

**Lab 3 – FlixPicks Prototype Test Plan**

Team Orange

CS411W

Professor J. Brunelle

April 11, 2024

Version 1

## **Table of Contents**

1. Objectives
2. References
3. Test Procedures
  - 3.1 Account Management
    - 3.1.1 Account Creation Test (O:Cody Rabie)
    - 3.1.2 User Login Test (O:Cody Rabie)
    - 3.1.3 Guest User Profile Dropdown Test (O: Zachary Carpenter)
    - 3.1.4 Registered User Profile Dropdown Test (O: Zachary Carpenter)
    - 3.1.5 User Profile Settings Test (O:Cody Rabie)
  - 3.2 Movie Library
    - 3.2.1 HotPicks Test (O:Cody Rabie)
  - 3.3 Content Filtering
    - 3.3.1 Recommender (O:Nicholas Vantol)
    - 3.3.2 Taste Profile
  - 3.4 Database
    - 3.4.1 Users Table (O:James Powers)
    - 3.4.2 Watch History (O:James Powers)
    - 3.4.3 Media Content (O: James Powers)
    - 3.4.x (logan)
  - 3.5 Analytics (Hunter/Roxy)
    - 3.5.1 Data retrieval tests
    - 3.5.2 User rating tests
    - 3.5.3 User feedback tests
    - 3.5.4 Media recommendation count tests
  - 3.6 Simulation

## 1. Objectives

The primary objective of this test plan is to ensure the approach to developing and testing the FlixPicks Prototype can lead to a successful demonstration of the FlixPick's product operation. FlixPicks is an internet application that can be accessed from any device able to support a modern web browser. FlixPicks will make the movie streaming experience easier and more enjoyable.

## 2. References

Team Orange. (\_\_\_\_). Lab 1 - FlixPicks

Retrieved \_\_\_\_, from

Team Orange. (\_\_\_\_). Lab 2 - FlixPicks

Retrieved \_\_\_\_, from

### 3. Test Procedures

#### 3.1 Account Management

##### 3.1.1 Account Creation Test

<b>Test Category:</b> System test		<b>Description:</b> This test will verify that a user can create an account and successfully have their information posted to the database.		
<b>Test Case:</b> 3.1.1		<b>Case Name:</b> Account Creation	<b>Version:</b> 1.0	<b>Written By:</b> Cody Rabie
<b>Requirements Fulfilled:</b> (3.1.1.2)		<b>Purpose:</b> This test verifies that a user can create an account through the registration page. Successful account creation requires that the provided username does not already exist in the database and that all required fields (username, password, name, email) are completed.		
<b>Setup Conditions:</b> FlixPicks registration page is open				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user clicks the “Register” button on the dashboard (3.1.1.2.1)			Registration page loads with empty fields for the user to input the username, password, name, and email address.
2	The user enters a pre-existing username (3.1.1.2.1)			The user shall be notified that the missing field(s) information is required and will not be able to submit a registration request.

3	The user leaves any of the fields blank (3.1.1.2.1)			The user shall be notified that the missing field(s) information is required and will not be able to submit a registration request.
4	The user fills out all of the form entries and does not pick a pre-existing username (3.1.1.2.1)			The user will successfully register and their information will be stored in the database as an account.

### 3.1.3 Guest User Profile Dropdown Test

<b>Test Category:</b> System Test		<b>Description:</b> This test verifies that the dropdown in the header of the FlixPicks Guest Home Page works correctly.		
<b>Test Case:</b> 3.1.3		<b>Case Name:</b> Guest User Profile Dropdown	<b>Version:</b> 1.0	<b>Written By:</b> Zachary Carpenter
<b>Requirements Fulfilled:</b> 3.1.1.3.1.1		<b>Purpose:</b> This test ensures that all the options in the profile dropdown on the Guest Page redirect to the correct locations.		
<b>Setup Conditions:</b> 1.   Navigate to the Guest Home Page (Open FlixPicks for the first time, or log out)				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	Hover the mouse over the circular icon labeled “Guest” in the top right corner of the Guest Home Page.			A dropdown appears containing options for “Login” and “Register”
2	Click on “Register”.			FlixPicks redirects to the “Register” page which displays a field for each of the following: 1. Email

				2. First name 3. Last name 4. Username 5. Password 6. Password-Conformation
3	Hover the mouse over the circular icon labeled “Guest” in the top right corner of the Register Page.			A dropdown appears containing options for “Login” and “Register”
4	Click on “Login”.			FlixPicks redirects to the “login” page which displays a field for both of the following: 1. Username 2. Password

### 3.1.4 Registered User Profile Dropdown Test

<b>Test Category:</b> System Test		<b>Description:</b> This test verifies that the dropdown in the header of the FlixPicks Registered User Home Page works correctly.		
<b>Test Case:</b> 3.1.3		<b>Case Name:</b> Registered User Profile Dropdown	<b>Version:</b> 1.0	<b>Written By:</b> Zachary Carpenter
<b>Requirements Fulfilled:</b> 3.1.1.3.1.2		<b>Purpose:</b> This test ensures that all the options in the profile dropdown on the Registered User Page redirect to the correct locations.		
<b>Setup Conditions:</b> <ul style="list-style-type: none"><li>1. Navigate to the Registered User Home Page (Sign in)</li><li>2. If this is the first time signing in on this account: complete the Taste Profile Survey</li></ul>				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	Hover the mouse over the circular icon labeled with the current user’s first name in the top right corner of the Registered User Home Page.			A dropdown appears containing options for “Settings”, “Watch History” and “Logout”

2	Click on “Settings”.			FlixPicks redirects to the “Settings” page which displays the following: 1. Change Password 2. Subscription Management
3	Hover the mouse over the circular icon labeled with the current user’s first name in the top right corner of the Settings Page.			A dropdown appears containing options for “Settings”, “Watch History” and “Logout”
4	Click on “Watch History”.			FlixPicks redirects to the “Watch History” page which displays the following: 1. Watched Category 2. Liked Category 3. Disliked Category
5	Hover the mouse over the circular icon labeled with the current user’s first name in the top right corner of the Settings Page.			A dropdown appears containing options for “Settings”, “Watch History” and “Logout”
6	Click on “Logout”.			FlixPicks redirects to the “Guest Home” page

### 3.1.5 User Profile Settings Test

<b>Test Category:</b> System test	<b>Description:</b> This test verifies that the user profile settings page and functions display and function properly		
<b>Test Case:</b> 3.1.5	<b>Case Name:</b> User Profile Settings	<b>Version:</b> 1.0	<b>Written By:</b> Cody Rabie
<b>Requirements Fulfilled:</b> (3.1.1.3.2)	<b>Purpose:</b> To ensure the user profile settings page displays correctly. Ensures a user has the options to change their password, view current subscriptions, add subscriptions, and remove subscriptions.		

<b>Setup Conditions:</b> 1. FlixPicks application is open to the Settings page (3.1.1.3.2)				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user clicks the “Settings” button on the Userhome page (3.1.1.3.2).			<p>User profile settings page loads and displays options to change password, add subscriptions, and remove subscriptions. Current subscriptions are also displayed on this page.</p> <p>The user’s current username is displayed in the top right header.</p>
2	The user types in the “Change Password” field and presses submit (3.1.1.3.2).			A prompt stating the user password has been successfully changed is displayed, and the user is able to freely access the settings page again.
3	The user types nothing in the “Change Password” field and presses submit (3.1.1.3.2).			An error message stating that the field is empty is displayed.
4	The user clicks the add subscriptions button (3.1.1.3.2).			A text box prompt displays, asking the user for the name of the subscription service they wish to add.
	The user types a valid streaming service that isn’t already in their current subscriptions (3.1.1.3.2).			A successfully added subscription message is displayed and the service is added to the users services. The current subscriptions information is updated to include the new service.



	The user instead types in a non valid input (3.1.1.3.2)..			An error message stating that this is not a valid streaming service is displayed.
	The user instead types a streaming service that already exists in the list (3.1.1.3.2).			An error message is displayed, and the user is redirected to the settings page.
	The user clicks the remove subscriptions button (3.1.1.3.2).			A text box prompt displays, asking the user for the name of the subscription service they wish to remove.
	The user types a valid streaming service that is already in their current subscriptions (3.1.1.3.2).			A successfully removed subscription message is displayed and the removal is reflected in the current subscriptions information displayed on the page.
	The user instead types non valid input (3.1.1.3.2).			An error message is displayed, and the user is redirected to the settings page.
	The user instead types a streaming service that does not exist in the current subscriptions (3.1.1.3.2).			An error message is displayed, and the user is redirected to the settings page.

## 3.2 Movie Library

### 3.2.1 HotPicks

<b>Test Category:</b> System test		<b>Description:</b> This test verifies that the HotPicks media category displays popular movie options to both the guest and registered user UI correctly.		
<b>Test Case:</b> 3.2.1		<b>Case Name:</b> HotPicks Test	<b>Version:</b> 1.0	<b>Written By:</b> Cody Rabie
<b>Requirements Fulfilled:</b> (3.1.2.2 )		<b>Purpose:</b> To ensure that the popular movie options are displayed properly in the HotPicks media category. It also ensures that these options can be selected, then loading the mediaInfo page.		
<b>Setup Conditions:</b> <div>1. FlixPicks application is open.</div> <div>2. The FlixPicks guest homepage UI is open.</div> <div>3. The FlixPicks registered user homepage UI is open.</div>				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	The user loads the guest home page.			The HotPicks category displays title and poster information for popular_movies. Each movie is separated into individual tiles that can be selected.
2	The user loads the registered user home page.			The HotPicks category displays title and poster information for popular_movies. Each movie is separated into individual tiles that can be selected.
3	The user selects a tile from the HotPicks category.			The ‘mediaInfo’ page submits a ‘GET’ request to the FlixPicks API to retrieve media information about the selected movie.

				The mediaInfo page loads with title, plot, release date, genre, and where-to-watch information from the selected movie.
--	--	--	--	---

### 3.3 Content Filtering

#### 3.3.1 Recommender

<b>Test Category:</b> System Testing		<b>Description:</b> This test will verify that the recommender generates recommendations for a user based on their watch history		
<b>Test Case:</b> 3.3.1		<b>Case Name:</b> Recommender	<b>Version:</b> 1.0	<b>Written By:</b> Nicholas Vantol
<b>Requirements Fulfilled:</b> 3.1.3.1		<b>Purpose:</b> To ensure that the recommender generates recommendations based on watch history		
<b>Setup Conditions:</b> <ul style="list-style-type: none"><li>1. FlixPicks application is open</li><li>2. FlixPicks simulation page is open</li><li>3. User registration, login, and taste profile completion is known</li></ul>				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	On simulations press generate watch history			Popup window appears with simulate watch history and operations
2	On popup, keep default operation of preset patterns selected			Preset patterns button is highlighted
3	Click Run Simulation button			Confirmation of preset patterns being generated appears
4	On flixpicks application, register an account and log in in accordance to test case x.x.x			User has a homepage with no recommendations

5	User types a movie, “UP”, into the “Search for a movie...” box in the top right hand corner of the home page			“Up” appears in the search line
6	User clicks the search icon			Movies including the key word “UP” now appear on the screen
7	User clicks the movie “UP”			A description of UP, where to watch, a poster, and a TMDB link appear
8	User clicks on the TMDB link			User is directed to the TMDB page in a new tab
9	On media description page user click “FLixPicks” located in the top left			User is returned to the homepage
10	User repeats steps 5-9 three more times, replacing “UP” with the following titles: “The Conjuring: The devil made me do it”, “Whiplash”, “SaltBurn”			User is on homepage
11	User waits on the homepage for around one minute and then refreshes the page			Recommended content now displays “Jurassic World: Fallen Kingdom” and “Mulan”

### 3.4 Database

#### 3.4.1 Users Table

<b>Test Category:</b> System Test		<b>Description:</b> This test verifies the Users table exists and fields are properly mapped with FlixPicks backend systems.		
<b>Test Case:</b> 3.4.1		<b>Case Name:</b> Users Table	<b>Version:</b> 1.0	<b>Written By:</b> James Powers
<b>Requirements Fulfilled:</b> 3.1.4.1		<b>Purpose:</b> To ensure Users table exists and fields are properly mapped when a new users are added.		
<b>Setup Conditions:</b> 1. FlixPicks docker containers are currently running locally for testing. 2. FlixPicks database home page is open (http://localhost:3000)				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	Select Tables from the top menu bar.			List of available tables are shown.
2	From the User Data menu, choose Users.			User table data is shown with default data present.
3	Visually inspect presented data to ensure the following fields are present: id, username, fname, name, role, email, registration_date, last_login, password			All fields identified have been validated.
4	Navigate to the Simulations page using the top menu bar.			Run simulations options are shown.
5	Press the Generate Users button.			Simulate Users popup dialog is displayed showing options.

6	Press Run Simulations, leaving the default options as is.			Simulated users will be added and a green confirmation message is displayed on the Simulations page.
7	Navigate to Tables using the top menu bar, then choose Users from the User Data menu.			User table data is shown with current data present.
8	Visually inspect the table to ensure a series of new simulated users have been generated and all fields are consistent with their description from requirement 3.1.4.1.			All fields and new users have been verified.

### 3.4.2 Watch History

<b>Test Category:</b> System Test		<b>Description:</b> This test verifies the Watch History table exists and fields are properly mapped with FlixPicks backend systems.		
<b>Test Case:</b> 3.4.2		<b>Case Name:</b> Watch History	<b>Version:</b> 1.0	<b>Written By:</b> James Powers
<b>Requirements Fulfilled:</b> 3.1.4.1		<b>Purpose:</b> To ensure the Watch History table exists and fields are properly mapped when new entries are added.		
<b>Setup Conditions:</b> 1. FlixPicks docker containers are currently running locally for testing. 2. FlixPicks database home page is open ( <a href="http://localhost:3000">http://localhost:3000</a> )				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	Select Tables from the top menu bar.			List of available tables are shown.
2	From the User Data menu, choose Watch History.			Watch History table data is shown with default data present.

3	Visually inspect presented data to ensure the following fields are present: id, user_id, movie_id, watched, from_recommended, suppressed			All fields identified have been validated.
4	Navigate to the Simulations page using the top menu bar.			Run simulations options are shown.
5	Press the Generate Watch History button.			Simulate Watch History popup dialog is displayed showing default options.
6	Press Run Simulations, leaving the default options as is.			Simulated watch history will be added and a green confirmation message is displayed on the Simulations page.
7	Navigate to Tables using the top menu bar, then choose Watch History from the User Data menu.			Watch History table data is shown with current data present.
8	Visually inspect the table to ensure a series of new simulated users have been generated and all fields are consistent with their description from requirement 3.1.4.1.			All fields and new watch history has been verified.

### 3.4.3 Media Content

<b>Test Category:</b> System Test	<b>Description:</b> This test verifies the Movies, Genre, and Streaming Providers tables are present.		
<b>Test Case:</b> 3.4.3	<b>Case Name:</b> Media Content	<b>Version:</b> 1.0	<b>Written By:</b> James Powers
<b>Requirements Fulfilled:</b> 3.1.4.1	<b>Purpose:</b> To ensure the Movies, Genre, and Streaming Providers tables are present and properly mapped to backend systems.		

<b>Setup Conditions:</b> 3. FlixPicks docker containers are currently running locally for testing. 4. FlixPicks database home page is open (http://localhost:3000)				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Select Tables from the top menu bar.			List of available tables are shown.
2	From the Content menu, choose Movies.			The Movies table is shown with default data present.
3	Visually inspect to ensure the following fields are shown and have contents: Movie_id, tmdb_id, title, release_date, runtime, poster_path, backdrop_path, language			All fields identified have been validated.
3	Navigate to Tables using the top menu bar, then choose Genre from the Content menu.			The Genre table is shown with default data present.
4	Visually inspect to ensure the following fields are shown and have contents: Genre_id, tmdb_id, genre_name			All fields identified have been validated.
5	Navigate to Tables using the top menu bar, then choose Streaming Providers from the Content menu.			The Streaming Providers table is shown with default data present.
6	Visually inspect to ensure the following fields are shown and have contents: Provider_id, tmdb_id, provider_name, display_priority, logo_path, region			All fields identified have been validated.

### 3.6 Analytics

<b>Test Category:</b> Analytics	<b>Description:</b> The test case shall verify the system can increment recommendation numbers.
------------------------------------	--



<b>Test Case:</b> 3.1.6.4		<b>Case Name:</b> Recommendation analytics test	<b>Version:</b> 1.0	<b>Written By:</b> Roxy Egger
<b>Requirements Fulfilled:</b> 3.1.6.4.1 3.1.6.4.2		<b>Purpose:</b> To ensure that the system properly increments and stores recommendation data.		
<b>Setup Conditions:</b> 1. FlixPicks is running 2 The user is logged in 3. The user is watching or has watched a piece of media				
<b>Test Case Activity</b>		<b>Pass/Fail</b>	<b>Comments</b>	<b>Expected Result</b>
1	The user interface section for recommending media appears			The user interface section appears correctly
2	The user recommends a piece of media			The system increments the recommendation count for that media.
3	The recommendation count is incremented for a piece of media			The system stores the new recommendation count in the database.

<b>Test Category:</b> Analytics	<b>Description:</b> This test case shall verify that the feedback form is accessible and running.		
<b>Test Case:</b> 3.1.6.3	<b>Case Name:</b> Feedback form test	<b>Version:</b> 1.0	<b>Written By:</b> Roxy Egger
<b>Requirements Fulfilled:</b> 3.1.6.3.1	<b>Purpose:</b> To test that the feedback form is accessible and saved in the backend when submitted.		
<b>Setup Conditions:</b> <div>1. FlixPicks is running</div> <div>2. The user is logged in</div> <div>3. The user is on the web page</div>			

Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user clicks on the the button for leaving feedback			The feedback form is displayed
2	The user submits the feedback form			The entered feedback is stored in the backend
3	The user exits or clicks off the feedback form			The feedback form closes and the data is discarded

<b>Test Category:</b> Analytics		<b>Description:</b> The test case shall verify that the category rating works properly.		
<b>Test Case:</b> 3.1.6.2		<b>Case Name:</b>	<b>Version:</b>	<b>Written By:</b>
<b>Requirements Fulfilled:</b> 3.1.6.2		<b>Purpose:</b> To test that users can rate categories properly and that the rating information is stored in the backend.		
<b>Setup Conditions:</b> 1. FlixPicks is running 2. The user is logged in 3. The user has finished watching a piece of media				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user interface element for rating categories appears			The user interface element appears correctly
2	The user likes a category			The user rating is stored in the backend

3	The user dislikes a category			The user rating is stored in the backend
4	The user exits or clicks off the category rating element			The element is closed