WatchTime

YouTube Reimagined

# **Description**

## **The reputation of YouTube**

In our day and age, YouTube is the most used entertainment platform, with content made by people for people.

However, YouTube has a very dark side of abusing new/small content creators, having systems that can be abused by the big corporations where 1 second of audio can completely remove the revenue from a video without warning, and many more.

Not only that but the trending algorithms used by YouTube reward short and lazy content posted frequently, meaning that the overall quality of the content on YouTube is going down in order to be picked up more by the algorithm.

## **Target demographic.**

The target of this application is small and new creators from YouTube that have been denied the chance to showcase their content by the unfair algorithms and new creators that want a better chance at recognition.

## **Proposed changes in order to make WatchTime better than YouTube**

1. **Reward creators for high quality content.**

In order to reward the creators who put time and effort into their content, a new system needs to be implemented.

Also, to make this system as user friendly and community integrated as possible, I think that a new rating system should be implemented for videos, called Quality.

Of course, this rating system can be abused, but a report system can also be introduced in order to find people who abused it.

1. **Reward good videos, and punish lazy long content.**

Using the previously mentioned Quality rating system, we can identify the good videos using algorithms and promote them. We can also punish content that is lazy and long in order to get watch time on their channel.

1. **Help new creators.**

YouTube doesn’t like new creators; this is common knowledge for people who try to make content for YouTube. My idea for fixing this is having a Discover section, where new creators will be showcased based on the Quality rating and other things.

This will give users an easier time finding new content, while also giving a chance to small and new creators.

1. **Better communication tools between creators and admins.**

If you have a problem on YouTube, good luck solving it within a month. The only way to contact YouTube is either trough e-mail or unorganized tickets, and both take a lot of time to get answers.

YouTube’s solution was to assign personal contact employees to the biggest of creators, however this showcases the fact that they do not care enough about their smaller (still big) creators.

The proposed solution is to have a more complex ticket system, where the problems can be organized into different urgency levels. The abuse of this system will lead to the creator having their future tickets not as important as the others.

1. **Better communications for DMCA/Copyright claims**

Currently on YouTube, creators have no way to respond to a DMCA/Copyright claim. This means that even if the claim is false, they lose the video and/or revenue.

# **User Stories**

User

|  |  |  |
| --- | --- | --- |
| **As a(n):** | **I want to:** | **So that:** |
| User | Create an account | I can access the website’s functionality. |
| User | Customize my experience | I don’t have to see content I don’t like |
| User | Be able to watch videos | I can be entertained |
| User | Be able to pause videos | I can take a break |
| User | Be able to rate videos | I can express my opinion |
| User | Be able to change my rating on videos | I can express my change in opinion |
| User | Be able to comment | I can express my opinion |
| User | Be able to reply to comments | I can engage in discussions |
| User | Be able to report videos | I can help remove toxic content |
| User | Be able to become a content creator | I can post my videos |
| User | Be able to search by tags | I can find what interests me |

Content Creator (CC)

|  |  |  |
| --- | --- | --- |
| **As a(n):** | **I want to:** | **So that:** |
| CC | Be able to upload my videos | I can showcase my work |
| CC | Be able to view the ratings on my videos | I can know which videos do well and which don’t |
| CC | Be able to delete/hide my videos | I can remove my own videos that I do not like |
| CC | Be able to make Tickets | I can contact the admin team regarding help |

Admin

|  |  |  |
| --- | --- | --- |
| **As a(n):** | **I want to:** | **So that:** |
| Admin | Be able to view and reply to tickets | I can help users |
| Admin | Be able to remove toxic videos | I can keep a toxic-free environment |
| Admin | Be able to ban users | I can remove toxic users |

Outside Company (OC)

|  |  |  |
| --- | --- | --- |
| **As a(n):** | **I want to:** | **So that:** |
| OC | Be able to copyright videos due to copyright issues. | I can keep my Intellectual Propriety safe |

# **Use Cases**

|  |  |
| --- | --- |
| **Name** | UC01 : Login |
| **Summary** | The actor wants to Log In |
| **Actors** | User, Moderator, Admin , Content Creator |
| **Assumptions** | 1. Actor has an account |
| **Description** | 1. The actor tries to log in using his credentials 2. The system checks the given information |
| **Exceptions** | 1. The user information is incorrect 2. The account has been banned |
| **Results** | The actor is logged in, and can use the website’s functionality |

|  |  |
| --- | --- |
| **Name** | UC02 : Log out |
| **Summary** | The actor wants to Log out |
| **Actors** | User, Admin, Moderator, Content Creator |
| **Assumptions** | 1. The actor is logged in |
| **Description** | 1. The actor pressed the log out button 2. The actor is logged out of the system |
| **Exceptions** | NONE |
| **Results** | The actor is logged out. |

|  |  |
| --- | --- |
| **Name** | UC03 : Register |
| **Summary** | The actor wants to Register |
| **Actors** | User, Admin, Content Creator |
| **Assumptions** |  |
| **Description** | 1. The actor fills in the fields required for registration 2. The system checks the information and creates an account |
| **Exceptions** | 1. The actor has filled in incorrect information 2. The username/e-mail already exists |
| **Results** | The actor is registered. |

|  |  |
| --- | --- |
| **Name** | UC04 : Post Video |
| **Summary** | The actor wants to post a video |
| **Actors** | Content Creator |
| **Assumptions** | 1. The actor has a video in the correct file format 2. The video is not corrupted |
| **Description** | 1. The actor uploads the video to the upload page 2. The actor names the video and adds a description 3. The system uploads the video to the server |
| **Exceptions** | 1. The server is not online 2. The video is corrupted 3. The size of the video is too big |
| **Results** | The actor uploaded a video. |

|  |  |
| --- | --- |
| **Name** | UC05 : Delete Video |
| **Summary** | The actor wants to delete a video |
| **Actors** | Content Creator |
| **Assumptions** | 1. The actor has a video uploaded to the website 2. The actor is the owner of the video |
| **Description** | 1. The actor accesses video settings 2. The actor presses the delete button 3. The actor confirms the deletion of the video 4. The system removes access to the video to users |
| **Exceptions** | 1. The video is already removed 2. The user does not own the video |
| **Results** | The actor deleted his video. |

|  |  |
| --- | --- |
| **Name** | UC06 : Like video |
| **Summary** | The actor wants to like a video |
| **Actors** | User, Moderator, Admin, Content Creator |
| **Assumptions** | 1. The actor is logged in |
| **Description** | 1. The actor presses the like button 2. (optional) The system removes his dislike if he already disliked the video 3. The system adds his like to the video |
| **Exceptions** | 1. Actor is banned |
| **Results** | The actor likes a video. |

|  |  |
| --- | --- |
| **Name** | UC07 : Dislike video |
| **Summary** | The actor wants to dislike a video |
| **Actors** | User, Moderator, Admin, Content Creator |
| **Assumptions** | 1. The actor is logged in |
| **Description** | 1. The actor presses the dislike button 2. (optional) The system removes his like if he already disliked the video 3. The system adds his dislike to the video |
| **Exceptions** | 1. Actor is banned |
| **Results** | The actor dislikes a video. |

|  |  |
| --- | --- |
| **Name** | UC08 : Quality a video |
| **Summary** | The actor wants to quality a video |
| **Actors** | User, Moderator, Admin, Content Creator |
| **Assumptions** | 1. The actor is logged in |
| **Description** | 1. The actor selects the quality rating 2. The system adds/changes his quality rating of the video |
| **Exceptions** | 1. Actor is banned |
| **Results** | The actor “qualities” a video. |

# **Non-Functional Requirements.**

## Technology and performance

Since the service will be implemented as microservices, I will be using different languages in the back-end based on what the most efficient one is for the task. A good example is that I will be using a Python API for uploading the video to the server, since Python is one of the fastest languages for file uploading.

This will be the same for the Databases, I will use the one most suited for the task.

For the front-end, React will be used.

## **Storage**

Due to the nature of the project, storing videos takes a lot of storage data, so for this project I will be mostly using a few dummy videos with different lengths in order to showcase the algorithms.

## **Security**

Security is an important part of every application in this day and age, 2-factor authentication will be present together with a JsonWebTokens implementation for the accounts.

## **Scalability**

Due to the potential of million of users, the service has to be scalable. This will be done using Kubernetes and instancing of microservices.

In order for the service to be scalable, it also has to have a good architecture.

## **Privacy**

Privacy is a big part of every service that handles sensitive data, such as personal data.

In order to make sure that WatchTime will offer privacy, a users personal data will only be available to themselves.

In the back-end, checks will be run in order to make sure that the data will not be sent to the incorrect user.

# **Challenges**

In my opinion, this project is the perfect fit for Enterprise SD, as the main point of it, is to build an application that will be used by potentially millions of people, so the design has to accommodate for that.

1. Have good security.
2. Create a design that can accommodate potentially millions of people.
3. Implement a microservice system that can easily be scaled and replaced.
4. Make the services easily maintainable.
5. The use of many different technologies.

# **Constraints**

The biggest constraint during this semester is time. Personally, I do not think I will have enough time to implement all the features.

The following features will not be developed if time does not allow it:

* Video Editing
* DMCA/Copyright handling
* Discover Algorithms
* Ticketing system

# **Initial Ideas**

In no particular order:

* Use React for front-end (Easy to set up and uses JavaScript)
* Use Python for video Uploading
* Split User and Creator APIs so in case of high website traffic creators can still upload without fail.
* Use JWT for Users
* Implement E-mail 2FA
* Algorithms based on watch time, quality rating and like/dislike ratio