

Use cases of “Streaming Platform”

1)

RQ01	The system allows user registration
Description	This use case allows a new user to create an account in the system.
Pre-condition	<ul style="list-style-type: none">– User is not registered– The provided input data is valid (correct email format, unique username, etc.)
Post-condition	<ul style="list-style-type: none">– User account is created– User data is saved in the system– The user receives confirmation of successful registration
Dependencies	<ul style="list-style-type: none">– None
Basic path	<ol style="list-style-type: none">1. The user opens the registration page.2. The user enters the required information.3. System validates input data.4. The system verifies the uniqueness of the username and email.5. The system hashes the password and creates a user account.6. The system stores the user data in the database.7. The system sends a confirmation of successful registration (e.g., via email).8. The user receives confirmation and can log into the system.
Alternative path	<ol style="list-style-type: none">2a. If email or name already exists, system returns an error.
Exceptional path	<ol style="list-style-type: none">3a. If data validation fails, system prompts user to correct input.6a. If a system error occurs while saving the user data (e.g., database is unavailable), the system returns an error.

2)

RQ02	The system allows user authentication
Description	This use case allows a registered user to log in to the system.
Pre-condition	<ul style="list-style-type: none">– User is registered– The user provides valid login credentials (email/phone number and password)
Post-condition	<ul style="list-style-type: none">– The user is successfully logged in– The system generates and returns an authentication token– The user gains access to authorized functionalities
Dependencies	<ul style="list-style-type: none">– RQ01
Basic path	<ol style="list-style-type: none">1. The user opens the login page.2. The user enters login credentials.3. System verifies credentials against repository.4. The system generates an authentication token.5. The system returns the authentication token to the user.6. The user is successfully logged in and can access authorized features.
Alternative path	2a. If incorrect credentials, system returns an error.
Exceptional path	3a. If authentication service fails, system logs the error and denies login.

3)

RQ03	The system allows watching a film.
Description	This use case allows any user to watch a film through the system.
Pre-condition	<ul style="list-style-type: none">– The film is available in the system– The user is authorized (if necessary for access)

Post-condition	– The film is played back in the appropriate quality
Dependencies	– RQ08 – RQ02 (if required)
Basic path	<ol style="list-style-type: none"> 1. The user navigates to the film catalog or searches for a film. 2. The user selects a film to watch. 3. The system checks the availability of the film. <ol style="list-style-type: none"> a. Retrieves film metadata (title, duration, quality options, etc.). 4. The system determines the film source. <ol style="list-style-type: none"> a. If the film is stored in cache, the system retrieves it from the cache and starts playback. b. If the film is not cached, the system retrieves it from the repository and loads it. 5. The film starts playing for the user.
Alternative path	2a. If film data is not in cache, system fetches from repository.
Exceptional path	3a. If film is not found, system returns an error.

4)

RQ04	The system allows editing the user profile
Description	This use case allows the user to change their profile details, except for mail.
Pre-condition	– User is logged in
Post-condition	– Profile data has been updated (except for mail)
Dependencies	– RQ02
Basic path	<ol style="list-style-type: none"> 1. The user logs into their profile. 2. The user selects an option to edit the profile. 3. The system displays the current user data.

	<p>4. The user changes their data (e.g. first name, last name, password, photo, etc.).</p> <p>5. The user confirms the changes.</p> <p>6. The system saves the changes and updates the user profile.</p>
Alternative path	2a. If the user cannot log in to the profile (e.g., not authorized), the system displays an error and requests authorization.
Exceptional path	4a. If a user attempts to change mail, the system displays an error informing that mail cannot be changed through this process.

5)

RQ05	The system allows commenting on a film
Description	This use case allows logged-in users to comment on a film.
Pre-condition	<ul style="list-style-type: none"> – The user is logged in – The film exists in the system
Post-condition	<ul style="list-style-type: none"> – The comment is successfully posted and visible to other users
Dependencies	<ul style="list-style-type: none"> – RQ02 – RQ03
Basic path	<p>1. The user navigates to a film's page.</p> <p>2. The user enters a comment in the input field.</p> <p>3. The user submits the comment.</p> <p>4. The system validates the comment (e.g., checks for empty content or prohibited words).</p> <p>5. The system saves the comment to the repository.</p> <p>6. The system displays the comment under the film, making it visible to other users.</p> <p>7. The system confirms the action.</p>

Alternative path	4a. If comment contains restricted content, system requests correction.
Exceptional path	4b. If comment system fails, error is returned.

6)

RQ06	The system allows viewing watch history
Description	This use case allows logged-in users to view their watch history.
Pre-condition	<ul style="list-style-type: none"> – The user is logged in – The system has recorded watch history for the user
Post-condition	<ul style="list-style-type: none"> – The user can see a list of films they have watched
Dependencies	<ul style="list-style-type: none"> – RQ02 – RQ03
Basic path	<ol style="list-style-type: none"> 1. The user logs into their account. 2. The user navigates to the "Watch History" section. 3. The system retrieves the user's watch history from the database. 4. The system displays a list of previously watched films, including relevant details. 5. The user views the watch history.
Alternative path	2a. The system detects that no history data is available for the user.
Exceptional path	3a. System fails to retrieve watch history.

7)

RQ07	The system allows managing users
Description	This use case allows an administrator to manage user accounts by changing roles, blocking or unblocking accounts, and viewing user details.

Pre-condition	<ul style="list-style-type: none"> – The administrator must be logged into the system – The administrator must have either the RELEASER or ADMIN role
Post-condition	<ul style="list-style-type: none"> – The user's role or account status is updated in the system
Dependencies	<ul style="list-style-type: none"> – RQ02
Basic path	<ol style="list-style-type: none"> 1. Admin selects the Manage Users section. 2. The administrator searches for a specific user by username, email, or other identifier. 3. The system retrieves and displays the user's details. 4. The administrator updates the user's role, blocks, or unblocks the account. 5. The system validates and saves the changes. 6. The user's updated status is applied in the system.
Alternative path	<ol style="list-style-type: none"> 4a. If the user is already blocked, the system notifies the admin.
Exceptional path	<ol style="list-style-type: none"> 3a. If the user is not found, the system returns an error.

8)

RQ08	The system allows uploading a film
Description	Allows an administrator to upload new films to the platform.
Pre-condition	<ul style="list-style-type: none"> – The administrator must be logged into the system – The administrator must have either the RELEASER or ADMIN role
Post-condition	<ul style="list-style-type: none"> – The film is uploaded and available for viewing
Dependencies	<ul style="list-style-type: none"> – RQ02
Basic path	<ol style="list-style-type: none"> 1. Admin navigates to the Upload Film section. 2. Admin provides film details (title, description, genre, duration, quality). 3. Admin uploads the film file. 4. The system validates the file format and metadata.

	5. The system stores the film and updates the database.
Alternative path	4a. If the film format is incorrect, the system requests correction.
Exceptional path	5a. If an upload error occurs, the system notifies the admin.

9)

RQ09	The system allows editing film information
Description	Allows an administrator to edit film details (title, description, genre, etc.).
Pre-condition	<ul style="list-style-type: none"> – The administrator must be logged into the system – The administrator must have either the RELEASER or ADMIN role
Post-condition	<ul style="list-style-type: none"> – Film information is updated in the system
Dependencies	<ul style="list-style-type: none"> – RQ02 – RQ08
Basic path	<ol style="list-style-type: none"> 1. The admin navigates to the Manage Films section. 2. The admin selects a film to edit. 3. The system displays current film details. 4. The admin updates the necessary fields. 5. The system saves changes.
Alternative path	4a. If no changes are made, the system notifies the admin.
Exceptional path	3a. If the film is not found, the system returns an error.

10)

RQ10	The system allows deleting comments
-------------	--

Description	This use case allows a moderator to remove comments that violate platform rules, particularly those that have been reported.
Pre-condition	<ul style="list-style-type: none"> – The user is logged into the system – The user has a MODERATOR or ADMIN role – The comment has been reported – The comment still exists in the system
Post-condition	<ul style="list-style-type: none"> – The comment is permanently removed from the system
Dependencies	<ul style="list-style-type: none"> – RQ02 – RQ05 – RQ12
Basic path	<ol style="list-style-type: none"> 1. The moderator is notified of a report on a comment. 2. The moderator reviews the comment and any associated reports. 3. The moderator decides whether to delete the comment or leave it. 4. If the moderator decides to delete the comment, the system removes it from the platform.
Alternative path	<ol style="list-style-type: none"> 2a. If the comment is already deleted, the system notifies the moderator. 4a. If the moderator decides that the comment does not violate the rules, the moderator will leave the comment.
Exceptional path	<ol style="list-style-type: none"> 4b. If a comment cannot be deleted, the system displays an error.

11)

RQ11	The system allows banning users
Description	This use case allows a moderator to ban users who violate platform policies, preventing them from accessing the platform.

Pre-condition	<ul style="list-style-type: none"> – The moderator is logged into the system – The moderator has a MODERATOR or ADMIN role
Post-condition	<ul style="list-style-type: none"> – The user is banned, and their access to the platform is revoked
Dependencies	<ul style="list-style-type: none"> – RQ02
Basic path	<ol style="list-style-type: none"> 1. The moderator navigates to the User Moderation section. 2. The moderator searches for a specific user by username, email, or other identifier. 3. The system retrieves and displays the user's information. 4. The moderator selects the option to ban the user. 5. The system updates the user's status to banned, restricting their access.
Alternative path	4a. If the user is already banned, the system notifies the moderator.
Exceptional path	3a. If the user is not found, the system returns an error.

12)

RQ12	The system allows reporting a comment
Description	This use case allows users to report inappropriate comments.
Pre-condition	<ul style="list-style-type: none"> – The user is logged into the system
Post-condition	<ul style="list-style-type: none"> – The report is sent to moderators
Dependencies	<ul style="list-style-type: none"> – RQ02 – RQ05
Basic path	<ol style="list-style-type: none"> 1. The user selects the comment he wants to report. 2. The user clicks on the “Complaint” button under the comment. 3. The system asks the user for the reason for the complaint (e.g., insults, spam, etc.). 4. The user selects or enters the reason for the report.

	5. The system saves the report and notifies the moderator of the new report.
Alternative path	<p>2a. If the comment is not found, the system displays an error that the comment does not exist.</p> <p>3a. If the user cannot submit a report (e.g., not authorized), the system displays an error.</p>
Exceptional path	<p>2b. If the user does not have permission to submit a report, the system displays an error.</p> <p>5a. If the comment is already reported, the system does not duplicate the report.</p>