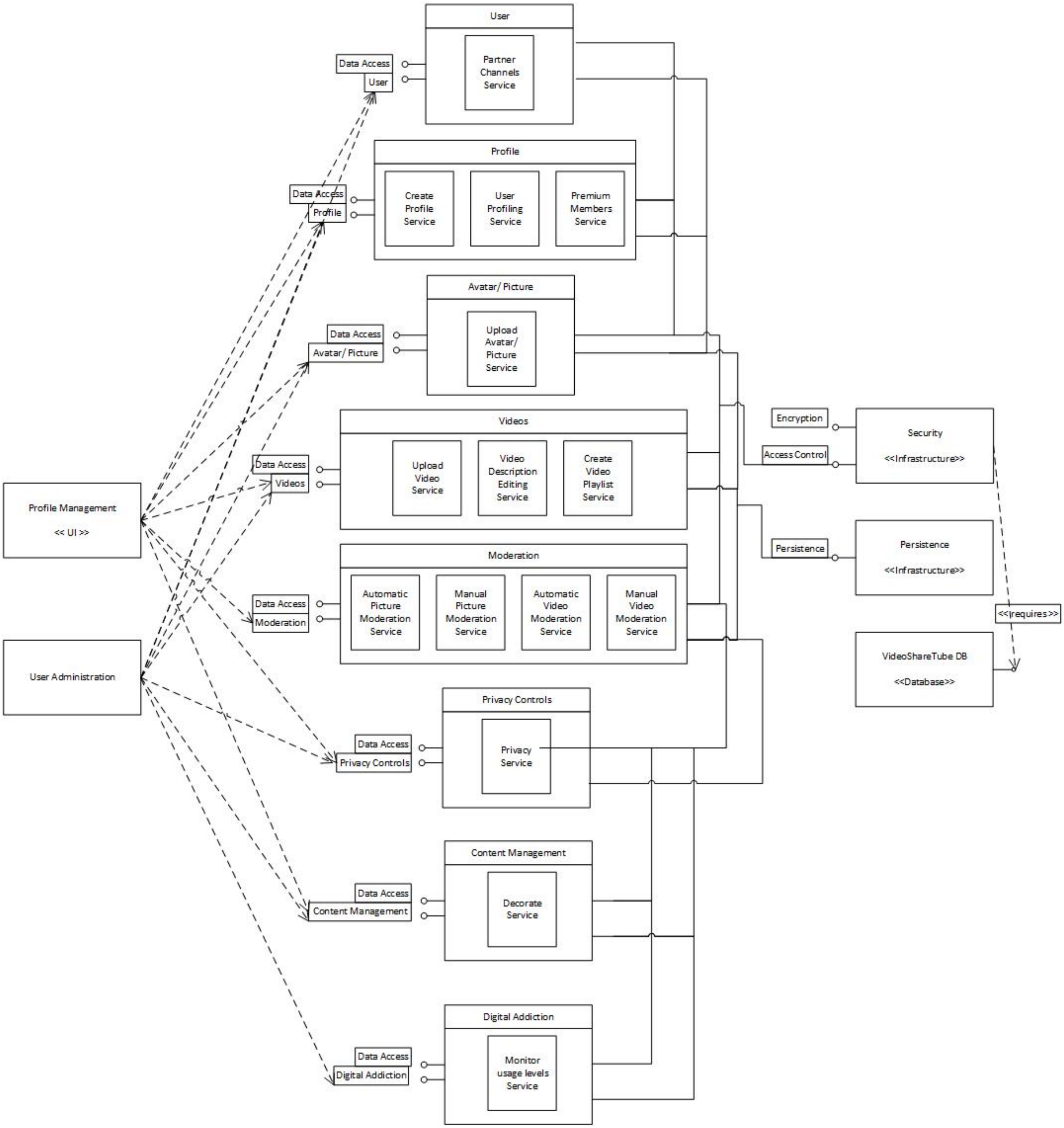
Sequence diagram


:User
:Session
:AccountProfile
open()
message("input username, password")
SignIn(username, password)
validate(username, password)
message("Welcome to Videosharetube")
message("account valid")
alt
loop
[time >= 1 hour && time <= 4 hours]
[execute A every 45 minutes]
actionPerformed
interactive avatar displayed
A
[time > 4 hours]
[execute B every 45 minutes]
actionPerformed
total time progress bar displayed and warning sound occurs
B
SignOut()
message("Thank you for visiting Videosharetube")
close()

State diagram



Component Diagram for Micro-Services Architecture



Discuss relevant architecture tradeoffs related to your solution.

The solution for monitoring Videosharetube digital addiction is to add it as a new component in the Micro-services architecture alongside the existing services. It exists as a “Monitor Usage Levels Service” within the group “Digital Addiction”. The high level of performance is maintained as the component is contained within a group. Maintainability should remain the same as before as there is only one component in the Digital Addiction group. If there is a failure in this service it should not affect other grouped areas but if new services are added to the group maintainability will be compromised as a failure in one service will affect other services within the same group. Changing code in this area should be easy as the code is contained within one place.

Can you think of four possible test cases that build on scenarios for testing software for addiction? Be pragmatic yet creative in your suggestion. Briefly discuss how software engineering practices can benefit from your suggestion.

Digital addiction is becoming an ever increasing trend among today’s generation, with the use of digital devices and software at an all-time high. Although there are now schemes to try and “detox” [3] digital addicts, as well as an increasing importance put on digital education[2], ultimately the responsibility to check for possible instances of digital addiction has fallen upon software engineers, as part of the software development process.

Software may be tested for causing digital addiction for a number of reasons. It may be due to the amount of time users spend using the software, or the fact that users may neglect other work or responsibilities in order to use the software. Concerned relatives or friends of those thought to be suffering from digital addictions often state that users become defensive about their use of the software, and cut their physical social interactions down, in lieu of those that they can have online. All of these are serious consequences of digital addiction that can lead to other health[1] related issues, such as sleep deprivation and irritability when confronted, as well as less obvious problems, such as heart disease[4].

One test case that can be used to test software for addiction is to see if the software has a time limit associated with it. One of the key concerns for those who try and combat digital addiction is the sheer amount of time that is spent by users using the software. For an application such as Videosharetube and the like, this is often too common. Users can spend excessive time watching videos, without realising it, neglecting other responsibilities.

Another test case would be to check the software to see if it has any features associated with digital addiction. For example, social media gives users the ability to send private messages to one another via a messaging system. These interactions with other users online can cause digital addiction. If the software does implement a system such as private messaging, it would perhaps be a beneficial idea to limit the number of interactions a user can have with another user within a given time period, i.e. a certain message limit within 24 hours. An example of where this concept works well would be Twitter, and their temporary banning of users who tweet too often. Tweeters refer to this as being “put in Twitter jail”, as the user is not allowed to send tweets for a certain time period.

Software could also be tested for digital addiction by examining the demographics of users. Digital addiction is most common among teenagers and young children, and by looking at the software insights, key predictions can be made as to whether software is likely to cause digital addiction. These demographics can then be compared to the functions that the software offers, to see if they coincide with those that have been documented to cause digital addiction for those particular demographics.

Finally, another test case that could be used would be to develop several ways of making a function, and then test which one is most likely to cause digital addiction, and which one is least likely to cause digital addiction. For example, say a messaging service needed to be implemented as part of the requirements. Software engineers could look at several ways of implementing that messaging service, without compromising on the messaging service itself. Perhaps they would choose a simple user interface which is less appealing, which means less time is spent on there, or that message length is limited. Software engineers would provide several solutions to each function, instead of just one, so that they are able to research which option prevents or decreases the chances of digital addiction, as opposed to having just one solution and working around it.

Software engineering practices can benefit from these practices as they will reduce instances of digital addiction. Digital addiction will be much harder to handle after software has been released, and so by implementing these practices within the software engineering process, we will be able to take greater preventative measures which will have a greater effect in decreasing levels of digital addiction.

References

1. Keen A. Digital addiction is the world's next great health crisis [Internet]. The Next Web. 2015 [cited 15 January 2016]. Available from: http://thenextweb.com/insider/2015/12/21/digital-addiction-is-the-worlds-next-great-health-crisis/#gref

2. IBNLive. 2016: Digital detox, digital literacy, digital adaptability [Internet]. 2016 [cited 15 January 2016]. Available from: http://www.ibnlive.com/news/tech/2016-digital-detox-digital-literacy-digital-adaptability-1186664.html

3. Examiner.com. Digital detox: Internet addiction may require rehab [Internet]. 2015 [cited 15 January 2016]. Available from: http://www.examiner.com/article/digital-detox-internet-addiction-may-require-rehab

4. Clive J, Allen A. Technology junkies: do families need a digital detox? - Journalism & News from Bournemouth University [Internet]. Buzz. 2016 [cited 15 January 2016]. Available from: http://buzz.bournemouth.ac.uk/technology-junkies-do-families-need-a-digital-detox/