Lab 2 - FlixPicks Prototype Product Specification

Chance Sadler

Old Dominion University

CS 411W

Professor J. Brunelle

8 April 2024

Version 1

Table of Contents

List of Figures

List of Tables

1. Introduction

In recent years streaming services have overtaken cable television in popularity. Streaming services represent a 34.8 percent of the American viewership compared to cable's 34 percent (Fischer, 2022). 35 percent of Americans are browsing for something to watch A 2016 study revealed that Netflix users spend 17.8 minutes on average searching for something to watch (Moscaritolo, 2016). Many shows are 20 minutes long. That means on average a user misses one episode every time they search for new content. The recommendation system for each service is only based on the user's choices within the app. This can lead to shows the user enjoys not getting recommended because they never chose similar shows on that service.

Many people use multiple streaming services to have more content options. Netflix is just one of over 200 streaming platforms (Cook, 2023). With so many services and shows to choose from the problem of finding content is exacerbated. Browsing by genre in every individual service takes too much time away from consumers. Many consumers pay for multiple streaming services to expand their options or watch exclusive content. While this can be seen as a positive it also adds complexity when the person starts the search for a new show or movie. If a service no longer has content that the user wants to watch it will go unused or canceled. Half of the people who subscribe to multiple services tend to stay subscribed without using the service (Glover, 2023). The price of streaming content varies from 2 dollars with ads to 20 dollars or more per month per service (Clark, 2022). There is a lot of potential for consumers to waste money. The solution to this problem is FlixPicks a cross-platform app that recommends shows across all streaming services while allowing the user to filter by the services they use.

1.1 Purpose

FlixPicks is a desktop web application designed to be an cross streaming platform library and movie recommendation service. The goal is to allow users to find content to watch that is relevant to them more easily. The recommendation system works through tracking user activity through watch history. FlixPicks does not have actual video content on the application. FlixPicks would have to get the rights for each movie if that were the case.

1.2 Scope

The FlixPicks prototype demonstrates the key features on a smaller scale. The prototype has limited movies that it has access to. Users are simulated for testing and demonstration of important features. The simulated users can be accessed and manipulated to see how features interact with users. User activity is tracked, and fake user activity is simulated. The prototype can generate recommendations based on patterns made my multiple user activities.

1.3 Definitions, Acronyms, and Abbreviations

Android: An operating system for mobile devices manufactured by Google, Samsung, and other companies.

Apache Tomcat: An open-source implementation of the Java Servlet, Java Server Pages, Java Expression Language, and WebSocket technologies. Tomcat provides a “pure java” HTTP web server environment in which Java code can run.

API: An Application Programming Interface is an interface that allows for interactions between multiple software applications or mixed hardware-software intermediaries.

CineMap: CineMap is an optional overlay that allows FlixPicks users to interact, make comments, and view other users’ interactions. It shows a timeline that highlights points of interest throughout a movie based on activity.

CineRoll: CineRoll is a FlixPick's feature that generates random selections based on a user's interests. CineRoll uses the Taste Profile to generate selections for a user based on their recommendations.

CineWheel: CineWheel is a FlixPick's feature that randomly selects from a set of user inputted choices. CineWheel is a tool to use for when a user or multiple users in a group are indecisive.

CSS: Cascading Style Sheets is a style sheet language used for customizing the appearance of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Git: Software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

HTML: Hyper Text Markup Language. HTML is the standard markup language for creating web pages. HTML elements tell the browser how to display the content and define the structure of web pages.

IDE: An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development.

iOS: An operating system used for mobile devices manufactured by Apple Inc.

JavaScript: A scripting or programming language that allows you to implement complex features and interactivity on web pages.

MySQL: An open-source relational database management system.

Netflix: A subscription-based streaming service that allows members to watch TV shows and movies on internet-connected devices.

Stakeholders: A person with interest or concern in something, especially a business.

Streaming: A method of transmitting or receiving data over a computer network as a steady, continuous flow, allowing playback to start while the rest of the data is being processed.

Taste Profile: A user profile on FlixPicks that has access to their subscriptions, recommendations, and other settings. The taste profile recommendations grow as a user makes selections on the website and can be reset at anytime by the user.

1.4 References

Clark, Travis. (2022, April 20). How Netflix, Disney+, HBO Max, and more major streamers compare on content and cost. https://www.businessinsider.com/major-streaming-services-compared-cost-number-of-movies-and-shows-2022-4

Cook, Sam. (2023, September 06). The Complete List of Streaming Services – 200+ Services. https://flixed.io/us/en/complete-list-streaming-services

Durrani, Ana. (2023, March 27). The Average American Spends Over 13 Hours A Day Using Digital Media—Here's What They're Streaming. https://www.forbes.com/home-improvement/internet/streaming-stats/

Fischer, Sara. (2022, August 18). Streaming surpasses cable as top way to consume TV. https://www.axios.com/2022/08/18/streaming-surpasses-cable-tv-market-share

Glover, Emily. (2023, March 9). Nearly 50% of people pay for streaming services they don't use. According to new Forbes survey. https://www.forbes.com/home-improvement/internet/streaming-survey/#:~:text=According%20to%20the%20survey %20findings,services%20the%20average%20person%20uses.

Moscaritolo, A. (2016, July 21). Netflix users waste a ton of time searching for something to watch. PCMAG. https://www.pcmag.com/news/netflix-users-waste-ton-of-time-searching-for-something-to-watch

Natal, G., & Saltzman, B. (2022, January) Decisions, decisions, decisions: decision fatigue in academic librarianship. The Journal of Academic Librarianship, 48(1) https://doi.org/10.1016/j.acalib.2021.102476

O'Brien, Clodagh. (2023, June 14). The Unstoppable Success of Netflix. https://digitalmarketinginstitute.com/blog/the-unstoppable-success-of-netflix#:~:text=Netflix's%20marketing%20budget%20has%20remained,to%20the%20New%20York%20Times.

Pattison, S. (2023, September 17). 35 streaming services statistics you need to know in 2023. Cloudwards. https://www.cloudwards.net/streaming-services-statistics/#:~:text=Although%20we%20may%20only%20think,Netflix%20the%20%231%20Streaming%20Service%3F

Sadler, N. C. (2024, April #). Lab 1 Final Version Old Dominion University. https://www.cs.odu.edu/~411orang/

1.5 Overview

The main purpose of FlixPicks is to allow users to view all their available content in one place. Additionally, the app can show the user what other services offer if they want to explore their options. FlixPicks allows users to leave comments and interact with specific media. FlixPicks keeps track of user's content habits such as shows and movies they interact with and

user watch history. The information collected allows FlixPicks to generate recommendations for new content on the services they own. Additionally, FlixPicks will recommend a few options from services they are not subscribed to.

2. General Description

The FlixPicks prototype includes much of the functionality of the real-world version. The prototype handles a smaller sample size of subscription services. The FlixPicks prototype is designed to demonstrate the key features of the final product. Simulation is used to show different types of users interacting with app for testing and demonstrating proof of concept.

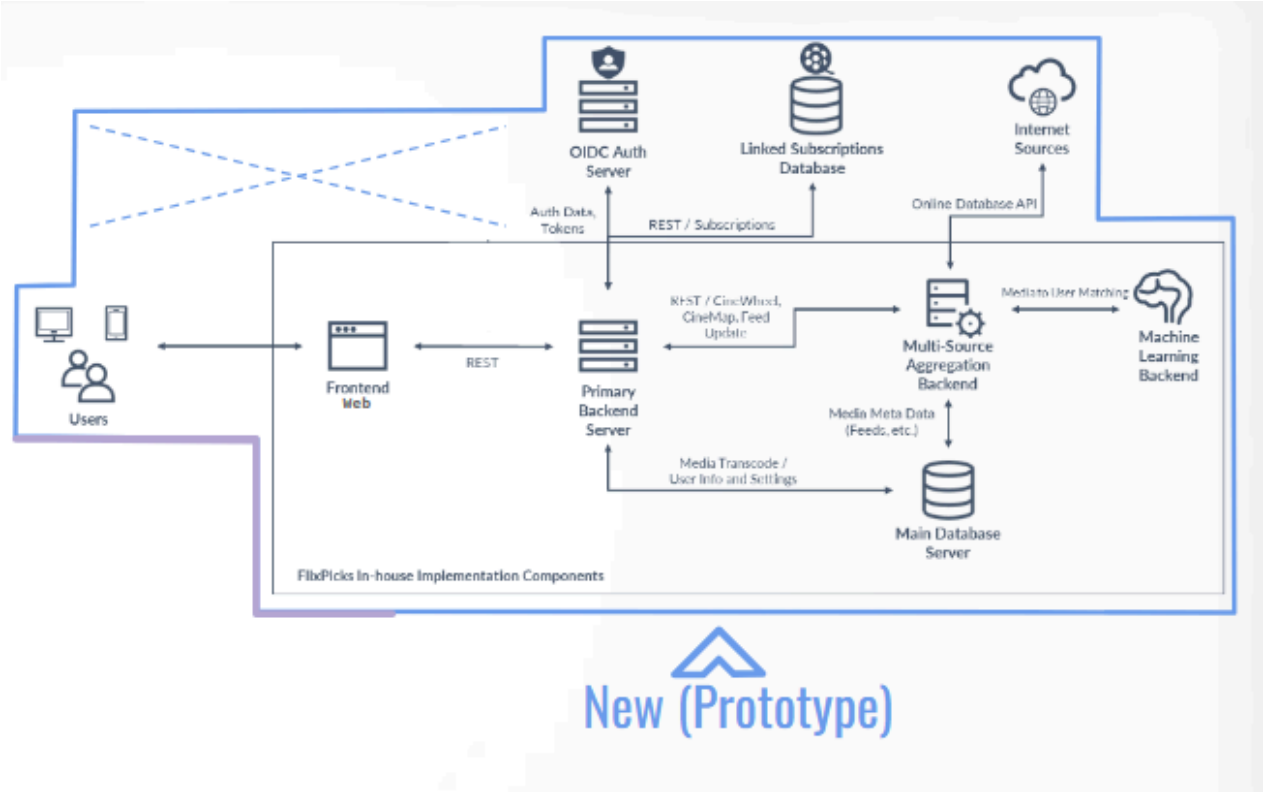
2.1. Prototype Architecture Description

The prototype components are stored on the Old Dominion University servers. A virtual machine runs the backend using a Docker container with Apache, SQLite, and Python. The prototype works on any device that can access web pages and use chrome extensions. Omitted from the prototype are the phone and tv interfaces that the real-world product does. The user interface allows users to interact with the backend through a webpage. The front-end uses a combination of HTML,CSS, and JavaScript so users can visually interact with the app. Project management and version control are handled through Git and GitHub.

Figure 2 illustrates the major functional component diagram (MFCD) for the FlixPicks prototype. The main difference between the real world MFCD and prototype MFCD is that the prototype does not use a library API.

Figure 1

FlixPicks Major Functional Component Diagram



2.2 Prototype Functional Description

The key real world product features of FlixPicks are included in the prototype. The key features being Cineroll, Cinewheel, Hotpicks, and the recommendation system. Additionally, to demonstrate the key features user accounts, a movie library, and analytics are partially implemented. The prototype demonstrates the end solution of lowering the time to find a movie across multiple streaming services based on user watch history. The recommendation algorithm uses users' watch history and cross references it with other users to recommend movies. Cineroll and Cinewheel are both fully functional, but the database of movies is limited. Other users are faked for the purposes of demonstration. Table 1 describes the features that are included in the FlixPicks prototype.

Table 1

FlixPicks Prototype Features Table

Category	FlixPicks Feature	RWP	Prototype
Subscription Service Management	Account/Subscription Service Management	Fully Implemented	Fully Implemented
	User Account Creation/Registration	Fully Implemented	Fully Implemented
	User Subscription Integration	Fully Implemented	Partially Implemented
	User Tier Level Feature Access	Fully Implemented	Partially Implemented
Taste Profile	Taste Profile	Fully Implemented	Fully Implemented
	Taste Profile Form Pop-Up	Fully Implemented	Fully Implemented
	Taste Profile Content-Based Filtering	Fully Implemented	Fully Implemented
	Taste Profile Collaborative Filtering	Fully Implemented	Fully Implemented
Recommendations	Recommendations	Fully Implemented	Fully Implemented
	Filtered Recommendations (Criteria based)	Fully Implemented	Fully Implemented
Filtering	Browse/Search Filtering	Fully Implemented	Fully Implemented
CineRoll	CineRoll	Fully Implemented	Fully Implemented
CineWheel	CineWheel	Fully Implemented	Fully Implemented
CineMap	CineMap Overlay	Fully Implemented	Fully Implemented
	CineMap Commenting	Fully Implemented	Fully Implemented
	CineMap Export Data	Fully Implemented	Partially Implemented
	CineMap Data Analyzing	Fully Implemented	Partially Implemented
Analytics	Data analytics testing	Fully Implemented	Partially Implemented
	Analytics	Fully Implemented	Partially Implemented
Simulation	Simulation	Eliminated	Fully Implemented
Movie Info	Create/edit Movie Info	Eliminated	Fully Implemented

Reporting	Summary reporting for user/stakeholders	Eliminated	Fully Implemented
Feedback	Feedback	Fully Implemented	Partially Implemented

FlixsPicks is a user driven application and for that reason users are simulated in the prototype. The database of movies is limited due to lack of available web API interface at the time of development.

2.3 External Interfaces

The FlixPicks prototype runs on a host server and is accessed by users on their computer. APIs are utilized in order to access movie data.

2.3.1 Hardware Interfaces

The prototype requires a computer running on any OS that supports a chromium web browser. A mobile and TV applications are not part of the prototype. A mobile device using a web browser may still be able to access the prototype, but the formatting of the UI for mobile devices is not within the scope of the project.

2.3.2 Software Interfaces

The prototype application is a mix between Java Script and Python. The user interface is handled through html pages generated using templates. The content on the html templates are generated using a python script along with the backend database. Animations are created using Java Script and Cascading Style Sheets. Docker is utilized to build and run the prototype application. An external API called TMDB is used to add movie information to the backend database.

2.3.3 User Interfaces

The FlixPicks prototype includes user interfaces for account management, browsing and searching, Cinewheel, Cineroll, and the Media Info page. The Cinewheel feature allows for users

to submit a list of movies into it and spin the wheel. The Cineroll button allows users to click to a random movie from their recommendations.

2.3.4 Communications Protocols and Interfaces

The frontend and backend communicate to display the content onto the html templates. The backend requests movie information using TMDB to populate the media library. Docker connects the separate containers for frontend, backend, and recommender. The Quick Click link redirects to the IMBD page where users can select their streaming service.