

Project Mid Submission: Movie Rating Site

By: Kwan Tou To, Sawyer Bittner, and Trevor Buchanan

GitHub: <https://github.com/Assetvi/Cpts489-project>

Date: 8 March, 2024

Introduction:

This project is a movie rating site. Users will be able to select a movie to rate and provide a review or rating of their own. The site will also allow users to view movies rated and reviewed by others.

Functional Requirements:

Use Cases:

- Log in
- Log out
- Register
- Rate a movie
- Add a movie (if it doesn't exist in the database)
- Search for movies
- Filter movies (alphabetically, date added, genre, etc.)
- Search for users/friends (to add/remove/make recommendations to a friend)
- Add/remove a movie to watch later list
- Recommend a move to a friend

Name of Use Case:	Login		
Created By:	Kwan	Last Updated By:	Kwan
Date Created:	March 07, 2024	Last Revision Date:	March 07, 2024
Description:	This use case describes the process of a user logging into their account on the movie rating site.		
Actors:	Registered User		
Preconditions:	<ul style="list-style-type: none">• The user has a valid account on the movie rating site.• The user is not currently logged in.		
Postconditions:	The user is logged into their account and can access personalized features.		

Flow:	<ol style="list-style-type: none"> 1. The user navigates to the Log in page. 2. The system displays the Log in form with fields for username and password. 3. The user enters their username and password. 4. The user clicks the "Log In" button. 5. The system validates the provided credentials. 6. If the credentials are valid, the system logs the user into their account. 7. The system redirects the user to their personalized dashboard or the homepage.
Alternative Flows:	<p>If the user enters invalid credentials, the system displays an error message and prompts the user to try again.</p> <p>Or if the user is not a Registered User, they can click to Register.</p>
Exceptions:	If the system encounters an error during the login process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The Log in page should have a user-friendly interface with clear instructions. • The system should securely store and handle user credentials. • The system should provide appropriate feedback messages for successful login, invalid credentials, and errors. • The system should implement security measures to prevent unauthorized access and protect user data.

Name of Use Case:	Logout		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user logs out of their profile.		
Actors:	Registered User		
Preconditions:	<ul style="list-style-type: none"> • The user has a valid account on the movie rating site. • The user is currently logged in. 		
Postconditions:	The user is logged out of their account.		
Flow:	<ol style="list-style-type: none"> 1. The user selects the "Log out" option from a dropdown menu in the menu bar, available from any page. 2. The system logs the user out of their account. 		

	3. The system redirects the user to the general homepage.
Alternative Flows:	N/A
Exceptions:	If the system encounters an error during the logout process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The “Log out” option should be in an intuitive location in the menu bar, such as from clicking on the user’s profile icon.] • The option should only be available while the user is logged in.

Name of Use Case:	Register		
Created By:	kwan	Last Updated By:	kwan
Date Created:	March 07, 2024	Last Revision Date:	March 07, 2024
Description:	This use case describes the process of a new user registering for an account on the movie rating site.		
Actors:	New User		
Preconditions:	The user does not have an existing account on the movie rating site.		
Postconditions:	The user has successfully created a new account and can log in.		
Flow:	<ol style="list-style-type: none"> 1. The user navigates to the Register page. 2. The system displays the registration form with fields for username, email, password, and repeat password. 3. The user enters their desired username, email address, password, and repeats the password. 4. The user clicks the "Register" button. 5. The system validates the provided information: <ul style="list-style-type: none"> ○ Checks if the username is available and meets the required format. ○ Verifies that the email address is valid and not already associated with an existing account. ○ Ensures that the password meets the minimum security requirements and matches the repeat password field. 6. If the provided information is valid, the system creates a new user account with the provided details. 7. The system sends a confirmation email to the user's provided email address. 		

	8. The system redirects the user to a success page, confirming their registration and prompting them to log in.
Alternative Flows:	<ul style="list-style-type: none"> • If the username is already taken, the system displays an error message and prompts the user to choose a different username. • If the email address is invalid or already associated with an existing account, the system displays an error message and asks the user to provide a different email address. • If the password does not meet the minimum security requirements or does not match the repeat password field, the system displays an error message and asks the user to enter a valid password.
Exceptions:	If the system encounters an error during the registration process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The Register page should have a user-friendly interface with clear instructions. • The system should validate the uniqueness of usernames and email addresses to prevent duplicate accounts. • The system should enforce password security requirements, such as minimum length and complexity. • The system should securely store user credentials and protect sensitive information. • The system should send a confirmation email to the user's provided email address to verify their account. • The system should provide appropriate feedback messages for successful registration, input validation errors, and system errors.

Name of Use Case:	Rate a Movie		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user can give a movie their personal rating or change a previously given rating. (may be factored into aggregate rating and/or used to help generate recommendations)		
Actors:	Registered User		
Preconditions:	<ul style="list-style-type: none"> • The user has a valid account on the movie rating site. • The user is currently logged in. 		

Postconditions:	The user's personal rating for a movie is posted or updated.
Flow:	<ol style="list-style-type: none"> 1. The user navigates to a movie's page. 2. The user navigates to the "Write a review" button on that page and clicks it. 3. The user is given a control (button, dropdown box) to input a numerical rating for the movie (5 stars, 10/10, etc) and given two text boxes, one for a title, and one for the content of their review. 4. The user inputs their numerical rating, title, and content. 5. The user clicks the "Post" button. 6. The system checks to ensure sufficient data is present for a review (either a numerical rating and/or a title and text rating of sufficient length). 7. If the review is sufficient, it is added to the movie's list of current reviews displayed on the movie page. 8. The page is updated for the user's browser.
Alternative Flows:	<ul style="list-style-type: none"> • If the user has already inputted a review for a movie, the "Write a review" button will be replaced with an "Edit review" button, and upon pressing that button, the fields will be filled with their previous submission to edit. Submitting with new information will replace their previous review in the movie's list. • If insufficient information is present, the system displays an error message and asks the user to either give a numerical rating or a written review with a title and of sufficient length. • If the user clicks the "Cancel" button while inputting their review, the system should ask for confirmation to throw away any current input, then return the user to the movie page.
Exceptions:	If the system encounters an error during the review process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The "Write a review"/"Edit review" button should be plainly visible and in an intuitive location. • The system should prevent the user from posting more than one review per movie. • The system should prevent phenomena such as review bombing and bots by looking for obvious patterns • May want to introduce a waiting period on new accounts for reviewing to help with the problems above • Users should be able to remove their reviews as well

Name of Use Case:	Search for Movies
--------------------------	-------------------

Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user can input search terms into a textbox and receive a selection of movies from the database whose titles, directors, etc. are most similar to the input.		
Actors:	User (registered or otherwise)		
Preconditions:	N/A		
Postconditions:	The user is given a list of movies from the database with some relation to their search terms.		
Flow:	<ol style="list-style-type: none"> 1) The user locates the search bar on any given page. 2) The user inputs the terms they would like to search for. 3) The user clicks the "Search" button. 4) The system compares the search terms to the movies in the database and constructs a sublist of movies, ordered from most relevant to the input to least relevant. 5) The system redirects the user to the search results page and displays the movies in order, with a large number of results requiring multiple pages, which the user can navigate using the index at the bottom of these pages. 		
Alternative Flows:	<ul style="list-style-type: none"> • If the search results in an empty list, the system displays an error message and asks the user to input different terms. • If the user inputs an invalid character for searching, the system displays an error message and asks the user to input different terms. 		
Exceptions:	If the system encounters an error during the search process, it will display an error message and ask the user to try again later.		
Requirements:	<ul style="list-style-type: none"> • The search bar should maintain a consistent position on all pages for the website. • The database should be organized within a data structure that allows for quick searches by the system. • The user should not be able to modify the database through searching alone. • The search algorithm should return a sublist that is significantly smaller in size than the movie database in its entirety. 		

Name of Use Case:	Filter Movies		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user can reduce the amount of movies displayed, either from the database as a whole or a search query, by choosing parameters from a list that the movies must meet.		
Actors:	User (registered or otherwise)		
Preconditions:	N/A		
Postconditions:	The user is given a sublist of movies that meet the set criteria.		
Flow:	<ol style="list-style-type: none"> 1. The user inputs a search query, or navigates to the full movie listing page. 2. The system displays a list of filtering options within a sidebar. 3. The user picks one or more options from the list. 4. The user clicks the “Apply Filters” button at the top of the sidebar. 5. The system creates a sublist composed of only movies that meet all chosen criteria. 6. The system redirects the user to a new results page and displays the sublist. 		
Alternative Flows:	<ul style="list-style-type: none"> • If the chosen filters result in an empty list, the system displays an error message and asks the user to choose different options. 		
Exceptions:	If the system encounters an error during the filtering process, it will display an error message and ask the user to try again later.		
Requirements:	<ul style="list-style-type: none"> • The filter sidebar should be prominently displayed on the side of both the search results page and the full listing page. • The “Apply Filters” button should only be clickable if the user has changed the filters they picked. • The user should have the option to remove all filters. • The original list should be saved, and further filtering should be applied to it.. • The filter options should be organized into labeled categories, such as genre or length. • Filter options within a category may be mutually exclusive, such as whether a movie has been released or not. In these cases, 		

	attempting to select a second of these options will deselect the first option.
--	--

Name of Use Case:	Add a Movie		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user submits information about a film that does not yet exist in the database, which the administrator decides whether to add or not.		
Actors:	Registered User, Admin		
Preconditions:	<ul style="list-style-type: none"> • The user has an account. • The user is logged in. 		
Postconditions:	A new movie is added to the database.		
Flow:	<ol style="list-style-type: none"> 1) The user selects the “Add a movie” option from the dropdown menu in the menu bar. 2) The system redirects them to a movie listing form, with required fields noted with an asterisk. 3) The user fills out the form, and clicks the “Submit” button. 4) The system compares the form to the movie database to confirm that the film does not already exist there. 5) Upon confirmation, the system sends the request to the administrator. 6) The system sends an email to the user confirming that their request is being evaluated. 7) The administrator confirms that the film belongs in the database and is not there already. 8) Upon confirmation, the administrator approves the request. 9) The system adds the film to the database. 10) The system sends an email to the user notifying them that the film has been added. 		
Alternative Flows:	<ul style="list-style-type: none"> • If the system finds that the movie is already in the database, it displays an error message and notifies the user with a link to the existing page. • If the administrator rejects the request, they must input a given reason to the system, which sends an email to the user notifying them of the rejection and reasoning. 		

	<ul style="list-style-type: none"> • If the user does not input the required fields on the form, the system displays an error message and warns the user to fill out those fields.
Exceptions:	If the system encounters an error during the request process, it will display an error message and ask the user/admin to try again later.
Requirements:	<ul style="list-style-type: none"> • The option to add a movie should be in an intuitive place in the menu bar. • The form should have the option to add text and pictures for certain fields, such as director or the movie poster. • The system should be able to send emails to the user • The system should have a method of sending requests to the admin • The admin should be able to approve or deny requests. • The admin may need to be able to request additional info from the user. • The movie pages may also need to be modifiable in the future.

Name of Use Case:	Search for Users/Friends		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user may search through the database of publicly available users or their friends list by entering a name into the search bar.		
Actors:	User (registered or not)		
Preconditions:	N/A		
Postconditions:	The user is given a list of public users or friends based on their input.		
Flow:	<ol style="list-style-type: none"> 1. The user navigates to the users page or friends page. 2. The user inputs the name of a user to search for into the "Name" textbox on the users page. 3. The user clicks the "Search" button. 4. The system creates a sublist of users with a similar name to the input. 5. The system reloads the page for the user with the sublist displayed. 		

Alternative Flows:	<ul style="list-style-type: none"> • If the input does not return any matches, the system outputs an error message telling the user to try a different search term.
Exceptions:	If the system encounters an error during the search process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The system may need a way for users to mark their profiles as private to other users. • The search bar should be prominent at the top of the given pages. • The search algorithm should be able to make a reasonable sublist based on the input and not require exact matches.

Name of Use Case:	Add to/Remove from Watchlist		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/2024	Last Revision Date:	3/8/2024
Description:	The user can add a movie to their personal list of films to watch later, or remove said film.		
Actors:	Registered User		
Preconditions:	<ul style="list-style-type: none"> • The user has an account. • The user is logged in. 		
Postconditions:	The user has added a film to their watchlist, or removed one from it.		
Flow:	<ol style="list-style-type: none"> 1. The user inputs a search query, navigates to the full movie listing page, or for removal, may navigate to their watchlist page. 2. The user locates the film they would like to add/remove. 3. The user may optionally navigate to the film's page. 4. The user clicks the "Add to watchlist"/"Remove from watchlist" button, either next to the movie's listing or on its page. 5. The system adds/removes a link to the film's page to/from the user's watchlist. 6. The system changes the button to its counterpart. 7. If the user clicks the button again, the change is reverted. 		
Alternative Flows:	N/A		

Exceptions:	If the system encounters an error during the listing process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The watchlist should be available and intuitive for the user to find, such as by clicking their profile icon on the menu bar. • The watchlist should function as a specialized filter view of the full movie listing that only shows films the user has tagged. • The button to remove or add a film should be available for every film and should be in an obvious spot. • The watchlist may optionally have recommendations at the bottom of the page that the user can add to their watchlist.

Name of Use Case:	Recommend a Movie		
Created By:	Sawyer	Last Updated By:	Sawyer
Date Created:	3/8/24	Last Revision Date:	3/8/24
Description:	The user can send a link to a movie page as a recommendation to one of their friends on the site.		
Actors:	Registered User(s)		
Preconditions:	<ul style="list-style-type: none"> • The user has an account. • The user is logged in. • The user has at least one friend in their friend list. 		
Postconditions:	The user has sent a recommendation to their friend.		
Flow:	<ol style="list-style-type: none"> 1) The user navigates to the page of the film they'd like to recommend. 2) The user clicks the "Recommend" button on the film page. 3) The system displays a dropdown list of around 5 of the user's friends, chosen by some criteria, and a search bar to search for another friend. 4) The user may optionally search for another friend. 5) The user clicks the icon of the friend to recommend to. 6) The system displays the chosen film and friend, and gives a textbox for the user to input a message to the friend. 7) The user may optionally input a message. 8) The user clicks the "Send" button. 9) The system sends a notification to the friend notifying them of the recommendation. 		

Alternative Flows:	<ul style="list-style-type: none"> • If the user hits the “cancel” button or clicks away from the friend list or message input during the process, that control is hidden, and they must start the process over from the film page.
Exceptions:	If the system encounters an error during the recommendation process, it will display an error message and ask the user to try again later.
Requirements:	<ul style="list-style-type: none"> • The “Recommend” button must be in a visible spot on each film page. • The button may optionally be available on a film’s listing as well. • The process may be possible in reverse i.e. navigating to a friend’s page and choosing a movie to recommend from a list. • The system must be able to send notifications to users. • The system should have some method of deciding which friends to display, such as the first five in alphabetical order or which friends generate the most activity from the user.