

Project Specification for Group TUT0301-03

Team Name: TUT0301-03

Domain: Movies

User Stories:

User story 1: As a user, I want to create an account on a movie website and be able to log in and log out, so that I can browse and watch movies I'm interested in.

User story 2: As a user, I want to add and remove movies from my watchlist, so that I can keep track of the movies I plan to watch

User story 3: As a user, I want to be able to rate movies I have watched and view each movie's overall rating so I can keep track of which movies I have liked and share my opinion while seeing how others have rated it.

User story 4: As a user, I want to leave comments on a movie and reply to other users' comments, so that I can share my thoughts and join discussions about the movie.

User story 5: As a user, I want the homepage to automatically display the 100 most popular movies and recently released movies at any given time, and to have separate pages where I can view the entire list as well as sort them by alphabetically, by average rating, and by the number of ratings.

User story 6: As a user, I want to sign in with my Google account, so that I can log in without entering my password.

User story 7: As a user, I want to follow other users and view ratings, reviews, and watchlists, so that I can discover new movies through my friends.

Use Cases

Use Case 1.1: Create an Account

- Main Flow:
 - User clicks the Create Account button
 - The system will close the Log In window and open the Create Account window
 - User enters the username, password, and confirmed password
 - User clicks the Create button
 - The system saves the user information with username and password
 - The system will close the Create Account window and open the Log In window
- Alternative Flow:
 - If the password and the confirmed password are not the same
 - The system will open a Password Not Same window
 - User clicks the Back button
 - The system will close the Password Not Same window

Use Case 1.2: Log in and Log out of an account

- Main Flow:
 - User enters their username and password
 - User clicks the Log In button
 - The system verifies that the username and password match an existing account
 - The system will close the Log In window and open the account window
 - User clicks the Log out button
 - The system will close the account window and open the Log In window
- Alternative Flow:
 - If the username entered is not saved in the system, and the Log In button is being clicked
 - The system will open a User Not Exist window
 - User clicks the Back button
 - The system will close the User Not Exist window

Use Case 2.0: Manage Watchlists

- Main Flow:
 - User opens the account window (after logging in)
 - User searches for or selects a movie they are interested in
 - User clicks the “Add to Watchlist” button
 - The system adds the selected movie to the user’s watchlist
 - The system displays a confirmation message: “Movie added to your watchlist”
 - User can open the watchlist window to view all saved movies

- User selects a movie from the watchlist and clicks the “Remove from Watchlist” button
 - The system removes that movie from the user’s watchlist
 - The system displays a confirmation message: “Movie removed from your watchlist”
- Alternative Flow:
 - If the movie is already in the user’s watchlist and the “Add to Watchlist” button is being clicked
 - The system opens a “Movie Already in Watchlist” window
 - User clicks the Back button
 - The system closes the warning window and returns to the movie details page
 - If the movie list cannot be loaded (due to missing data or system errors)
 - The system opens a “Failed to Load Watchlist” window
 - User clicks the Retry or Back button
 - The system attempts to reload the watchlist or returns to the Account window

Use case 3.1: Rate a movie (view overall rating)

- Main Flow:
 - User navigates to a movie page
 - System displays movie information and overall rating
 - User can select a rating from 1-5 and submit it
 - System would recalculate the overall rating
 - Updated rating would be displayed
- Alternate flows (connection or database error)
 - The system shows an error message and asks the user to retry

Use case 3.2: Update movie rating

- Main Flow:
 - User navigates to a movie page that they have already rated
 - System displays movie information, overall rating, and their rating
 - The user can select and submit a new rating or delete their rating
 - updates/deletes rating and recalculated the overall rating
- Alternate flows
 - If there is a connection or database error, the system shows an error message and asks the user to retry
 - If the user rates the same rating, the system ignores the update

User case 4.1: Leave comment on a movies and see overall comments

- Main Flow:
 - User navigates to the movie page

- System displays the movie information (title, description, rating, etc.) and a Comments section.
 - System loads and displays existing comments for that movie
 - User types a new comment into the comment input box
 - User clicks the “Post Comment” button and system will saves the new comment
 - System refreshes the Comments section to include the new comment
- Alternate flows:
 - If the user not logged in
 - User clicks “Post Comment”
 - System detects there is no active logged-in session
 - System opens a “Log In” window
 - If the user post empty comment
 - User clicks “Post Comment” without typing anything (or only whitespace)
 - System opens an “Invalid Comment” message “Comment cannot be empty”
 - System keeps that message until the user types valid comment
 - If there is a connection or database error
 - System opens an “Unable to Post Comment” window
 - User clicks “Retry”.
 - System attempts to save the same comment again.
(If retry still fails, system shows the same error window again)

User case 4.2: Reply to a comment and view other replies

- Main Flow:
 - User navigates to a movie page and go to the comment section
 - System displays the Comments section. For each comment, the system shows: commenter username, comment text, comment's replies and a “Reply” button
 - User clicks “Reply” under a specific comment
 - System opens a small reply input box directly under that comment
 - User types a reply and clicks “Post Reply” button and system will saves the new reply
 - System refreshes the Comments section to include the new reply
- Alternate flows:
 - If the user not logged in
 - User clicks “Reply”
 - System detects there is no active logged-in session
 - System opens a “Log In Required” window
 - User clicks “Log In”

- System opens the Log In window
- If the user post empty reply
 - User clicks “Post Reply” without typing anything (or only whitespace)
 - System opens an “Invalid Comment” message “Comment cannot be empty”
 - System keeps that message until the user types valid reply
- If there is a connection or database error
 - System opens an “Unable to Post Comment” window
 - User clicks “Retry”.
 - System attempts to save the same comment again.
(If retry still fails, system shows the same error window again)

User story 5.0: Homepage

- Main Flow:
 - User navigates to homepage
 - Software fetches movie information for the 100 most popular movies using <https://api.themoviedb.org/3/movie/popular> api call
 - Page displays user relevant movie information (title, overview, released date, average ratings, number of ratings) in an organised manner

User story 5.1: Fetching 100 Most Popular Movies

- Main Flow:
 - User navigates to “100 Most Popular Movies” page
 - Software fetches movie information for the 100 most popular movies using <https://api.themoviedb.org/3/movie/popular> api call
 - Page displays user relevant movie information (title, overview, released date, average ratings, number of ratings) in an organised manner
- Alternative Flow:
 - If fewer than 100 movies are returned, software displays a note: Showing N of 100 movies due to limited results.
 - If movies contain missing fields, software omits the missing item, continues, and displays a note: Some items were skipped due to missing data.

User story 5.2: Fetching 100 Most Recently Released Movies

- Main Flow:
 - User navigates to “100 Most Recently Released Movies” page
 - Software fetches movie information for the 100 most popular movies using <https://api.themoviedb.org/3/movie/upcoming> api call, sorts by release date, and returns the 100 most recently released movies
 - Page displays user relevant movie information (title, overview, released date, average ratings, number of ratings) in an organised manner
- Alternative Flow:

- If fewer than 100 movies are returned, redo api call with wider parameters (earlier start date)
- If movies contain missing fields, software omits the missing item, continues, and displays a note: Some items were skipped due to missing data.

Use Story 5.3: Sort the Current Movie List

- Main Flow:
 - User starts at either “100 Most...” page
 - User presses the sort button and selects one of the following options:
 - Alphabetical (A→Z / Z→A) based on movie title
 - Average Rating (descending / ascending)
 - Number of Ratings (descending / ascending)
 - Software re-orders the in-memory list immediately (no need for a network call)
 - Page displays updated list view and highlights the active sort and sort direction
- Alternative Flow:
 - If there are sorting ties, software applies a tiebreaker: sort again by title A→Z
 - If there are missing values e.g. rating or rating count, software pushes it to the end for decreasing sorts or start for ascending sorts and marks the missing field with a “—”
 - If movies contain missing fields, software omits the missing item, continues, and displays a note: Some items were skipped due to missing data.

User case 6.0: Login by Google

- Main Flow:
 - User clicks Login by Google button.
 - The system will open the email confirmation window
 - User enters their Gmail account.
 - User presses the confirm button.
 - The system sends a confirmation email to the Gmail account
 - User confirms login on their Gmail
 - The system closes the email confirmation window.
-

MVPs

| Lead | Use Case | User Story |
|---------|---|---------------|
| Felix | Use Case #1 Account Create and Log in/out | User Story #1 |
| Cyndi | Use Case #2 Manage Watchlists | User Story #2 |
| Rian | Use Case #3 Movie rating and overall rating | User Story #3 |
| Kristin | Use Case #4 Movie comments and replies | User Story #4 |
| Jerome | Use Case #5 Fetching Top 100 Movies and sorting | User Story #5 |
| Felix | Use Case #6 Login by Google | User Story #6 |

Proposed Entities for the Domain

- User
 - Instance:
 - username: String
 - password: String
 - watchlist: Watchlist
 - Uses: Watchlist
- GoogleUser
 - Instance:
 - gmail: String
 - password: null
- Movie
 - Instance:
 - name: String
 - rate: double
 - comments: List<Comment>
 - Uses:
 - Comment
- Watchlist
 - Instance:
 - list: List[Movie]
 - Uses:
 - Movie
- Comment
 - Instance:
 - commentID: String
 - commenter: User
 - parentComment: Comment | null

- text: String
 - MovieLists
 - Instance:
 - length: int
 - name: String
 - List: List<Movie>
 - sort: String
 - Uses:
 - Movie
-

Proposed API for the project

The Movie Database (TMDB)

- Fetch movie details
 - Title
 - Popularity
 - Genre
 - Release date
 - Average rating and number of votes
 - Poster
 - Overview (summary)