Use Case ID	***UC001***
Use Case Name	Register
Summary	This use case describes the process of registering a new user in the system
Actors	Guest
Pre-Conditions	- The system is operational. - The guest has not been registered in the system.
Post-conditions	- The guest is registered in the system with a unique identifier. - The system confirms successful registration.
Main Flow	1. The guest accesses the registration page. 2. The system presents the registration form. 3. The guest enters the required information (e.g., username, password, email). 4. The system validates the entered information. 5. If the information is valid, the system creates a new user account. 6. The system generates a unique identifier for the new user. 7. The system stores the user information and identifier in the database. 8. The system confirms successful registration to the guest and sending an activation mail.
Alternate Flow(s)	Invalid Information: - If the entered information is invalid, the system notifies the guest and prompts them to correct the errors. The flow returns to step 3. Existing User: - If the entered username or email is already associated with an existing user, the system notifies the guest and prompts them to choose a different username or email. The flow returns to step 3.
Nonfunctional requirement	Performance - The registration process should be completed within 30 seconds under normal system load. Security - User passwords must be securely hashed and stored in the database. Usability - The registration form should be intuitive and accessible, following web accessibility standards.

Use Case ID	***UC002***
Use Case Name	Log in
Summary	This use case outlines the process by which a registered user logs into the system.
Actors	Registered User
Pre-conditions	The user is registered in the system. The system is operational.
Post-conditions	The user is successfully logged into the system.
Main Flow	1. The registered user navigates to the login page. 2. The system presents the login form. 3. The user enters their valid credentials (e.g., username and password).

	4. The system verifies the entered credentials against the stored user information. 5. If the credentials are valid, the system authenticates the user. 6. The system grants access to the user's authorized functionalities. 7. The user is now logged into the system.
Alternate Flow(s)	Invalid Credentials: - If the entered credentials are invalid, the system notifies the user and prompts them to re-enter the correct information. The flow returns to step 3. Account Lockout: - After a certain number of unsuccessful login attempts, the system may temporarily lock the user's account for security purposes. The system informs the user about the account lockout and provides instructions for unlocking.
Non-functional requirement	Security - User passwords must be securely hashed and stored in the database. The system should implement measures to prevent unauthorized access, such as account lockout after multiple failed login attempts. Usability - The login process should be user-friendly and provide clear error messages for incorrect login attempts. Performance - The login process should be responsive, with a maximum response time of 10 seconds under normal system load.

Use Case ID	*UC003*
Use Case Name	Recover password
Summary	This use case describes the process by which a registered user recovers their password.
Actors	Registered User
Pre-conditions	The system is operational. The registered user has forgotten their password or wishes to reset it.
Post-conditions	The user successfully recovers their password. The system updates the user's password in the database.
Main Flow	1. The user clicks on the "Forgot Password" or a similar link. 2. The system presents a password retrieval/reset form. 3. The user enters their registered email address or username. 4. The system verifies the provided information against the database. 5. If the information is valid, the system generates a unique token or link and sends it to the user via email. 6. The user clicks on the token or link to verify their identity. 7. The system presents a form for the user to enter a new password. 8. The user enters a new password. 9. The system validates and updates the password in the database. 10. The system confirms successful password retrieval/reset to the user.
Alternate Flow(s)	Invalid Information:

	- If the entered email address or username is not found in the system, the system notifies the user and prompts them to enter valid information. The flow returns to step 3. Token/Link Expiry: - If the token or link has expired, the system prompts the user to initiate the password retrieval process again.
Non-functional requirement	Security - The password retrieval/reset process should include secure mechanisms, such as one-time tokens or expiring links, to prevent unauthorized access. Usability - The password retrieval/reset process should be user-friendly and provide clear instructions at each step. Performance - The password retrieval/reset process should be responsive, with a maximum response time of 10 seconds under normal system load.

Use Case ID	***UC004***
Use Case Name	Manage Profile
Summary	This use case involves the actions a registered user can take to manage and update their profile information within the system.
Actors	Registered User
Pre-conditions	The user is logged into the system. The system is operational.
Post-conditions	The user's profile information is successfully updated in the system.
Main Flow	1. The user navigates to the "Profile" or "Account Settings" section. 2. The system displays the user's current profile information. 3. The user selects the option to edit their profile. 4. The system presents an editable form with the user's current information pre-filled. 5. The user modifies the desired fields (e.g., name, contact information, profile picture). 6. The system validates the updated information. 7. If the information is valid, the system updates the user's profile in the database. 8. The system confirms successful profile update to the user.
Alternate Flow(s)	Invalid Information: - If the entered information is invalid, the system notifies the user and prompts them to correct the errors. The flow returns to step 4. Cancel Edit: - If the user decides not to save the changes, they can cancel the edit, and the system retains the previous profile information.
Non-functional requirement	Usability - The profile management interface should be intuitive, providing a seamless experience for users to edit and update their information. Performance: The profile update process should be responsive, with a maximum response time of 10 seconds under normal system load.

	Security: User authentication is required before allowing access to the profile management functionality, ensuring that only authorized users can update their profiles.

Use Case ID	***UC005***
Use Case Name	Log in by Google
Summary	This use case outlines the process by which a user can log into the system using their Google credentials.
Actors	Registerd user, Google Authentication Service
Pre-conditions	The system is operational. The user has a Google account.
Post-conditions	The user is successfully authenticated and logged into the system using their Google credentials.
Main Flow	1. The user selects the "Login with Google" option on the login page. 2. The system redirects the user to the Google authentication service. 3. The user enters their Google credentials (if not already logged in). 4. The Google authentication service verifies the user's credentials. 5. Upon successful authentication, the Google authentication service generates an authentication token. 6. The Google authentication service redirects the user back to the system's callback URL, providing the authentication token. 7. The system receives the authentication token and verifies it with the Google authentication service. 8. If the token is valid, the system identifies the user associated with the Google account or creates a new user account if it's the first login. 9. The system grants access to the user's authorized functionalities. 10. The user is now logged into the system using their Google credentials
Alternate Flow(s)	Failed Authentication: - If the Google authentication service fails to authenticate the user, the system notifies the user and may provide guidance on resolving the issue.
Non-functional requirement	Security: The communication between the system and the Google authentication service should be secure and use industry-standard protocols to protect user credentials. Usability: The "Login with Google" process should be seamless and user-friendly, providing clear instructions for users. Performance: The authentication process should be responsive, with a maximum response time of 10 seconds under normal system load.

Use Case ID	***UC006***
Use Case Name	Log In with Account System
Summary	This use case describes the process by which a user logs into the system using their locally registered account credentials.
Actors	Registered user
Pre-conditions	The system is operational. The user is registered in the system.
Post-conditions	The user is successfully authenticated and logged into the system.

Main Flow	1. The user navigates to the login page on the system. 2. The system presents the login form, requesting the user's credentials (e.g., username and password). 3. The user enters their valid credentials. 4. The system verifies the entered credentials against the stored user information. 5. If the credentials are valid, the system authenticates the user. 6. The system grants access to the user's authorized functionalities. 7. The user is now logged into the system
Alternate Flow(s)	Failed Authentication: - If the Google authentication service fails to authenticate the user, the system notifies the user and may provide guidance on resolving the issue.
Non-functional requirement	Security: The communication between the system and the Google authentication service should be secure and use industry-standard protocols (e.g., OAuth 2.0) to protect user credentials. Usability: The "Login with Google" process should be seamless and user-friendly, providing clear instructions for users. Performance: The authentication process should be responsive, with a maximum response time of 10 seconds under normal system load.

Use Case ID	***UC007***
Use Case Name	Update profile
Summary	This use case outlines the process by which a registered user updates their profile information within the system
Actors	Registered user
Pre-conditions	The system is operational. The user is logged into the system.
Post-conditions	The user's profile information is successfully updated in the system.
Main Flow	1. The user navigates to the "Profile" or "Account Settings" section. 2. The system displays the user's current profile information. 3. The user selects the option to edit their profile. 4. The system presents an editable form with the user's current information pre-filled. 5. The user modifies the desired fields (e.g., name, contact information, profile picture). 6. The system validates the updated information. 7. If the information is valid, the system updates the user's profile in the database. 8. The system confirms successful profile update to the user.
Alternate Flow(s)	Invalid Information: - If the entered information is invalid (password or username don't follow a general rule of the system), the system notifies the user and prompts them to correct the errors. The flow returns to step Cancel Edit: - If the user decides not to save the changes, they can cancel the edit, and the system retains the previous profile information.

Non-functional requirement	Usability: The profile management interface should be intuitive, providing a seamless experience for users to edit and update their information. Security: User authentication is required before allowing access to the profile management functionality, ensuring that only authorized users can update their profiles.

Use Case ID	***UC008***
Use Case Name	Get profile
Summary	This use case outlines the process of displaying a user's profile information within website.
Actors	Registered user
Pre-conditions	The user must be logged into their account. The website must be running and accessible.
Post-conditions	The user's profile information is successfully retrieved and displayed to the user. The system remains in the current state.
Main Flow	1. The user navigates to the profile section within the application. 2. The system authenticates the user's session. 3. The system retrieves the user's profile information from the database. 4. The system displays the user's profile information on the application interface, including details such as name, email, profile picture, and any other relevant information.
Alternate Flow(s)	Incomplete Profile: - If the user's profile information is incomplete or missing essential details, the system may prompt the user to complete their profile or provide the missing information. The user is redirected to the profile editing section. Profile Not Found: - If there is an issue retrieving the user's profile information (e.g., technical error or database connectivity issues), the system displays an error message and prompts the user to try again later.
Non- functional Requirement	Performance - The profile retrieval process should be quick and responsive, with the user's profile information displayed within a few seconds of the request. Security - User profile information should only be accessible to authorized users with the appropriate permissions, and the communication between the application and the database should be secure. Scalability - The profile retrieval process should scale efficiently as the number of users or the amount of profile data increases, ensuring consistent performance even under high user loads.

Use Case ID	***UC009***
Use Case Name	Watch Free Movie
Summary	This use case describes the process by which a user can watch a free movie on an online streaming platform.
Actors	Registered User
Pre-conditions	The system is operational. The user logged into system
Post-conditions	The user successfully watches the selected free movie.
Main Flow	1. The user accesses the Movie page of the website. 2. The system presents the user with a catalog of free movies. 3. The user selects a specific free movie. 4. The system retrieves and plays the selected free movie. 5. The user enjoys the movie.
Alternate Flow(s)	The server is unable to load the movie. - Encouraging the user to either reload the page or return later.
Non-functional requirement	Usability - The process of selecting and watching a free movie should be user-friendly and intuitive. Availability - The platform should ensure high availability, allowing users to watch free movies without significant downtime. Scalability - The system should be scalable to accommodate an increasing number of users watching free movies simultaneously.

Use Case ID	***UC010***
Use Case Name	View Movie Infomation
Summary	This use case outlines the process by which a user can access detailed information about a specific movie within the website
Actors	Guest or Registered User
Pre-conditions	The system is operational. The user is logged in.
Post-conditions	The user has viewed detailed information about the selected movie.
Main Flow	1. The user accesses the movie page of the website. 2. The system presents the user with a catalog of movies. 3. The user selects a specific movie from the catalog. 4. The system displays detailed information about the selected movie, including but not limited to: - Title - Release Year - Genre - Synopsis - Cast and Crew - Ratings and Reviews - Trailers or Clips 5. The user explores the displayed movie information.

	6. The user may choose to go back to the movie catalog or proceed to watch the movie.
Alternate Flow(s)	The server is unable to load the movie. - Encouraging the user to either reload the page or return later.
Non-functional requirement	Usability - The movie information interface should be user-friendly, providing an engaging and informative experience. Availability - The movie information service should be available 24/7 to ensure users can access details whenever they want.

Use Case ID	*UC011*
Use Case Name	Rate and review
Summary	This use case describes the process by which a user can provide a rating and write a review for a movie on the online streaming platform.
Actors	Registered User
Pre-conditions	The user is logged into the system. The user has watched the movie they want to rate and review
Post-conditions	The user's rating and review are recorded in the system. The movie's overall rating may be updated based on the user's input.
Main Flow	1. The user accesses the website. 2. The user navigates to the movie they want to rate and review. 3. The system verifies whether the user has viewed the movie. 4. The system presents options for the user to provide a rating (e.g., star rating). 5. The user selects a rating for the movie. 6. The system allows the user to write a review for the movie. 7. The user enters their review text and submits it. 8. The system records the user's rating and review in the database associated with the specific movie.
Alternate Flow(s)	User has not watched the movie. - Prompt them to view the movie before providing a rating. Edit Review: - If the user wishes to edit their review, the system allows them to do so, updating the information in the database. Delete Review: - If the user wishes to delete their review, the system removes the review from the database.
Non-functional requirement	Usability: The rating and review interfaces should be intuitive and easy to use. Data Integrity: The system should ensure that ratings and reviews are accurately recorded and associated with the correct movie. Performance: The process of submitting a rating and review should be responsive, with a maximum response time of 5 seconds under normal system load.

Use Case ID	*UC012*
Use Case Name	Search Movie

Summary	This use case outlines the process by which a user can search for a specific movie within the online streaming platform.
Actors	Registered User
Pre-conditions	The user is logged in to the system.
Post-conditions	The user has successfully found and accessed information about the searched movie.
Main Flow	1. The user accesses the website. 2. The system presents the user with a search bar on the top of the Website. 3. The user enters the title, keyword, actor, director, or any other relevant information related to the desired movie. 4. The system processes the search query and retrieves relevant results. 5. The system displays a list of movies matching the search criteria. 6. The user selects the specific movie they were searching for
Alternate Flow(s)	No Results: - If the search query does not yield any results, the system informs the user and may suggest refining the search terms.
Non-functional requirement	Usability: The rating and review interfaces should be intuitive and easy to use. Data Integrity: The system should ensure that ratings and reviews are accurately recorded and associated with the correct movie. Performance: The process of submitting a rating and review should be responsive, with a maximum response time of 5 seconds under normal system load.

Use Case ID	*UC013*
Use Case Name	Report Movie
Summary	This use case outlines the process by which a user can report issues or content concerns related to a specific movie on the online streaming platform.
Actors	Registered User
Pre-conditions	The user is logged into the system.
Post-conditions	The user's report is recorded and may trigger further actions by the system administrators.
Main Flow	1. The user accesses the website. 2. The user navigates to the movie they want to report. 3. The system checks whether the user has viewed the movie. 4. The system presents an option for the user to report an issue or concern. 5. The user selects the report option. 6. The system prompts the user to provide details about the issue or concern, such as:

	• Inappropriate content • Technical issues • Violations of guidelines • Other concerns 7. The user enters the necessary information and submits the report. 8. The system records the user's report in the database associated with the specific movie.
Alternate Flow(s)	User has not viewed the movie - The system prompts the user to watch the movie before generating a repor
Non-functional requirement	Usability: The reporting process should be straightforward, providing clear instructions to the user. Data Integrity: The system should ensure that reports are accurately recorded and associated with the correct movie.

Use Case ID	*UC014*
Use Case Name	Active premium account
Summary	This use case outlines the process by which a user upgrades their account to a premium subscription level on the online streaming platform.
Actors	Registered User
Pre-conditions	The user is logged into the system.
Post-conditions	The user's account is upgraded to a premium subscription level. The user gains access to premium content and features. The system records the user's premium subscription status.
Main Flow	1. The user logs into the website. 2. The user navigates to the account settings or premium subscription page. 3. The system presents information about the available premium subscription plans. 4. The user selects the desired premium subscription plan (e.g., monthly, yearly). 5. The system calculates the subscription cost and presents the payment options. 6. The user chooses a payment method. 7. The system redirects the user to the Payment Gateway for transaction processing. 8. The Payment Gateway processes the subscription payment. 9. Upon successful payment, the system updates the user's account to reflect the premium subscription status. 10. The system confirms the successful activation of the premium account to the user.
Alternate Flow(s)	Payment Failure: - If the Payment Gateway reports a failed transaction, the system notifies the user and provides guidance on resolving the issue. Cancel Subscription: - If the user decides not to proceed with the subscription, they can cancel the process at any step so the cancel button must be always displayed.

Non-functional requirement	Security - The payment process should be secure, and user authentication is required to access premium subscription features. Performance - The activation process should be responsive, with a maximum response time of 10 seconds under normal system load. Usability - The premium subscription activation process should be user-friendly, providing clear instructions and options.

Use Case ID	*UC015*
Use Case Name	Add film to Favorite list
Summary	This use case describes the process of a user adding a film to their favorite list within a movie streaming application
Actors	Paid viewer
Pre-conditions	The user must be logged into their account. The user must have a premium account on the website.
Post-conditions	The selected film is successfully added to the user's favorite list. The user interface reflects the updated favorite list.
Main Flow	1. The use log into the website. 2. The user navigates to the film details page. 3. The user identifies and selects the option to add the film to their favorite list. 4. The system validates the user's session and authorization. 5. The system adds the selected film to the user's favorite list. 6. The system updates the user interface to reflect the changes.
Alternate Flow(s)	Invalid User Authorization: - If the user does not have a premium account, the system will prompt the user to activate premium. - After successful activation, the user is redirected to the film details page to proceed with adding the film to their favorite list. Film Already in Favorites: - If the user attempts to add a film that is already in their favorite list, the system provides a notification indicating that the film is already in the list. - The user can choose to proceed or cancel the operation.
Non-functional requirement	Usability - The user interface for adding a film to the favorite list should be intuitive and easily accessible, promoting a positive user experience.

Use Case ID	*UC016*
Use Case Name	Extend Premium
Summary	This use case outlines the steps involved when a user extends their premium.

Actors	Paid viewer
Pre-conditions	The user's current premium subscription is active.
Post-conditions	The user's premium subscription is successfully extended. The system updates the user's account to reflect the new subscription period. Any applicable charges are processed, and a confirmation is provided to the user.
Main Flow	1. The user navigates to the premium subscription section within the application. 2. The user selects the option to extend their premium subscription. 3. The system presents the user with available subscription plans and pricing details. 4. The user chooses a subscription plan and provides necessary payment details. 5. The system validates the payment information. 6. If the payment is successful, the system updates the user's account with the extended premium subscription. 7. A confirmation message is displayed to the user, acknowledging the successful extension.
Alternate Flow(s)	Payment Failure: - If the payment information is invalid or the transaction fails, the system informs the user of the issue and prompts them to update their payment details. - The user is redirected to the payment information input screen. Cancellation During Process: - If the user decides to cancel the premium extension process at any point, the system provides a confirmation prompt. - If the user confirms cancellation, they are returned to the regular application interface without extending their premium subscription.
Non-functional requirement	Security - All payment transactions must adhere to industry-standard security protocols to safeguard user financial information. Availability - The premium extension functionality should be available 24/7 to accommodate users in different time zones and preferences.

Use Case ID	*UC017*
Use Case Name	Make Payment
Summary	This use case details the process a user undergoes to make a payment within the website
Actors	Registered user
Pre-conditions	The user must be logged into their account.
Post-conditions	The payment is successfully processed. The system updates relevant records to reflect the completed transaction. The user receives confirmation of the successful payment.

Main Flow	1. The user navigates to the checkout or payment section within the website. 2. The system presents the user with a summary of their order or selected service and the total amount due. 3. The user selects a payment method (bank account, system method). 4. The user provides the necessary payment details (credit card number, expiration date, CVV, etc.). 5. The system validates the payment information. 6. If the payment is successful, the system deducts the amount from the user's account and updates relevant transaction records. 7. The system generates and displays a payment confirmation message to the user.
Alternate Flow(s)	Payment Failure: - If the payment information is invalid or the transaction fails, the system informs the user of the issue and prompts them to update their payment details. - The user is redirected to the payment information input screen. Cancellation During Process: - If the user decides to cancel the premium extension process at any point, the system provides a confirmation prompt. - If the user confirms cancellation, they are returned to the regular application interface without extending their premium subscription.
Non-functional requirement	Security - All payment transactions must adhere to industry-standard security protocols to safeguard user financial information. Availability - The premium extension functionality should be available 24/7 to accommodate users in different time zones and preferences.

Use Case ID	*UC018*
Use Case Name	Pay by VNPAY
Summary	This use case describes the steps involved when a user chooses to make a payment using the VNPAY method
Actors	Registered user
Pre-conditions	The user must be logged into their account. The VNPAY payment gateway must be integrated and accessible within the website.
Post-conditions	The payment is successfully processed. The system updates relevant records to reflect the completed transaction. The user receives confirmation of the successful payment.
Main Flow	1. The user navigates to the payment section or checkout page within the application. 2. The system presents the user with available payment methods, and the user selects VNPAY.

	3. The user is redirected to the VNPAY interface to provide payment details. 4. The user enters the required information, which may include a secure code or authentication method provided by VNPAY. 5. VNPAY validates the payment information and the user's identity. 6. If the payment is successful, VNPAY notifies the system, and the application updates relevant records. 7. The system generates and displays a payment confirmation message to the user.
Alternate Flow(s)	Payment Failure: - If the VNPAY transaction fails or encounters an issue, the user is redirected to an error page with guidance on resolving the problem. - The user may be prompted to review and correct payment details or choose an alternative payment method. Cancellation During Process: - If the user decides to cancel the premium extension process at any point, the system provides a confirmation prompt. - If the user confirms cancellation, they are returned to the regular application interface without extending their premium subscription.
Non-functional requirement	Performance - The interaction with the VNPAY gateway and the overall payment process should be efficient, completing the transaction within a reasonable time frame, typically within a few seconds. Security - All payment transactions must adhere to industry-standard security protocols to safeguard user financial information. Availability - The premium extension functionality should be available 24/7 to accommodate users in different time zones and preferences.

Use Case ID	*UC019*
Use Case Name	Pay by MOMO method
Summary	This use case describes the steps involved when a user chooses to make a payment using the MOMO method
Actors	Registered user
Pre-conditions	The user must be logged into their account. The MOMO payment gateway must be integrated and accessible within the website.
Post-conditions	The payment is successfully processed. The system updates relevant records to reflect the completed transaction. The user receives confirmation of the successful payment.
Main Flow	1. The user navigates to the payment section or checkout page within the application. 2. The system presents the user with available payment methods, and the user selects MOMO.

	3. The user is redirected to the MOMO interface to provide payment details. 4. The user enters the required information, which may include a secure code or authentication method provided by MOMO. 5. MOMO validates the payment information and the user's identity. 6. If the payment is successful, MOMO notifies the system, and the application updates relevant records. 7. The system generates and displays a payment confirmation message to the user.
Alternate Flow(s)	Payment Failure: - If the MOMO transaction fails or encounters an issue, the user is redirected to an error page with guidance on resolving the problem. - The user may be prompted to review and correct payment details or choose an alternative payment method. Cancellation During Process: - If the user decides to cancel the premium extension process at any point, the system provides a confirmation prompt. - If the user confirms cancellation, they are returned to the regular application interface without extending their premium subscription.
Non-functional requirement	Performance - The interaction with the MOMO gateway and the overall payment process should be efficient, completing the transaction within a reasonable time frame, typically within a few seconds. Security - All payment transactions must adhere to industry-standard security protocols to safeguard user financial information. Availability - The premium extension functionality should be available 24/7 to accommodate users in different time zones and preferences.

Use Case ID	*UC020*
Use Case Name	Watch paid movie
Summary	This use case describes the process of a user accessing and watching a paid movie within a streaming application
Actors	Paid viewer
Pre-conditions	The user must be logged into their account. The user's account must have an active subscription or available funds for purchasing the paid movie. The streaming application must be accessible and operational.
Post-conditions	The user successfully watches the paid movie. The system susccessfully load movie
Main Flow	1. The user navigates to the movie page. 2. The user identifies a paid movie they want to watch.

	3. The system checks the user's account for an active subscription. 4. If the user has an active subscription, the system allows the user to select and watch the paid movie immediately. 5. The user starts watching the movie.
Alternate Flow(s)	The user does not have an active subscription. - If the user does not have an active subscription, the system prompts the user to purchase or rent the movie. - The user selects the payment method (credit card, in-app purchase, etc.). - The system processes the payment and grants the user access to the paid movie.
Non-functional requirement	Performance - The process of initiating and authorizing the payment and providing access to the paid movie should occur seamlessly, with minimal delay, to ensure a positive user experience. Security - All payment transactions and user viewing activities should adhere to industry-standard security protocols to protect user financial information and viewing history. Content Availability - The system should ensure that paid movies are available for viewing immediately after a successful payment transaction.

Use Case ID	*UC021*
Use Case Name	Publish Movie
Summary	This use case outlines the process by which an administrator can publish a movie, making it available to users on the website.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged into the system.
Post-conditions	The selected movie is published and becomes available to users on the website.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the movie management section. 3. The administrator selects the publish movie button. 4. The administrator selects the movie to be published from a file or link. 5. The administrator enters movie's details like plot, genres, … 6. The system displays the published movie's details and waits for confirmation from the administrator. 7. The administrator confirms action. 8. The system gets the published movie's data and makes it available to users.
Alternate Flow(s)	The administrator decides not to proceed with the movie publication: 1. The administrator cancels the publishing action.

	2. The system returns to the movie management section without making changes.
Non-functional requirement	- Security: The movie publication process should be secure and require proper authentication to prevent unauthorized publishing. - Visibility: The published movie should be prominently displayed on the website for users to access. - Confirmation: The system should provide a clear confirmation message after the movie publication is successful. - Logging: Logging: The system should log the movie publication action, including the administrator's information and timestamp, for auditing purposes. - User Interface: The movie publication interface should be user-friendly, guiding the administrator through the process with clear instructions and feedback. - Performance: The movie publication process should be efficient and not cause significant delays in system response.

Use Case ID	*UC022*
Use Case Name	Update Movie
Summary	This use case outlines the process by which an administrator can update information about a movie in the system.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged into the system.
Post-conditions	The movie information is updated in the system.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the movie management section. 3. The system presents a list of movies. 4. The administrator selects the movie to be updated. 5. The system displays the current information about the selected movie. 6. The administrator chooses the option to update the movie information. 7. The system deletes the movie and associated data. 8. The system presents a form or interface with editable fields for movie details (e.g., title, release year, genre, synopsis). 9. The administrator modifies the desired movie information. 10. The administrator submits the updated information to the system. 11. The system updates the movie information. 12. The system provides a confirmation message to the administrator.
Alternate Flow(s)	The administrator decides not to proceed with the movie update: 1. The administrator cancels the update action.

	2. The system returns to the movie management section without making changes.
Non-functional requirement	- Security: The movie update process should be secure, requiring proper authentication to prevent unauthorized updates. - Validation: The system should perform validation checks to ensure that the updated information adheres to specified criteria. - Confirmation: The system should provide a clear confirmation message after the movie update is successful. - Logging: The system should log the movie update action, including the administrator's information and timestamp, for auditing purposes. - User Interface: The movie update interface should be user-friendly, guiding the administrator through the process with clear instructions and feedback. - Performance: The movie updating process should be efficient and not cause significant delays in system response.

Use Case ID	*UC023*
Use Case Name	Remove Movie
Summary	This use case outlines the process by which an administrator can initiate the removal of a movie from the system.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged into the system.
Post-conditions	The selected movie is removed from the system, and associated data is deleted.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the movie management section. 3. The system presents a list of movies. 4. The administrator selects the movie to be removed. 5. The administrator confirms the decision to remove the selected movie. 7. The system initiates the movie removal process. 8. The system deletes the movie and associated data. 9. The system provides a confirmation message to the administrator.
Alternate Flow(s)	The administrator decides not to proceed with the movie removal.: 1. The administrator cancels the removal action. 2. The system returns to the movie management section without making changes.
Non-functional requirement	- Security: The movie removal process should be secure and require proper authentication to prevent unauthorized removal. - Data Deletion: The system should permanently delete movie data associated with the removed movie. - Confirmation: The system should provide a clear confirmation message after the movie removal process is completed. - Logging: The system should log the movie removal action, including the administrator's information and timestamp, for auditing purposes. - Performance: The movie removal process should be efficient and not cause significant delays in system response.

Use Case ID	*UC024*
Use Case Name	Pin movies on recommended tabs
Summary	This use case outlines the process by which an administrator can pin specific movies to the recommended tabs for enhanced visibility.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged in.
Post-conditions	The selected movies are pinned to the recommended tabs.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the movie management or recommendation section. 3. The system presents a list of movies or a catalog. 4. The administrator or user selects the movies to be pinned to the recommended tabs. 5. The administrator confirms the decision to pin the selected movies. 6. The system updates the recommended tabs, adding the selected movies. 7. The system provides a confirmation message to the administrator or user.
Alternate Flow(s)	The administrator or user decides not to proceed with pinning the movies: 1. The administrator or user cancels the pinning action. 2. The system returns to the movie management or recommendation section without making changes.
Non-functional requirement	- Security: The pinning action should be secure, and only authorized administrators should have the privilege to pin movies. - Visibility: The pinned movies should be prominently displayed in the recommended tabs for users to easily access. - Confirmation: The system should provide a clear confirmation message after the pinning action is successful. - Logging: The system should log the pinning action, including the administrator's and timestamp, for auditing purposes. - User Interface: The interface for pinning movies should be user-friendly, allowing administrators to easily select and pin movies. - Performance: The pin movies process should be efficient and not cause significant delays in system response.

Use Case ID	*UC025*
Use Case Name	Ban Account
Summary	This use case outlines the process by which an administrator or a system moderator can ban a user account on the website.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged in.
Post-conditions	1. The user account is banned, and the user is denied access to the website.

Main Flow	2. The administrator logs into the system. 3. The administrator navigates to the user management section. 4. The system presents a list of user accounts. 5. The administrator selects the user account to be banned. 6. The administrator confirms the decision to ban the selected user account. 7. The system updates the user account status to "banned." 8. The system sends a notification to the banned user about the account status change. 9. The banned user is denied access to the website.
Alternate Flow(s)	The administrator decides not to ban the user account: 1. The administrator cancels the ban action. 2. The system returns to the user management section.
Non-functional requirement	- Security: The ban action should be secure and only accessible to authorized administrators to prevent misuse. - Notification: The system should promptly notify the banned user about the account status change. - Logging: The system should log the ban action, including the administrator's information and timestamp, for auditing purposes. - Performance: The ban process should be efficient and not cause significant delays in system response.

Use Case ID	*UC026*
Use Case Name	Remove account
Summary	This use case outlines the process by which an administrator can initiate the removal of a user account from the website.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged in.
Post-conditions	The user account is removed from the system, and associated data is deleted.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the user management section. 3. The system presents a list of user accounts. 4. The administrator selects the user account to be removed. 5. The administrator confirms the decision to remove the selected user account. 6. The system initiates the account removal process. 7. The system deletes the user account and associated data. 8. The system provides a confirmation message to the administrator.
Alternate Flow(s)	The administrator decides not to proceed with the account removal: 1. The administrator cancels the removal action. 2. The system returns to the account settings or user management section.
Non-functional requirement	- Security: The account removal process should be secure and require authentication to prevent unauthorized removal. - Data Deletion: The system should permanently delete user data associated with the removed account.

	- Confirmation: The system should provide a clear confirmation message after the account removal process is completed. - Logging: The system should log the account removal action, including the user or administrator's information and timestamp, for auditing purposes. - Performance: The account removal process should be efficient and not cause significant delays in system response.

Use Case ID	*UC027*
Use Case Name	Update account
Summary	This use case outlines the process by which an administrator can initiate the removal of a user account from the website.
Actors	Administrator
Pre-conditions	The system is operational. The administrator is logged in.
Post-conditions	The user's account information is successfully updated in the system.
Main Flow	1. The administrator logs into the system. 2. The administrator navigates to the user management section. 3. The system presents a list of user accounts. 4. The administrator selects the user account for which information needs to be updated 5. The system displays the current information associated with the selected user account. 6. The administrator chooses the option to update the user account information. 7. The system presents a form or interface with editable fields for user details (e.g., username, email, ...). 8. The administrator modifies the desired user account information. 9. The administrator submits the updated information to the system. 10. The system validates the updated information. 11. If validation is successful, the system updates the user account information. 12. The system provides a confirmation message to the administrator.
Alternate Flow(s)	The administrator decides not to proceed with the user account update: 1. The administrator cancels the update action. 2. The system returns to the account settings or user management section. The updated information is not valid: 1. The system displays an error message telling what is invalid. 2. The administrator corrects the invalid field. 3. The administrator submits the updated information to the system. 4. The system validates the updated information. 5. If validation is successful, the system updates the user account information. 6. The system provides a confirmation message to the administrator.

Non-functional requirement	- Security: The user account update process should be secure, requiring proper authentication to prevent unauthorized updates. - Validation: The system should perform validation checks to ensure that the updated information adheres to specified criteria. - Confirmation: The system should provide a clear confirmation message after the user account update is successful. - Logging: The system should log the user account update action, including the administrator's information and timestamp, for auditing purposes. - User Interface: The user account update interface should be user-friendly, guiding the administrator through the process with clear instructions and feedback. - Performance: The account update process should be efficient and not cause significant delays in system response.