# Software Requirements and Design Document

**for**

# Amaranth

**Redefining Streams, Preserving Dreams**

**Prepared by**

**Hassan Ismail i22-1011**

**Rehal Saeed i22-1071**

**Samra Mashaam i22-0757**

# Table of Contents

# 1. Introduction

## 1.1 Purpose

This document presents the design and analysis of the development of the software Amaranth. It includes the aims, objectives, use cases, and design diagrams required to understand the construction of the software.

## 1.2 Product Scope

This product aims to be a better alternative to current video sharing platforms by improving the content moderation and sponsorship structures commonly found across these platforms. It is created using Java, with MySQL for the database and JavaFX for the UI.

## 1.3 Title

Amaranth: Redefining Streams, Preserving Dreams. A video sharing and live streaming service that provides a safe and well-regulated experience for all.

## 1.4 Objectives

Amaranth is a platform where registered and approved users can upload and live stream videos. Any content uploaded to the platform will be archived indefinitely for media preservation, so creators - particularly livestreamers - don't have to worry about archival and reuploads of their content. Video streamers and uploaders can earn through the site as well. All content on the site will be moderated and classified by age group, meaning kids and adults alike can enjoy content appropriate for their age ranges.

## 1.5 Problem Statement

Amaranth hopes to address the issue of content moderation that is faced by multiple similar services on the internet, which affects both content creators and viewers. "Children's content" is poorly moderated, and children are exposed to inappropriate content and advertisements. Content for general audiences, on the other hand, is over moderated and heavily censored to appease advertisers, who want to make use of midroll ads. This means that content creators often find their content demonetised, restricted, or removed entirely from the platform, affecting their income and viewership. This also ruins the experience for viewers and silences discussion on important topics.

Amaranth intends to fix this through proper moderation and removal of midroll ads. A dedicated 'For Kids' section, where all videos are strictly searched through before being uploaded, will ensure that children of all ages can browse in safety, and the removal of midroll ads will ensure that children are not exposed to unsavory content from there either. On the other hand, general content will only be moderated to a reasonable level, and censorship to appease advertisers will not be an issue.

Due to the removal of midroll ads, Amaranth will also empower content creators by providing them with alternate revenue streams including brand deals, subscriptions, and donations through their viewers. With the help of these technologies, creators will be able to make money directly from the platform without relying on sponsors, which will give them more autonomy over the financial success of their work. By implementing these programs, Amaranth hopes to guarantee everyone's safety and happiness while fostering a more balanced and creator-friendly environment.

# 2. Overall Description

## 2.1 Product Perspective

Amaranth is a brand-new product that is not contained within any pre-existing product. It is inspired by popular video sharing platforms such as YouTube and Twitch, but aims to improve upon their current model.

## 2.2 Product Functions

The general functions are as follows:

- Allow content creators to livestream and upload videos
- Allow viewers to view videos, and donate and subscribe to creators
- Allow moderators to regulate content safely and avoid censorship

## 2.3 List of Use Cases

- Search content
- View content
- Engage with content
- Register as viewer
- Register as content creator
- Donate to content creator
- Manage account settings and history
- Access exclusive content
- Renew/cancel subscription
- Flag content
- Change content rating
- Handle report
- Suspend/ban users
- Ban content
- Upload video
- Stream live
- Handle payment
- Manage video library
- View stats

## 2.4 Extended Use Cases

**Use case name:** Search Content

**Scope:** Video streaming
**Level:** User goal
**Primary actor:** Viewer
**Stakeholders:**

- Viewer: Wants the video they searched to be brought up
- Content creator: Wants the viewer to watch their videos

**Precondition:** Uploaded videos exist on the platform
**Postcondition:** Videos that match the viewer's input are displayed to them
**Main Success scenario:**

| Viewer: | System: |
|---|---|
| Viewer navigates to search bar | |
| Viewer inputs their desired content | |
| | System takes input and selects all video titles that match |
| | System displays videos to user |

**Extensions:**

3a. System finds no matching video titles
3a1. System informs user that no matching titles were found

**Use case name:** View content
**Scope:** Video streaming
**Level:** User goal
**Primary Actor:** Viewer
**Stakeholders:**

- Viewer: Wants to watch a video uninterrupted.
- Content creator: Wants their video to be watched by the viewer.
- Streaming platform: Wants the viewer to keep using their services and track content views.

**Preconditions:**

- Videos exist on the platform.
- The viewer has selected a video to watch.

**Postconditions:**
Selected video is being played for the viewer and view count has been incremented accordingly.
**Main success scenario:**

| Viewer: | System: |
|---|---|
| Viewer selects a video from the list of available videos. | |

| | |
|---|---|
| | The system loads and plays the video. |
| | The system tracks the video view and increments the view count. |
| Viewer watches the content. | |

**Extensions:**

2a. Video fails to load.
2a1. The system informs the viewer that the video failed to load and suggests reloading or trying again later.

3a. Viewer pauses the video.
3a1. The system pauses the video and allows the viewer to resume it later.

3c. Internet connection is lost.
3c1. The system pauses the video and shows a message informing the viewer of connection issues.

**Use case name:** Engage with content
**Scope:** Video streaming
**Level:** User goal
**Primary actor:** Viewer
**Stakeholders:**

- Viewer: Wants to interact with content
- Streaming platform: Wants to record the interaction with the content

**Preconditions:**

- User is logged in/ registered as a viewer
- User has viewed the video

**Postconditions:** User has completed the interaction, and the system has recorded it.
**Main Success Scenario:**

| Viewer: | System: |
|---|---|
| Viewer selects an interaction type<br>a. If "Like" option, the likes display is increased<br>b. If "comment" option, see section "Comments"<br>c. If "share" option, see section "Shares"<br>d. If "report" option, see section "Reports" | |

| | |
|---|---|
| | System records the interaction information |

**Section:** Comments

| | |
|---|---|
| User inputs their comment | |
| | System takes comment and displays it in the comments section |
| | The comments display is increased |

**Alternative course:** User decides not to comment. Interaction is canceled.

**Section:** Shares

| | |
|---|---|
| System provides video link to user | |
| | User copies it into their clipboard |

**Section:** Report

| | |
|---|---|
| | System displays list of rules and regulations that uploaded videos should not break |
| | System presents user with options to explain their reason for reporting |
| User inputs their choice(s) | |
| | System sends report information to moderation team |

**Alternative course:** User decides not to report. Interaction is canceled.

**Use case name:** Register as viewer
**Scope:** Video streaming
**Level:** User goal
**Primary actor:** Viewer
**Stakeholders:**

- Viewer: Wants to be registered with platform, so that they can access engagement and donation features
- Platform: Wants users information registered and recorded with them

**Preconditions:** User is on the website
**Postconditions:** The viewer has an account on the platform, where their account information is stored.
**Main success scenario:**

| Viewer: | System: |
|---|---|
| | System requests user information. |
| User inputs information, including name, password, email, age, and a unique user ID, among other things. | |
| | Based on age, the system sorts the account into "Kids" or "Adults" account. |
| System records the information and registers it as a new account. | |

**Extensions:**

2a. Email already exists as a registered account, see section "Log in"

2b. Another account has the same user ID.
2b1. User is requested to enter a new ID

2c. Email is invalid
2c1. User is prompted to re enter email address

**Section:** Log in

| System prompts user to enter old password | |
|---|---|
| | User inputs password |
| System logs the user in | |

**Alternative Courses:**

2a. User enters wrong password
2a1. User made to reenter password

**Use case name:** Register as content creator
**Scope:** Video streaming
**Level:** User goal
**Primary actor:** Viewer

**Stakeholders:**

- Viewer: Wants to be registered as content creator to upload content
- Platform: Wants information regarding the content creators to monitor them

**Preconditions:** User is logged in as a viewer with an adults account
**Postconditions:** Viewer account is given access to content creator features and is recorded as such
**Main success scenario:**

| Viewer: | System: |
|---|---|
| | System prompts user to input information such as bank details, among other things |
| User inputs information | |
| | System records user as content creator |

**Extensions:**

2a. Bank information is invalid
2a1. User is informed and requested to re enter information

**Use case name:** Donate to content creators
**Scope:** Video streaming
**Level:** User goal
**Primary actor:** Viewer
**Stakeholders:**

- Viewer: Wants to support content creators by donating.
- Content Creator: Wants to receive donations from viewers.
- Streaming Platform: Wants to facilitate the donation process to earn commission.
- Payment Portal (External Actor): Processes the viewer's donation securely.

**Preconditions:**

- Viewer is logged in/registered.
- Payment portal is integrated with the platform.

**Postconditions:**

- The donation is successfully processed, and the content creator and platform receive the funds.
- Viewer is informed of the successful transaction.
- System logs the donation transaction.

**Main success scenario:**

| Viewer: | System/ portal: |
|---|---|

| | |
|---|---|
| Viewer navigates to a content creator's page and selects "Donate" option | |
| Viewer enters the donation amount and relevant information. | |
| | System sends payment details to the external Payment Portal. |
| | Payment Portal processes the transaction. |
| | System receives confirmation from the Payment Portal. |
| | System informs the viewer and content creator that the donation was successful. |
| | System records the donation details. |

**Extensions:**

4a. Viewer cancels the donation process.
4a1. Viewer exits the donation form. No transaction is made, and no further action is required.

6a. Payment is declined or fails.
6a1. Payment Portal informs the system of the failure.
6a2. System informs the viewer of the failure and suggests retrying or using a different payment method.

6b. Viewer's connection is lost during payment.
6b1. System notifies the viewer to check their payment status after reconnecting and suggests retrying if necessary.

**Use case name:** Manage account settings and history
**Scope:** User goal
**Primary actor:** Viewer
**Stakeholders:**

- Viewer: Wants to manage personal information, preferences, and view history.
- Streaming Platform: Wants to provide users with an interface for managing account settings and to store user data.

**Preconditions:**

- Viewer is logged in/registered.

- Account settings and history data exist in the system.

**Postconditions:**

- Account settings and preferences are updated as per the viewer's input.
- The system updates and displays the settings and viewing history based on viewer requests.

**Main success scenario:**

| Viewer: | System: |
|---|---|
| Viewer navigates to the "Account Settings" section of the platform. | |
| Viewer selects the option to update account information. | |
| Viewer makes desired changes. | |
| | System validates the inputs and saves the updates. |
| Viewer navigates to the "View History" section and views it. | |
| Viewer selects the option to clear all or part of the view history. | |
| | System confirms the action and clears the specified viewing history. |
| Viewer receives confirmation that the settings and history changes were successfully saved. | |

**Extensions:**

3a. Viewer inputs incorrect information.
3a1. System informs the user of incorrect data and prompts them to reenter it.

4a. Viewer cancels account changes.
4a1. Viewer exits the page and no updates are made.

5a. Viewer's viewing history is empty.
5a1. System informs the viewer that there is no history available to display.

7a. Viewer decides not to clear history.
7a1. Viewer cancels the action, and the viewing history remains unchanged.

**Use case name:** Access exclusive content
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Subscriber
**Stakeholders:**

- Subscriber: Wants to see exclusive content that they paid for
- Content Creator: Wants their paid content to be viewed

**Preconditions:** User is logged in and has paid for a subscription
**Postconditions:** User account is allowed to access and engage with exclusive content
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. User navigates to content creator's page | |
| 2. User selects 'For Subscribers' | |
| | 3. System filters and shows content for subscribers only |
| 4. User can now view and interact with any exclusive content present | |

**Extensions**

2a. No exclusive content created by creator
2a1. System will not show any results

**Use case name:** Renew/cancel subscription
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Subscriber
**Stakeholders:**

- Subscriber: Wants to manage their subscription status
- Content Creator: Wants to be subscribed to

**Preconditions:** User is logged in, and has a subscription
**Postconditions:** User account has canceled or continued subscription
**Main success scenario:**

| **User** | **System** |
|---|---|

| | |
|---|---|
| 1. User navigates to content creator's page | |
| 2. User selects subscription options | |
| 3. User can select:<br>a. 'Unsubscribe': user is no longer subscribed<br>b. 'Renew subscription': user can extend subscription | |

**Section:** Unsubscribe

| **User** | **System** |
|---|---|
| | 1. System warns that user will not be refunded for any remaining months on the plan |
| | 2. System gives user the options to continue or go back |
| 3. User presses continue | |
| | 4. System ends subscription |

**Alternative Courses:**

2a. User decides to not continue

2a1. System brings them back to subscription options

**Section:** Renew

| **User** | **System** |
|---|---|
| 1. User inputs month range of subscription | |
| | 2. System generates total cost and presents payment options |
| 3. User selects payment option | |

| | |
|---|---|
| | 4. System processes payment |
| | 5. System informs user of a successful payment and subscription |

**Alternative Courses:**

2a. User decides to not continue
2a1. System brings them back to subscription options
3a. User's payment fails
3a1. System rolls back transaction and informs user of cause of failure

**Use case name:** Flag content
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Moderator
**Stakeholders:**

- Platform: Wants content on their platform to be moderated
- Viewer: Wants to engage with content suited to them

**Preconditions:** Moderator has viewed the content at least once
**Postconditions:** System flags the content sends a warning to the creator with the reason
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Moderator navigates to watch history | |
| 2. Moderator selects video | |
| 3. Moderator chooses to flag video | |
| 4. Moderator fills out form with reasoning for flag | |
| | 5. System sets flag for video and adds it to Flagged list |
| | 6. System records details and sends them to the video's creator |

**Extensions**

2a. Video already flagged

2a1. Moderator can change flag status after viewing the video

**Use case name:** Change content rating
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Moderator
**Stakeholders:**

- Platform: Wants content on their platform to be sorted by age rating
- Viewer: Wants to engage with content suited to their age

**Preconditions:** Moderator has viewed the content at least once
**Postconditions:** System sets the age rating to the one chosen by moderator and informs the content creator of the change
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Moderator navigates to watch history | |
| 2. Moderator selects video | |
| 3. Moderator selects ‘change age rating’ | |
| 4. Moderator selects new rating | |
| 5. Moderator fills form with reason for change | |
| | 6. System changes age rating |
| | 7. System records details and sends them to the video’s creator |

**Use case name:** Handle Report
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Moderator
**Stakeholders:**

- Platform: Wants user to participate in content moderation
- Viewer: Wants their report to be reviewed

**Preconditions:** Content has been reported by at least 5 different users

**Postconditions:** Moderator can see and handle reports accordingly
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Moderator navigates to Reports list | |
| 2. Moderator selects a report | |
| | 3. System displays report details and reported content |
| 4. Moderator can:<br>a. Flag\ban reported content<br>b. Change age rating<br>c. Reject report as false | |
| | 5. System resolves report and removes it from list |

**Section:** Flag/ban content

| **User** | **System** |
|---|---|
| 1. Moderator selects video | |
| 2. Moderator chooses to flag video | |
| 3. Moderator fills out form with reasoning | |
| | 4. System sets flag and add video to Flagged list |
| | 5. System records details and sends them to the video's creator |

**Alternative Courses:**

2a. Video is already flagged
2a1. Moderator can ban video

**Section:** Change age rating

| **User** | **System** |
|---|---|
| 1. Moderator selects video | |
| 2. Moderator changes age rating | |
| 3. Moderator fills out form with reasoning | |
| | 4. System records details and sends them to the video's creator |

**Section:** Reject Report

| **User** | **System** |
|---|---|
| 1. Moderator selects video | |
| 2. Moderator chooses to reject report | |
| 3. Moderator fills out form with reasoning for rejection | |
| | 4. System records details and sends them to the reporters |
| | 5. System does not change anything about the video |

**Extensions**

2a. No video reported
2a1. Report section will be empty

**Use case name:** Suspend/Ban users
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Moderator
**Stakeholders:**

- Platform: Wants content creators to abide by Terms and Conditions

- Viewer: Wants to engage with content that are safe to consume

**Preconditions:** Content creator has at least three videos flagged and/or banned
**Postconditions:** System temporarily or permanently restricts the account from accessing the platform and informs creator of this
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Moderator navigates to creator's account | |
| 2. Moderator selects:<br>a. Suspension<br>b. Ban | |
| 3. Moderator fills out form with reasoning | |
| | 4. System restricts the account |
| | 5. System records details and sends them to the creator |

**Section:** Suspend

| **User** | **System** |
|---|---|
| 1. Moderator selects suspension duration | |
| 2. Moderator fills out form with reasoning | |
| | 3. System records details and sends them to the creator |
| | 4. System restricts the account |

**Alternative Courses:**

2a. Account does not have 3 or more flagged videos
2a1. Moderator cannot suspend account

**Section:** Ban

| **User** | **System** |
|---|---|
| 1. Moderator selects Ban | |
| 2. Moderator fills out form with reasoning | |
| | 3. System records details and sends them to the creator |
| | 4. System deletes the account |

**Alternative Courses:**

2a. Account does not have 3 or more banned videos
2a1. Moderator cannot ban account

**Use case name:** Ban content
**Scope:** Video Streaming
**Level:** User goal
**Primary actor:** Moderator
**Stakeholders:**

- Platform: Wants content to abide by Terms and Conditions
- Viewer: Wants to engage with content that are safe to consume

**Preconditions:** Content violates major rules of terms and conditions and has been flagged
**Postconditions:** System permanently removes the video from the platform and informs creator of this
**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Moderator navigates to Flagged list | |
| 2. Moderator selects video | |
| 3. Moderator chooses to ban it | |
| 4. Moderator fills out form with reasoning | |
| | 5. System deletes the video |

| | |
|---|---|
| | 6. System records details and sends them to the creator |

**Use case name:** Upload video (set age ratings and adverts)

**Scope:** Video streaming and content management

**Level:** User goal

**Primary actor:** Content Creator

**Stakeholders and interests:**

Content Creator: Wants to upload videos to reach an audience and monetize content through ads.

Platform: Needs to ensure content complies with platform guidelines and age restrictions.

Viewers: Expect appropriate content based on age rating.

Advertisers: Want ads placed on relevant content.

**Preconditions:** Content creator is logged in.Content creator has a linked bank account for monetization.

**Postconditions**: Video is uploaded, available to viewers, and subject to age restrictions and advertisements as applicable.

**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Content creator logs into the platform and navigates to the "Upload" section. | |
| | 2. System authenticates the user and loads the "Upload" section. |
| 3. Creator selects a video for upload. | |
| | 4. System processes the selected video file for upload. |
| 5. Creator sets the appropriate age rating for the video. | |

| | |
|---|---|
| | 6. System records and applies the selected age rating. |
| 7. Creator enables ads for the video, if applicable. | |
| | 8. System enables ad functionality based on the creator's settings. |
| 9. Creator submits the video for upload. | |
| | 10. System uploads the video and makes it available with the specified age rating and ad settings. |

**Extensions:**

1a. Upload fails due to network issues: System informs the creator and suggests retrying.

3a. If the age rating is not set or is inappropriate, the system prompts the creator to select a valid rating before proceeding.

**Use case name:** Stream live

**Scope:** Live streaming functionality

**Level:** User goal

**Primary actor:** Content Creator

**Stakeholders and interests:**

Content Creator: Wants to engage with the audience in real time and monetize live content.

Viewers: Expect to watch and interact with live content.

Platform: Ensures smooth live streaming and compliance with platform rules.

Advertisers: May want ads to run during live streams.

**Preconditions:** Creator has a valid account and access to live streaming features.Platform supports live streaming for the creator’s account.

**Postconditions:** The live stream is broadcast to viewers, and data (viewers, engagement, etc.) is tracked.

**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Creator configures live stream settings (title, description, category, etc.). | |
| | 2. System saves the configured settings and prepares for the live stream. |
| 3. Creator starts the live broadcast. | |
| | 4. System initiates the live stream and begins broadcasting. |
| 5. Creator monitors the stream. | |
| | 6. System tracks live stream performance metrics (viewers, duration, comments, etc.). |
| 7. Stream ends. | |
| | 8. System saves the live stream data for future analysis and performance reports. |

**Extensions:**

2a. Network failure during streaming: System informs the creator and tries to reconnect.

4a. Live content is flagged for violation: Moderation team is alerted, and the stream may be suspended.

**Use case name:** Handle Payment

**Scope:** Monetization through donations, ads, and subscriptions

**Level:** User goal

**Primary actor:** Content Creator

**Stakeholders and interests:**

Content Creator: Wants to receive payments for content through donations, ads, and subscriptions.

Viewers: Want to support content creators they enjoy.

Platform: Manages payment processing and ensures proper payouts.

Advertisers: Ensure ads are shown and monetized properly.

**Preconditions:** Creator is eligible for monetization (meets platform criteria).Viewers have valid payment methods set up for donations and subscriptions.

**Postconditions:** The creator receives payments based on viewer engagement, ad views, and subscriptions.

**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Viewers engage with content through donations, ad views, or subscriptions. | |
| | 2. System processes the transactions (donations, ad revenue, or subscriptions). |
| 3. Creator monitors engagement and revenue. | |
| | 4. System credits payments to the creator's linked bank account. |

**Extensions:**

2a. Payment fails due to insufficient funds: System notifies the viewer and retries the transaction.

2b. Invalid bank details or flagged payment for fraud: Creator is notified to update details.

**Use case name:** Manage video library

**Scope:** Content management

**Level:** User goal

**Primary actor:** Content Creator

**Stakeholders and interests:**

Content Creator: Wants to manage their content by adding, removing, or updating videos.

Viewers: Expect content to be accessible and organized.

Platform: Ensures the creator's content is properly managed and available to viewers.

**Preconditions:** Creator has uploaded videos to their library.

**Postconditions:** The creator's video library is updated, with content added, removed, or modified as needed.

**Main success scenario:**

| **User** | **System** |
|---|---|
| 1. Creator navigates to the video library management section. | |
| | 2. System displays the video library management interface. |
| 3. Creator selects videos to edit, delete, or add to the library. | |
| | 4. System processes the selected actions (edit, delete, add). |
| 5. Creator confirms the changes. | |
| | 6. System applies changes to the video library and reflects them in the creator's profile. |

**Extensions:**

2a. Video is linked to advertisements and cannot be deleted: System informs the creator, and they must request deletion through platform support.

2b. The moderation team blocks the video and orders the creator to take it down for compliance.

**Use case name:** View stats

**Scope:** Analytics and insights on content performance

**Level:** User goal

**Primary actor:** Content Creator

**Stakeholders and interests:**

Content Creator: Wants to review video performance metrics (views, likes, engagement).

Platform: Provides creators with insights and analytics for video performance.

**Preconditions**: The content has user engagement (e.g., views, likes, comments).

**Postconditions:** The creator views performance metrics and uses them for future decisions.

**Main success scenario:**

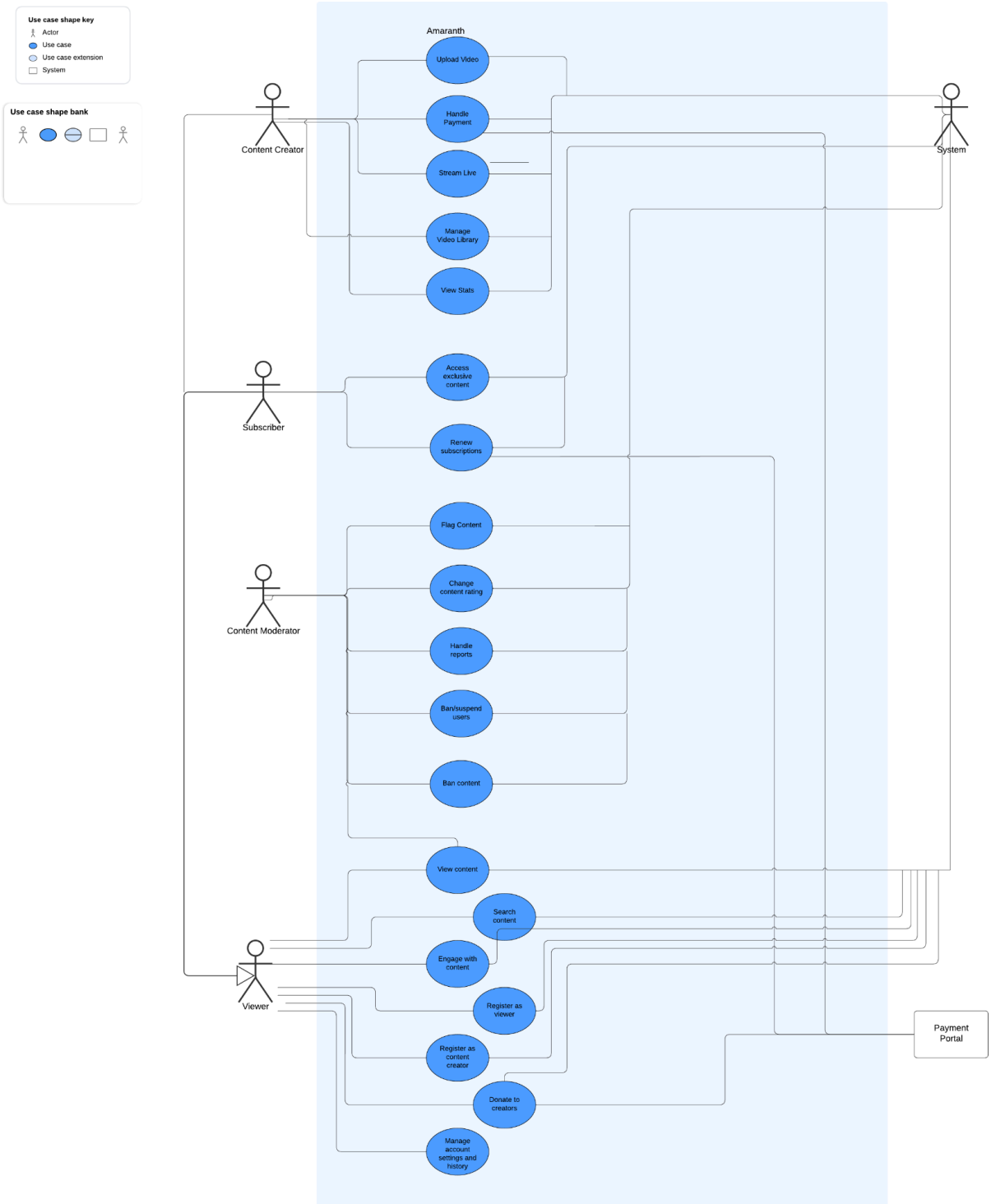
| **User** | **System** |
|---|---|
| 1. Creator navigates to the Analytics section. | |
| | 2. System displays detailed statistics on video performance (views, likes, comments, etc.). |
| 3. Creator reviews the performance data. | |
| | 4. System updates the statistics in real-time as new data comes in. |

**Extensions:**

2a. System fails to update stats due to server issues: System notifies the creator and attempts to resolve the issue.

2b. If the creator wants to filter stats for a specific date range or video, the system allows them to apply the necessary filters for precise insights.

## 2.5 Use Case Diagram



# 3. Other Nonfunctional Requirements

## 3.1 Performance Requirements

To ensure the platform runs smoothly during all kinds of circumstances, here are the following performance requirements:
The platform should be scalable and should be able to handle an average of 1,000,000 users during peak times without compromising performance. Uploaded videos must be processed and available for viewing within five minutes for shorter, standard-definition videos and up to twenty minutes for high-definition and longer videos. Videos must start playing within five seconds of a user pressing play. Real-time interactions like chat and donation notifications must update within five seconds, especially during livestreams. In case of server failing, the system must recover and restore services within thirty seconds, obscuring the failure from the users. Lastly, database queries and data retrieval must be executed within ten seconds, and should not slow down under multiple requests.

## 3.2 Safety Requirements

To ensure a safe and secure experience for all, the platform has multiple safety features in place. It ensures secure encryption for all transactions and communication between users. Content flagged as dangerous or inappropriate is reviewed and action is taken within twenty four hours. All content on the platform follows local regulations.

## 3.3 Security Requirements

The platform ensures that there are robust security features to protect users. All user information, such as payment details, are encrypted and authentication methods are used. The system will be protected against cyber threats, including DDoS attacks, SQL injection, and unauthorized access. Additionally, user-generated content will be monitored to prevent the sharing of malicious content.

## 3.4 Business Rules

Only verified and approved content creators can upload videos, ensuring safe and moderated tracking of all creators. Moderators are authorized to review flagged content and take actions such as suspension or banning. Age-restricted content must only be accessible to verified users within the appropriate age. Payment-related features, such as donations and subscriptions, require verified payment methods.

## 3.5 Operating Environment

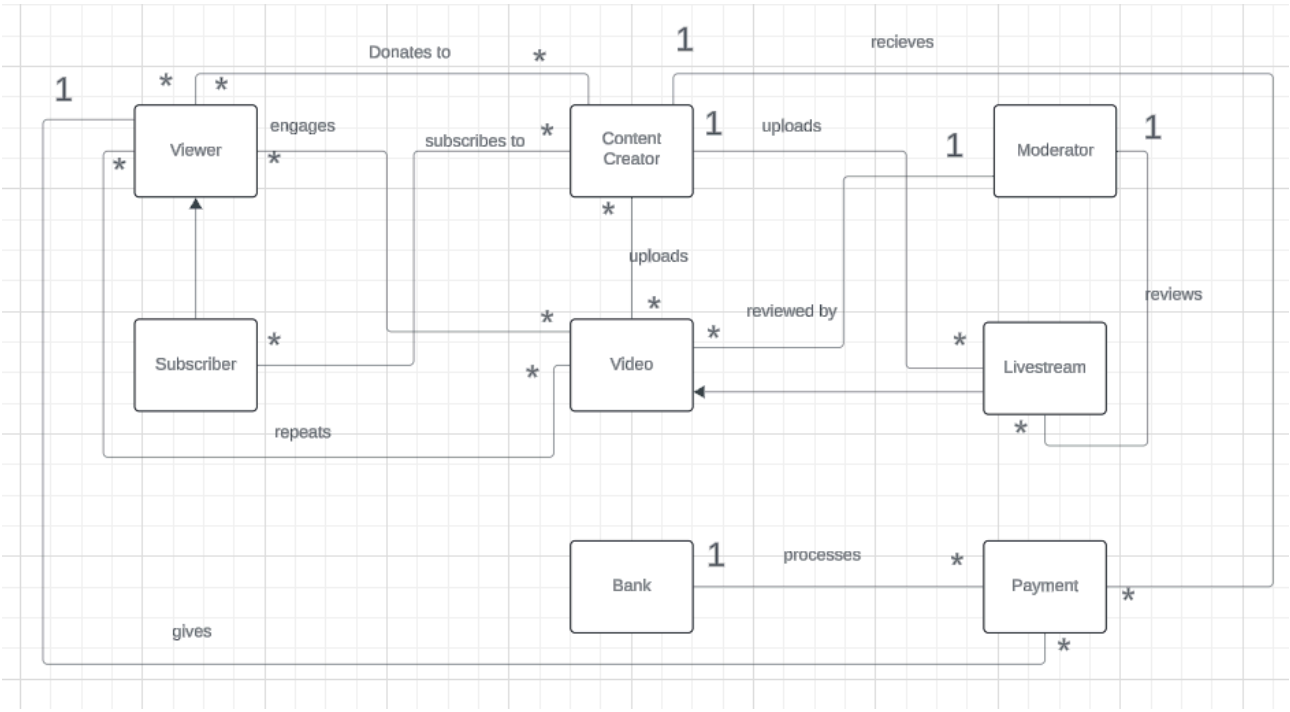
Amaranth is meant to operate on multiple platforms. It will support Java Runtime Environment (JRE), with MySQL as the database backend. The application is built using JavaFX for the UI,

ensuring compatibility with modern graphical environments. It must integrate seamlessly with web browsers and coexist with antivirus software without performance interference.

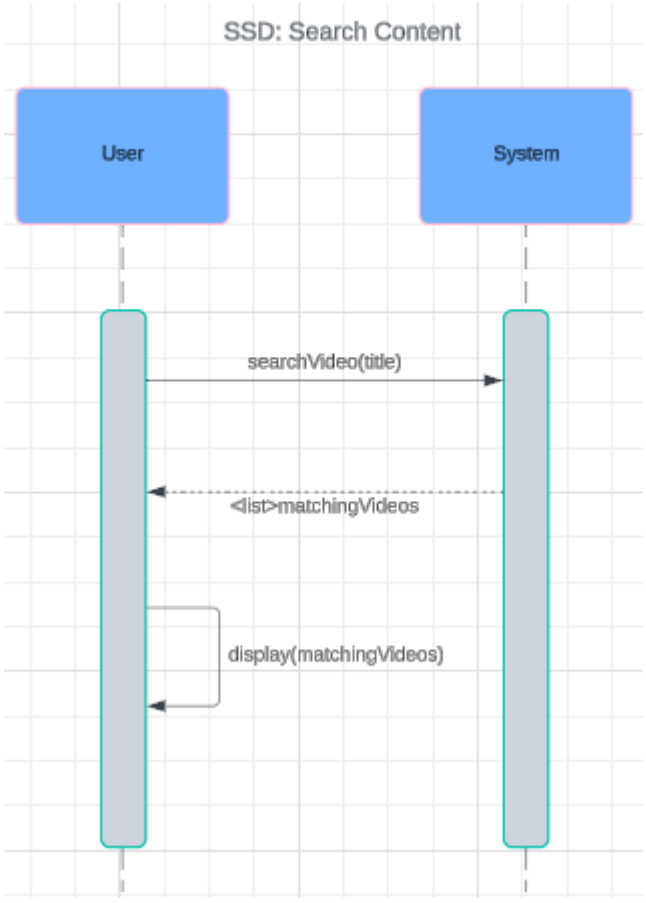
### 3.6 User Interfaces

There will be interfaces for users, content creators, and moderators. Subscribers will have access to a few interfaces that regular users won't. There will be general similarities in the interfaces, such as search bar, home button, and pause, play, rewind buttons on videos.

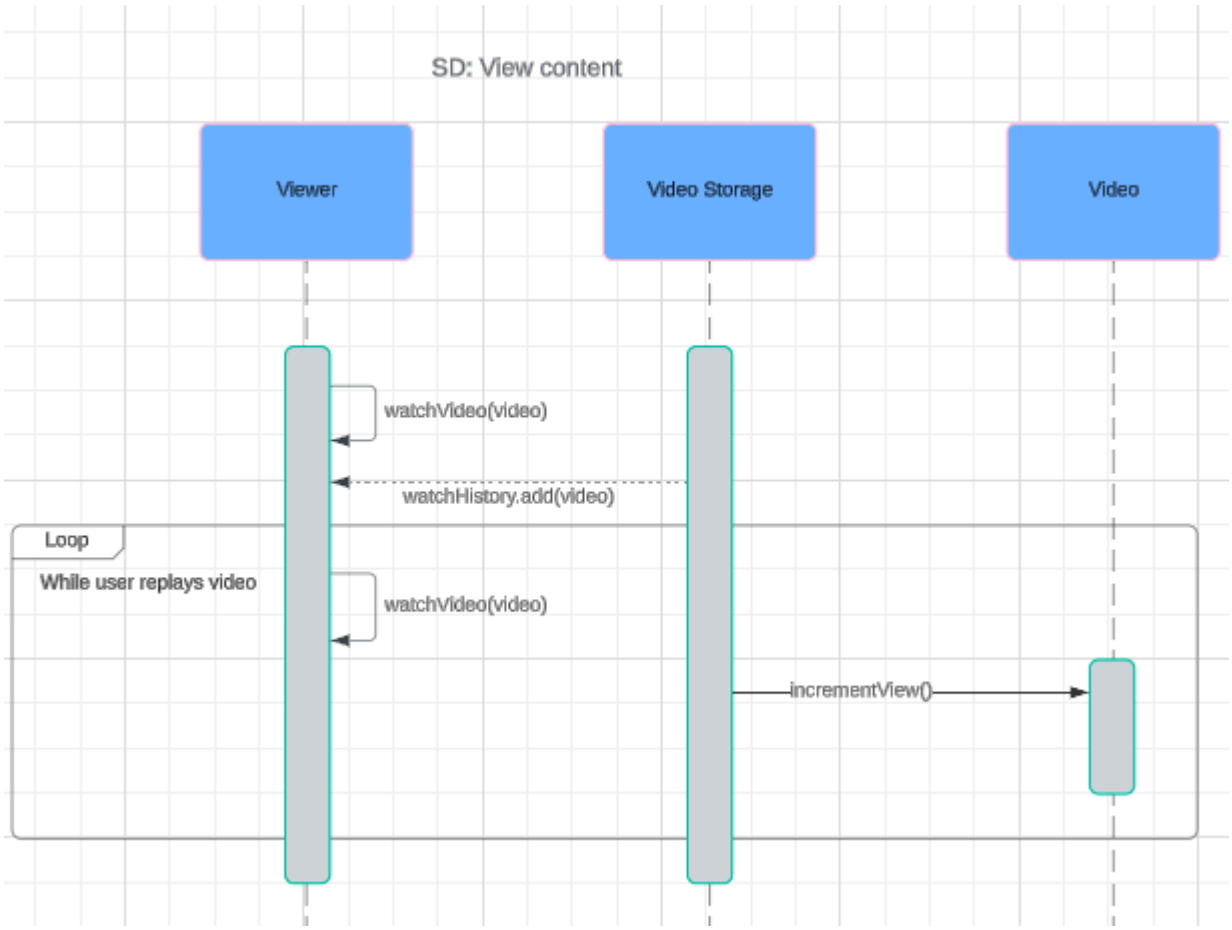
# 4. Domain Model



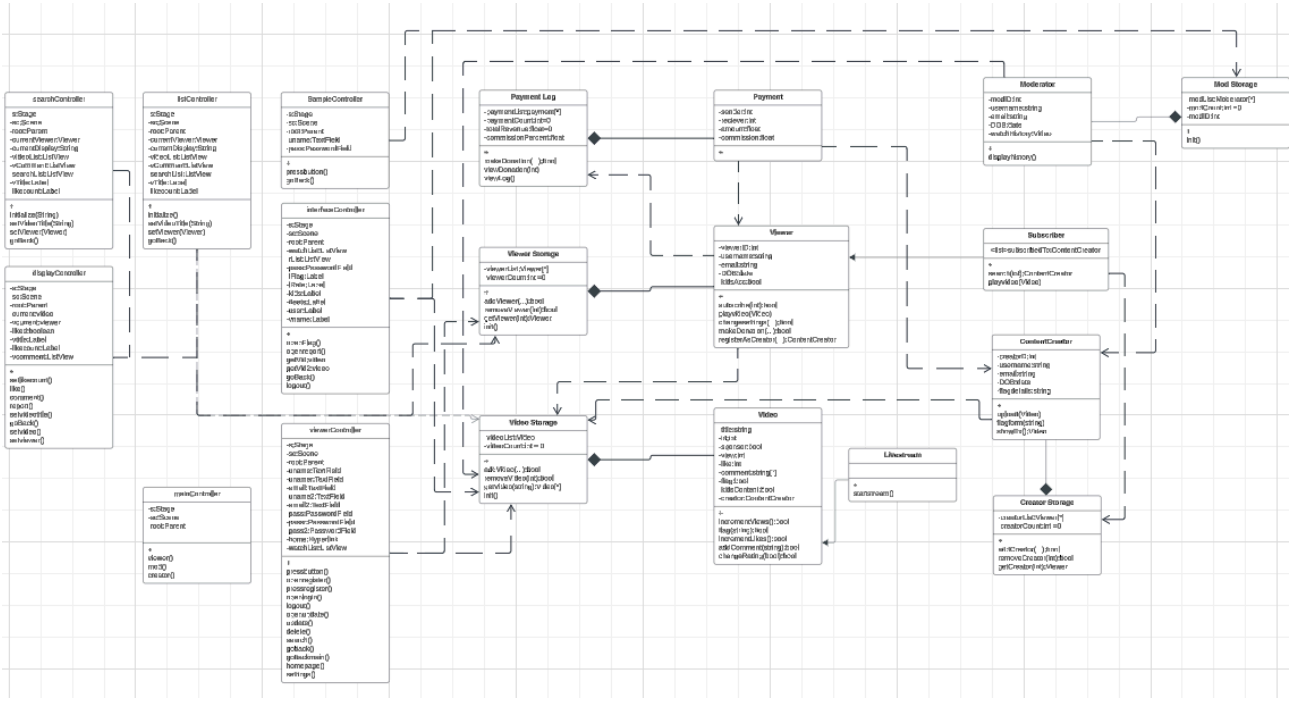
# 5. System Sequence Diagram



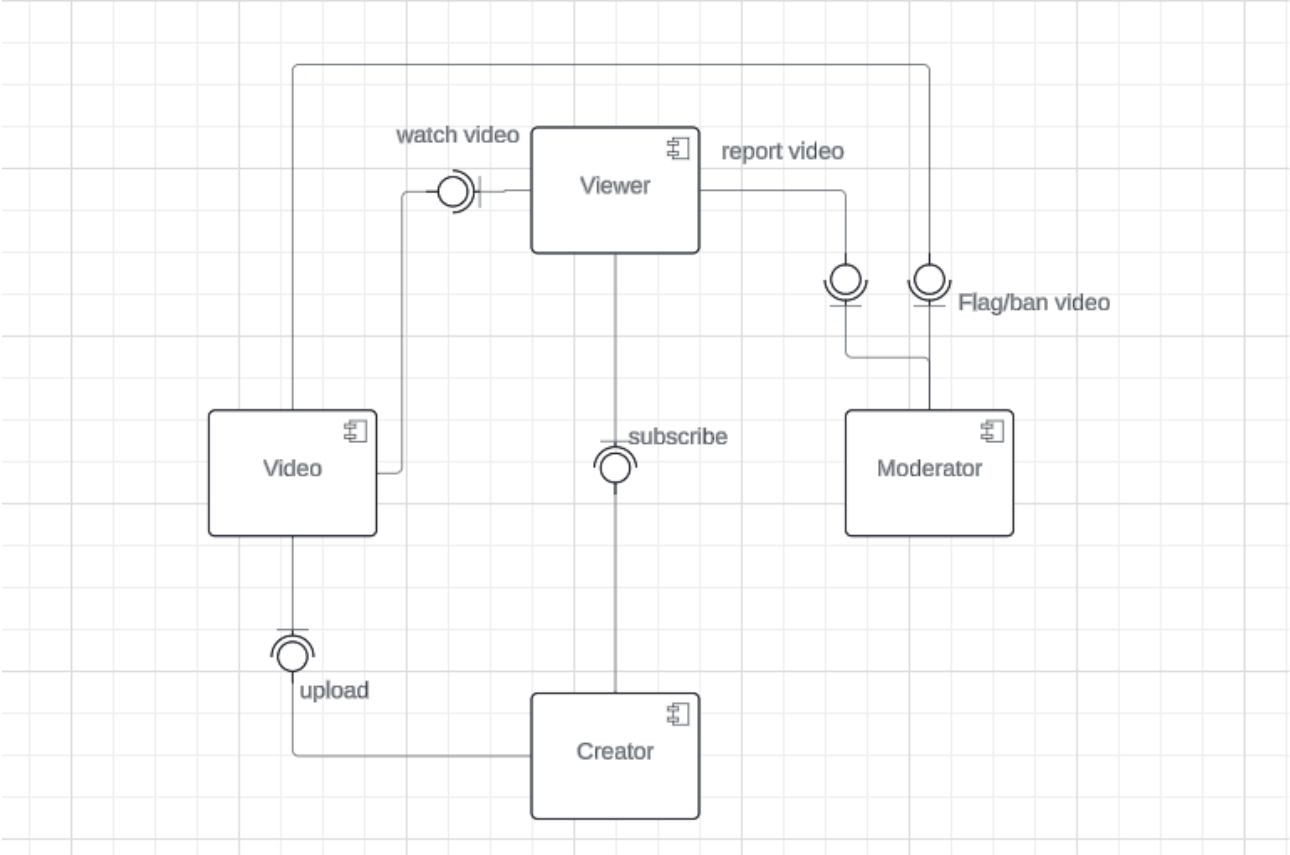
# 6. Sequence Diagram



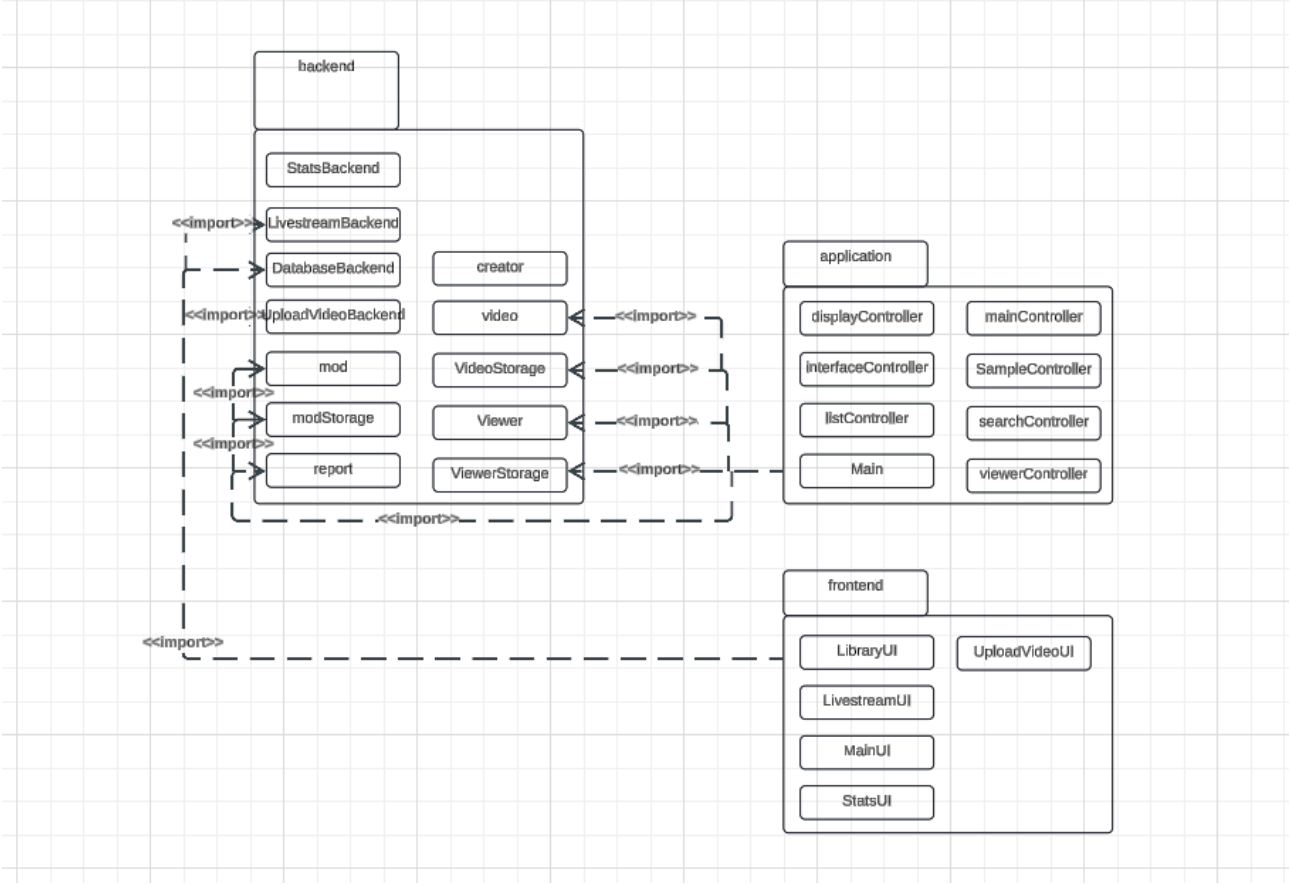
# 7. Class Diagram



# 8. Component Diagram

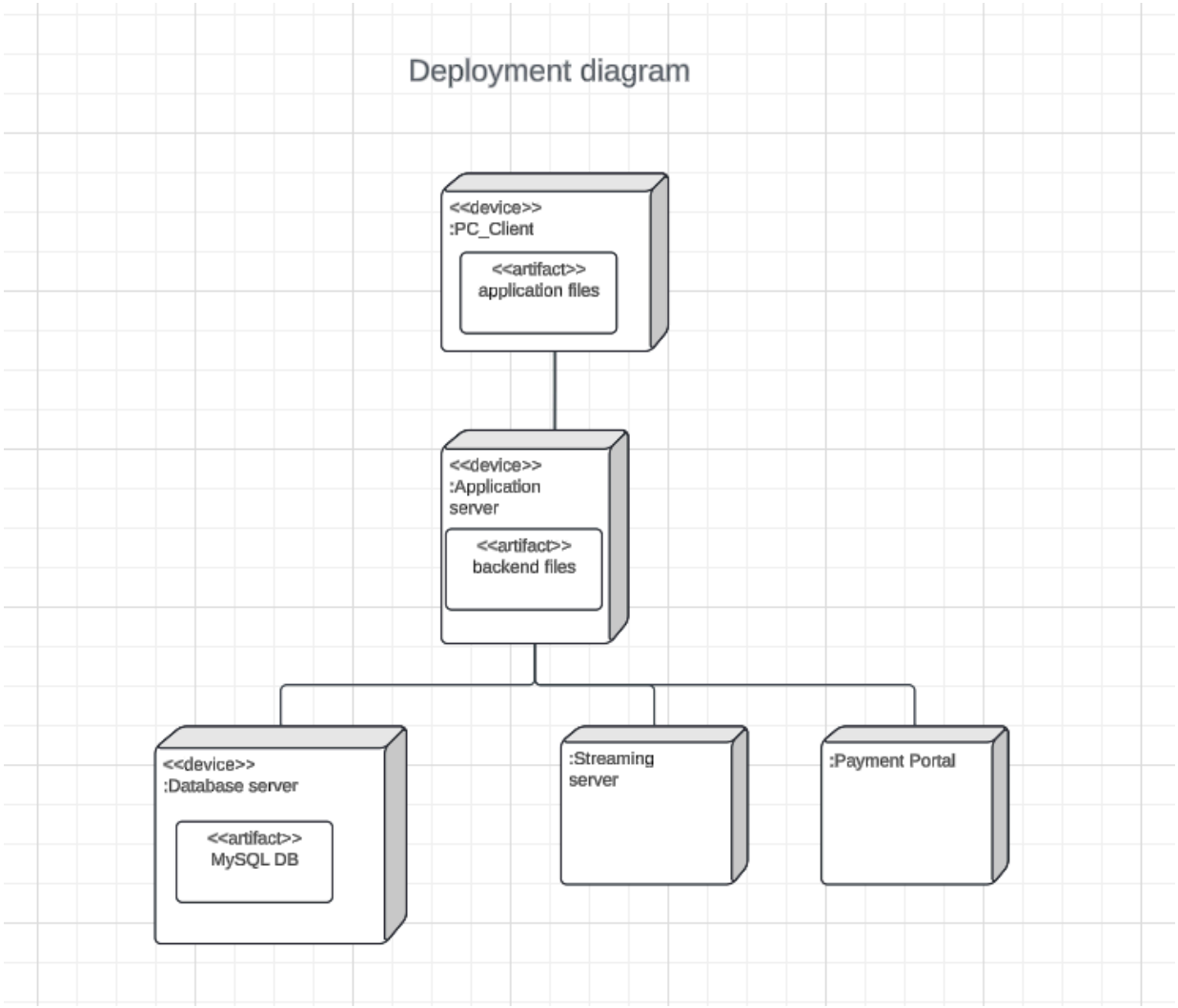


# 9. Package Diagram




# 10. Deployment Diagram

Below is what the deployment diagram may look like during actual implementation of the application on a commercial level:




Below is the deployment diagram of the current prototype of the application: