

Lab 2 – FlixPicks Product Specification

Team Orange: Cody Rabie, Maulahna Robinson, James Powers,
Logan Blanton, Zachary Carpenter, Roxy Egger, Nicholas Van Tol, Chance Sadler,

Hunter Hindmarsh

Old Dominion University

CS 411 Professional Workforce Development 2

Professor Janet Brunelle

23rd April, 2024

Table of Contents

3.1. Functional Requirements

3.1.1 Account Management

- 3.1.1.1 Login (A: Cody, I: Cody)
- 3.1.1.2 Registration (A: Cody, I: Zac)
- 3.1.1.3 User Profile
 - 3.1.1.3.1 Profile Dropdown (A: Zac, I: Zac)
 - 3.1.1.3.2 Settings Page (A: Zac, I: Cody)
 - 3.1.1.3.3 Change Password (A: Zac, I: Cody)
 - 3.1.1.3.4 Subscription Management (A: Zac, I: Cody)
 - 3.1.1.3.5 Owned Subscription Recommendation Toggle (A: Zac, I: N/a)
 - 3.1.1.3.6 CineMap Toggle (A: Zac, I: N/a)
 - 3.1.1.3.7 Watch History (A: Maula, I: Zac)

3.1.2 Movie Library (How we display content)

- 3.1.2.1 Browse/search Filtering (A: Chance, I: Zac)
- 3.1.2.2 Hotpicks (A: Cody, I: Cody)
 - 3.1.2.2.1 API Request
 - 1. Request content
 - 3.1.2.2.2 Media Display and Selection
 - 1. Display to Library
 - 2. Allow selection
 - 3. Display media content
 - 4. Redirect to hosting platform
- 3.1.2.3 CineRoll (Chance)
- 3.1.2.4 CineWheel (Chance)

3.1.3 Content Filtering (Behind the scenes)

3.1.3.1 Recommendations (Nicholas)

- 3.1.3.1.1 Load Data. description. Clause:
 - 1. User Identification.
 - 2. User watch History
- 3.1.3.1.2 Association Mining
 - 1. Rule Generation
 - 2. Content Matching
- 3.1.3.1.3 Clean Up
 - 1. Filter Watched
 - 2. Schedule Time
 - 3. Update User List

3.1.3.2 Taste Profile (Maula)

- 3.1.3.2.1 Taste Profile Survey
- 3.1.3.2.2 Taste Profile Content-Based Filtering

3.1.4 Database (James/Logan/Nicholas)

3.1.4.1 User Data

- 1 Users
- 2 Interactions (Logan)

3 Watch History (Logan)

4 Users Rating (Logan)

4 Subscriptions (Logan)

3.1.4.2 Recommendations

3.1.4.3 Media Content

1 Movies

2 Genres

3 People

4 Providers

5 Movie-Providers

6 Movie-Genres

7 Movie-Actors

3.1.4.4 Taste Profile

3.1.5 CineMap (A: Roxy, I: Feature cut)

3.1.5.1 Add reactions

3.1.5.2 Add/view comments

3.1.5.3 Heatmap

3.1.6 Analytics (A: Hunter, Roxy I: Hunter)

3.1.6.1 Subscription usage reports

3.1.6.2 User ratings of categories

3.1.6.3 User feedback

3.1.6.4 Media recommendation count

3.1.6.5 CineMap export data

3.1.7 Simulation (Hunter)

3.1.7.1 Users

3.1.7.1.1 Creation of new users

3.1.7.1.2 Editing users watch history

3.1.7.1.3 Editing users TasteProfile

3.1.7.2 Media

3.1.7.2.1 Creation of new media

3.2. Performance Requirements

3.3. Assumptions and Constraints

3.3.1 Assumptions.

1. Prototype will only be offered via web
2. A clicked link is considered watched content
3. Media titles are unique
4. Only movies will be included in the prototype
5. User data will be limited to X,Y,Z and missing QRS
6. X number of user data for recommendations will be generated
7. The service is for free for now
8. Voice commands for user interaction will be avoided
9. Remote control functionality for user interaction will be avoided
10. User explicit interactions will be set to private (watch history, ratings, comments)
11. One-to-one mapping of registered user to user profile

12. CineWheel will only provide movies that are accessible by the registered user
13. Category 1 header on the user dashboard is populated with apriori algorithm
14. Fair use for different subscription services/ inherently promotes popular content providers.
15. Users have fully paid for the subscriptions they are trying to access
16. 300 potential movies will be cached in the database for quick display
17. Movies selected in taste profile survey have been watched by the user

3.3.2 Constraints

1. FlixPicks will not perform 3rd party authentication
2. API data can only be stored for 6 months
3. FlixPicks will not host any video content itself
4. Collaborative filtering of recommendations will not be used within prototype
5. Links to movies will be directed to TMDb 'watch' page instead of directly to content
6. Content offered is limited to what is available from the APIs used.

3.4. Non-Functional Requirements

FLIXPICKS:**3.1. Functional Requirements**

3.1.1 Account Management. (Cody) Contains the tools and features for the user to create, customize, and maintain their FlixPicks account.

3.1.1.1 Login.(A: Cody; I: Cody) Process that compares user provided authentication credentials against a database for login authentication.

3.1.1.1.1 Login Page. 'Login' page displays a form with a login button allowing for user login.

3.1.1.1.2 Form Data. Login form that the user is required to fill out for authentication. The following functional requirements shall be met:

1. The function shall accept user credentials consisting of username and password data submitted through a login form.
2. The form shall include the following fields:
 - a. Username
 - b. Password
3. If the credentials are successfully verified the user homepage shall be redirected by backend.(3.1.1.1.3)
4. If the credentials are not successfully verified, an error message shall be displayed.

3.1.1.1.3 Redirect. Upon successful authentication of form data the user is redirected to the homepage. The following functional requirements shall be met:

1. The user shall be redirected to the 'userhome' endpoint if login is successful.
2. The function shall manage user sessions using Flask-Login.
3. The function shall depend on the Flask framework and extensions.

3.1.1.2 Registration. (Zac) Process that provides the new user with the ability to create unique user credentials.

3.1.1.2.1 Form Data. (A:Zac, I:Cody) Registration form that the user is required to fill out for the creation of a new account. [The following functional requirements shall be met:](#)

1. The form shall include the following fields:
 - a. Email
 - b. First Name
 - c. Last Name
 - d. Username
 - e. Password
 - f. Password Confirmation
2. The page shall send a request to the database to verify that the user provided a unique:

- a. Email
 - b. Username
3. The page shall verify that the user provided the same password in the confirmation as was provided in the original password field.
4. If the fields are successfully verified:
 - a. the account is added to the database
 - b. the user is redirected to the Login Page (3.1.1.1) otherwise,
5. If the fields are not successfully verified an error message shall be displayed.

3.1.1.3 User Profile. (Zac) Options and preferences that users shall be able to adjust to personalize their experience or manage account-related information.

3.1.1.3.1. Profile Dropdown. (A:Zac, I:Zac) When the user hovers the mouse cursor over the circular icon positioned to the right of the header, a menu of options shall appear for user interaction.

1. When the user is not logged in, the following links/options shall be available to the user in the dropdown:
 - a. Login Page (3.1.1.1)
 - b. Register Page (3.1.1.2)
2. When the user is not logged in, the following links/options shall be available to the user in the dropdown:
 - a. Settings Page (3.1.1.3.2)
 - b. Watch History Page (3.1.1.3.7)
 - c. CineMap Toggle (3.1.1.3.7)
 - d. Logout: shall redirect the user to the Guest Home Page and sign the user out of FlixPicks

3.1.1.3.2 Settings Page. (A:Zac, I:Cody) A page that is accessed from the Profile Dropdown and allows the user access to options and preferences that users are able to adjust to personalize their experience or manage account-related information. [The following options shall be available on the Settings Page:](#)

1. Change Password (3.1.1.3.3)
2. Subscription Management (3.1.1.3.4)
3. Owned Subscriptions Recommendation Toggle (3.1.1.3.5)

3.1.1.3.3 Change Password. (A:Zac, I:Cody) From the Settings Page the user has access to the option to change their password by filling out a form. The following functional requirements shall be met:

1. The form shall include the following fields:
 - a. New Password
 - b. New Password Confirmation

2. The page shall verify that the user provided the same password in the confirmation as was provided in the original password field, and shall update the user password.

3.1.1.3.4 Subscription Management. (A:Zac, I:Cody) From the Settings Page (3.1.1.3.2) the Subscription Management option allows the user to change the list of streaming services that they are subscribed to by filling out a form. The following functional requirements shall be met:

1. The form shall include a button for:
 - a. Netflix
 - b. Hulu
 - c. Disney+
 - d. Amazon Prime Video
 - e. Max
2. The user shall be able to edit their subscriptions by either adding or removing subscriptions and filling in the subscription name in the form.
3. Clicking the submit button shall send a push to the database to update the user's list of subscriptions based on what they selected.
4. This change shall affect what media is recommended to the user by only showing content that the user could view through their subscriptions unless the *Owned Subscription Toggle* (3.1.1.3.4) is set to off.

3.1.1.3.5 Owned Subscription Recommendation Toggle. (A:Zac, I:N/a) From the Settings Page, the user shall be able to access a toggle that changes what recommended content is visible:

Owned Subscription Toggle On	Owned Subscription Toggle Off
The only media that shall be recommended to the user is content that is viewable through their subscriptions.	All media, regardless of the user's subscriptions, shall be recommended to the user.

3.1.1.3.6 CineMap Toggle. (A:Zac, I: N/a) From the Profile Dropdown, the user shall be able to access a toggle that changes whether or not CineMap (3.1.5) is visible:

CineMap Toggle On	CineMap Toggle Off
When the user is redirected to the media of their choosing, the CineMap overlay (3.1.5) shall be visible.	When the user is redirected to the media of their choosing, the CineMap overlay (3.1.5) shall not be visible.

3.1.1.3.7 Watch History (A:Maula, I:Zac, Nicholas) A page in FlixPicks where users can see movies they have watched and the movie options they ranked in the Taste Profile Survey. Users can access the watch history at any time to track the media they are watching and to change whether they like or dislike the media.

1. Watch History shall contain movies that users select from the Taste Profile.
2. Movies that are watched shall automatically be added to the Watch History.
3. Movies removed from the Watch History which shall change the recommendation presented to users.
4. Users shall select if they liked or disliked movies in their watch history.

3.1.2 Movie Library (How we display content)

3.1.2.1 Browse/search Filtering (A:Chance, I:Zac) A page that allows users to browse the media library through searching for movies. Users can search for the movies they are interested in through the use of a search bar. [The following functional requirements shall be met:](#)

1. The system shall allow the user to browse the library of media
2. The system shall allow the user to search for a particular media
3. The system shall support filtering by category

3.1.2.2 HotPicks. A section of the user library that displays options of popular media not based on user specific recommendations.

3.1.2.2.1 API Request. Process for retrieving media information from the backend database. [The following functional requirements shall be met:](#)

1. The algorithm shall formulate a request to the api retrieve popular_media.

3.1.2.2.2 Media Display and Selection. Process for allowing user selection and displaying of media information from the HotPicks library.

[The following functional requirements shall be met:](#)

1. The system shall display popular_media in the HotPicks section of the Library.
2. The system shall provide the ability for users to select a tile representing a movie through its title, and be provided with the media content.
3. The media content shall consist of the following:
 - a. Title
 - b. Date of Publication
 - c. Description
 - d. URL
4. The system shall redirect the user to the platform hosting the selected media or video after selection.

3.1.2.3 CineRoll. (A:Chance, I:Chance, Nicholas) Process to allow a user to randomly select media based on their recommendations. [The following functional requirements shall be met:](#)

1. A button shall be provided to activate the functionality of CineRoll
2. The system shall pseudo randomly visually present a movie from the user's recommendations

3.1.2.4 CineWheel. (A:Chance, I:Chance) A feature that provides the user with the ability to enter a list of movie titles to pseudo randomly choose one. [The following functional requirements shall be met:](#)

1. The system shall provide the ability for users to add movie titles to a simulated spinning wheel
2. The wheel shall only contain up to 13 movie titles
3. The system shall provide the ability to spin the wheel
4. The wheel shall spin at a pseudo random amount of time between 4 and 8 seconds
5. Upon stopping the wheel shall search for the movie title in a new tab using the search feature

3.1.3 Content Filtering FlixPicks content filtering shall determine which content is suggested for all registered users. All data from content filtering shall be stored in the database.

3.1.3.1 Recommendations (A: Nicholas; I Nicholas) An algorithm that provides content recommendations based on a user's Watch History and ruleset created by association mining. Any rules in a ruleset that match a user's Watch History shall be stored after filtering out watched content.

[The following functional requirements shall be met:](#)

1. The system shall collect and maintain records of user watch history (3.1.4.1.3).
2. The algorithm shall utilize Apriori association mining to identify frequent itemsets in all users' watch history.
3. The system shall provide the ability to the administrator to adjust support thresholds.(default = .1)
4. The system shall provide the ability to the administrator to adjust confidence thresholds.(default = .1)
5. The algorithm shall use frequent itemsets identified to establish connections between content to generate a ruleset.

6. The system shall generate content combinations for an individual user's watch history.
7. The system shall match content combinations from a user's Watch History to the algorithm ruleset.
8. The system shall filter out any content that has already been watched from matched content.
9. The system shall store filtered matched content as recommendations for a user (3.1.4.2)

3.1.3.2 Taste Profile. (Maula) The Taste Profile is the initial data that users will provide to FlixPicks. The Taste Profile collects information on the subscription services that users are subscribed to. The Taste Profile also collects information on movies that the user has watched in the past.

3.1.3.2.1 Taste Profile Survey. The Taste Profile Survey is a questionnaire that users shall fill out upon making an account. The survey consists of questions about subscribed subscription services and movie preferences. The following requirements shall be met:

1. The Taste Profile must be completed upon registering for an account.
 - a. If skipped, the next time users log in they shall be prompted to fill out the survey.
2. The content of the fields shall be populated in the database (3.13)
3. The Taste Profile shall provide the user with the ability to select from the following subscriptions:
 - a. Netflix
 - b. Hulu
 - c. MAX
 - d. Disney+
 - e. Prime Video
 - f. ODU Live
4. The Taste Profile shall ask users to select movies they have watched from a variety of options.
5. Movies selected from the Taste Profile Survey shall be automatically added into the watch history.

3.1.3.2.2 Taste Profile Content-Based Filtering. Content based filtering shall be implemented to determine what users like and dislike based on their interactions with media and from their selections of the Taste Profile Survey. The following requirements shall be met:

1. Initial movie data shall be collected from filling out the Taste Profile Survey and shall be stored in the recommendation table.

2. The Taste Profile shall use survey data with Apriori to generate recommendations for users.
3. The initial Taste Profile Survey movie data shall be sent to the watch history for viewing and stored in the recommendations table after completing the Taste Profile Survey.

3.1.4 Database. (James/Logan/Nicholas) The FlixPicks database shall store all user data, movie metadata, and recommendations for each Registered User. All data shall be stored using an SQLite database and each field shall be designated to store specific data types. The database shall implement an API providing endpoints for frontend and backend components.

3.1.4.1 User Data. (James) User data consists of information stored about FlixPicks Registered Users. User Data includes authentication data, personal preferences, past interactions, watch history, and user subscriptions. The following database structures shall be implemented:

1 Users. The Users table shall store user profile information including the following fields and corresponding data types:

Field Name	Data Type	Description
id	Integer	Unique identifier for each user profile. Alternatively referred to as “user_id” when referenced in other tables.
username	String	User provided alias used for authentication. Each username must be unique for each user profile.
email	String	Registered User’s email address used for communication. E-mail addresses must be unique for each user profile.
Password	String	Encrypted or hashed form of user provided password used for authentication. Passwords are encrypted using bcrypt password-hashing function.
fname	String	Registered User’s first name.

Field Name	Data Type	Description
lname	String	Registered User's last name.
registration_date	Datetime	Date and time the user profile was created.
role	String	Defines the user's role on FlixPicks. Available options include Standard, Representative, Tester, and Administrator.
last_login	Datetime	Date and time of last successful login.
limit_subscriptions	Boolean	User preference to limit displayed media to the user's identified subscriptions. Default value is true for all new users.

2 Watch History. (A: Nicholas; I: Nicholas) The Watch History table shall store information regarding content that has been watched by users. [The watch history table includes the following fields and corresponding data types:](#)

Field Name	Data Type	Description
id	Integer	Unique identifier automatically generated when a content is watched. This value is the primary key.
movie_id	Integer	ID of the content that was watched. Foreign key to the movies table.
user_id	Integer	ID of the user to which the content was watched belongs to. Foreign key to the users table
watched_at	Datetime	Date and Time the content was watched.

from_recommended	Bool	Tracks if the content that was watched had been recommended to the user.
suppressed	Bool	Tracks if the content is being suppressed from user watch history calculations. Data is kept to ensure historical integrity.

3 User Interactions. (Logan) The system shall capture, store, and retain both *implicit* and *explicit* user interactions. The User Interaction Table stores explicit and implicit user interactions throughout the FlixPicks app. The data will be utilized by backend systems to drive personalization, user experience workflows, and content recommendations.

1 User Interactions Event Store. The explicit user interaction dataset shall contain persistent data. The implicit dataset is temporarily stored as an input for *online analytical processing* (OLAP). The following database structures shall be implemented:

Field	Data Type	Interaction Type	Public Visibility	Description
user_id	Integer	n/a	System	ID of the user to which the content was watched
movie_id	Integer	n/a	System	ID of the content that was watched. Foreign key to the movies table.
media_paused	Datetime	Explicit/Implicit	System	Records timestamps when media content is paused.
trailer_watched	Boolean	Implicit	System	Indicates whether a user has watched a trailer for a particular media content,

4 User Ratings. (A: Nicholas; I Nicholas) The user rating table shall store information regarding content that has been rated by users. [The](#)

user rating table includes the following fields and corresponding data types:

Field Name	Data Type	Description
id	Integer	Unique identifier automatically generated when a content is rated. This value is the primary key.
movie_id	Integer	ID of the content that was rated. Foreign key to the movies table.
user_id	Integer	ID of the user to which the content was rated belongs to. Foreign key to the users table
user_liked	Boolean	Rating of the content provided by the user. True is the user liked the content. False is the user did not like the content.

5 User Subscriptions. (Logan) When a Registered User selects or modifies their preferred media content providers, the information shall be stored within the Users Subscriptions table. Specifics for Provider data are listed in 3.1.4.3.4.

Field Name	Data Type	Description
user_id	Integer	ID of the user to which the content was rated belongs to. Foreign key to the users table
provider_id	Integer	Unique identifier for each service provider
provider_name	String	Service provider name

3.1.4.2 Recommendations (A: Nicholas; I: Nicholas) The recommendations table shall store all information regarding recommendations made for users. The recommendations table includes the following fields and corresponding data types:

Field Name	Data Type	Description
------------	-----------	-------------

id	Integer	Unique identifier automatically generated when a new recommendation is made. This value is the primary key.
movie_id	Integer	ID of the recommended movie that is provided by the recommender system. Foreign key to the movies table.
user_id	Integer	ID of the user to which the recommendation is being made for. Foreign key to the users table
recommended_at	Datetime	Date and Time the recommendation was made.

3.1.4.3 Media Content. (James) Consists of all metadata and title information stored about movies known to FlixPicks. The following database structures shall be implemented:

1 Movies. The Movies table shall store a list of movies including the following fields and corresponding data types:

Field Name	Data Type	Description
movie_id	Integer	Unique identifier for each movie
tmdb_id	Integer	Unique identifier for each movie used by TMDB
title	String	Title of movie
summary	String	Brief plot summary
release_date	Date	Date movie was initially released Format: yyyy-dd-mm
runtime	Integer	Length of movie in seconds
adult	Boolean	Identifies if a movie contains adult content
poster_path	String	Path to primary poster image
backdrop_path	String	Path to primary background
language	String	Two character code to identify primary language (EN = English)

2 Genres. The Genres table shall store a list of genres including the following fields and corresponding data types:

Field Name	Data Type	Description
genre_id	Integer	Unique identifier for each genre
tmdb_id	Integer	Unique identifier for each genre used by TMDB
genre_name	String	Genre/Category Name

3 People. The People table shall store a list of actors and directors including the following fields and corresponding data types:

Field Name	Data Type	Description
person_id	Integer	Unique identifier for each person
tmdb_id	Integer	Unique identifier for each person used by TMDB
name	String	Person's full name
profile_image	String	Path to primary profile image
biography	String	Person's biography

4 Providers. The Providers table shall store a list of streaming service providers including the following fields and data types:

Field Name	Data Type	Description
provider_id	Integer	Unique identifier for each service provider
tmdb_id	Integer	Unique identifier for each service provider used by TMDB
display_priority	Integer	Display priority used by sorting
provider_name	String	Service provider name
logo_path	String	Path to service provider's logo
region	String	Two character code denoting regional availability (EN=English)

5 Movie-Providers. The Movie-Providers table shall establish relationships between the Movies and Providers tables using the following fields and corresponding data types:

Field Name	Data Type	Description
id	Integer	Unique entry identifier
movie_id	Integer	Movie identifier
provider_id	Integer	Provider identifier
link	String	Link to access movie

6 Movie-Genres. The Movie-Genres table shall establish relationships between the Movies and Genres tables using the following fields and corresponding data types:

Field Name	Data Type	Description
id	Integer	Unique entry identifier
movie_id	Integer	Movie identifier
genre_id	Integer	Genre identifier

7 Movie-Actors. The Movie-Actors table shall establish relationships between the Movies and People tables using the following fields and corresponding data types:

Field Name	Data Type	Description
id	Integer	Unique entry identifier
movie_id	Integer	Movie identifier
person_id	Integer	Person identifier
character	String	Name of character portrayed in the movie

8 Movie-Directors. The Movie-Directors table shall establish relationships between the Movies and People tables using the following fields and corresponding field types:

Field Name	Data Type	Description
id	Integer	Unique entry identifier
movie_id	Integer	Movie identifier
person_id	Integer	Person identifier

3.1.4.4 Taste Profile

3.1.5 CineMap (Roxy) An overlay that shall go on top of streamed content that shall allow users to see the most replayed sections of that content and see and leave comments and reactions.

3.1.5.1 Reactions

1. CineMap shall allow users to leave reactions on media at certain timestamps.
2. CineMap shall allow users to view reactions on media at certain timestamps.
3. CineMap shall save user reactions associated with a certain piece of media to the database.

3.1.5.2 Comments

1. CineMap shall allow users to leave comments on media.
2. CineMap shall allow users to view comments left on media.
3. CineMap shall save comments left on a certain piece of media to the database.

3.1.5.3 Heatmap

1. The CineMap shall save what parts of the content users are watching that they replay to the database.
2. The CineMap shall allow users to view the most replayed sections of media.

3.1.6 Analytics (Hunter/Roxy) The system shall store analytics data in the backend database for several categories such as subscription data, ratings, feedback, recommendations, and CineMap data.

3.1.6.1 Subscription Usage Reports The system shall allow for the data of what subscriptions users use to be stored for analytics purposes.

3.1.6.1.1 Data Retrieval. Process for fetching subscription usage data from the backend database **tables (3.1.4)**. The following functional requirements shall be met:

1. The system shall retrieve subscription **usage data** from the backend database.
2. The system shall organize subscription usage data by relevant parameters:

- a. time period,
- b. user demographics,
- c. subscription plans.

3.1.6.2 User Ratings of Categories. Collection of Ratings. Process for gathering user ratings on different content categories. The following functional requirements shall be met:

1. The system shall provide users the ability to rate content categories with a like or dislike after consuming media.
2. The rating shall be saved for users with media content in their watch history.
3. The system shall record and store user ratings in the backend database for analysis.

3.1.6.3 User Feedback

3.1.6.3.1 Feedback Submission. Process for users to submit feedback regarding the platform's features and content. The following functional requirements shall be met:

1. The system shall provide a feedback form accessible to users.
2. The system shall store user feedback in the backend database for further analysis.

3.1.6.4 Media Recommendation Count. Process for tracking the number of recommendations for each media item. The following functional requirements shall be met:

1. The system shall increment the recommendation count each time a user recommends a media item.
2. The system shall store the recommendation count for each media item in the backend database.

3.1.6.5 CineMap Export Data. Process for exporting data to CineMap format for external analysis. The following functional requirements shall be met:

1. The system shall format analytics data according to CineMap specifications.
2. The system shall export formatted data to CineMap-compatible files for external use.

3.1.7 Simulation (Hunter)

3.1.7.1 Users

3.1.7.1.1 Creation of New Users

3.1.7.1.1.1 Registration Form. Process for users to register and create new accounts. The following functional requirements shall be met:

1. The system shall provide a registration form for new users to input required information.
2. The system shall validate user inputs and store new user data in the backend database upon successful registration.

3.1.7.1.2 Editing Users' Watch History

3.1.7.1.2.1 Edit Interface. Process for users to modify their watch history. The following functional requirements shall be Met:

1. The system shall provide a user-friendly interface for users to view and edit their watch history.
2. The system shall update the backend database to reflect changes made to users' watch history.

3.1.7.1.3 Editing Users' Taste Profile

3.1.7.1.3.1 Profile Modification. Process for users to adjust their taste preferences profile. The following functional requirements shall be met:

1. The system shall present users with options to modify their taste profile parameters such as preferred genres, actors, and directors.
2. The system shall save updated taste profiles in the backend database for personalized recommendations.

3.1.7.2 Media**3.1.7.1 Creation of New Media**

3.1.7.1.1 Content Submission. Process for content creators to submit new media to the platform. The following functional requirements shall be met:

1. The system shall provide a submission form for content creators to upload media files and input relevant metadata.
2. The system shall validate submitted media content and store it in the backend database upon successful submission.