Android App Development for Amazon Fire TV

CS4991 Capstone Report, 2024

Spencer Jackson
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
slj5hwr@virginia.edu

ABSTRACT

IMDb.com, an online film and television database company, required a new feature to be developed for its Amazon Fire TV application, to allow users to view featured video content for trending or popular celebrities. To create this feature, I utilized the Android SDK to create a Netflix-style carousel row in the application, which showcased curated lists of popular film-related celebrities along with featured videos. I created a technical design doc for review by my team, and created the feature using Kotlin and the standard Android architecture guidelines, along with some custom IMDb components. I also created unit tests for all code written, as well as created and executed a UI testing plan. After the initial testing was completed, the feature was merged into a UAT build, before eventually being fully released. Future work is required to add features such as dynamic lists and Amazon Alexa support, as well as improved video performance.

1. INTRODUCTION

Since the 1950s, the television set has been the center of entertainment for nearly every modern household in America. However, the way this technology is used has changed significantly in recent years: rather than flipping through cable channels to watch something that is currently airing, we instead have hundreds of thousands of TV shows and movies available to us on-demand through streaming services. This has had a significant impact on the entertainment experience: rather than a single exhaustive list of content, we now do our consumption through various different services, owned by different companies, offering different content. This creates a frustrating experience, where you can potentially spend more time searching than actually watching, due to the sheer number of possible decisions.

The IMDB.com What to Watch App is designed to eliminate this annoying user experience, by providing users with a one-stop-shop for finding new entertainment content. Users can play short "games" to get recommendations, or view collections of trending and popular shows and movies from different genres, all while the application tailors the results for the streaming services you subscribe to. To give users an even wider breadth of new content suggestions, I created a feature for the application which displays curated collections of trending celebrities (actors, directors, etc.), along with featured short-form videos or trailers to showcase their newest projects.

2. RELATED WORKS

Cohen (2022) describes how, according to a survey by IPSOS and NPR, "69% of respondents think there are too many streaming services", and "58% of streamers feel overwhelmed by the sheer amount of content being pitched to them". My team's application and my feature are designed to eliminate this overwhelming feeling by narrowing down streaming options across services.

According to Hinkle (2021), 80% of Netflix's minutes streamed come from users viewing titles suggested by their recommender system, and other popular platforms like Hulu and Disney+ use machine learning models to curate recommendations for users. While this suggests that existing recommendation algorithms for specific individual streaming platforms are effective, they merely provide an intra-platform solution. When navigating between a half-dozen streaming platforms, users can still become overwhelmed, especially when not all recommendation systems are built equally.

3. PROJECT DESIGN

My project for the internship focused on the creation of a new feature called the names widget, which I developed based on requirements given to me by my team's project manager.

3.1 Overview

The What to Watch app features a main view made up of various rows of items in a "carousel" format, with each row being a custom component called a widget. Each widget features a different set of items based on a certain theme or criteria, e.x. "Popular Superhero Movies" or "Trending TV Shows in the US". The existing widgets all contained either movies or TV series, these are called "titles widgets". The project described here involved the creation of a new widget type, the "names widget", which contains people rather than movies or series (e.g., "10 Oscar Nominated Actors"). An example names widget row is shown in Figure 1. The names widget itself loosely follows the standard Android app architecture guidelines.

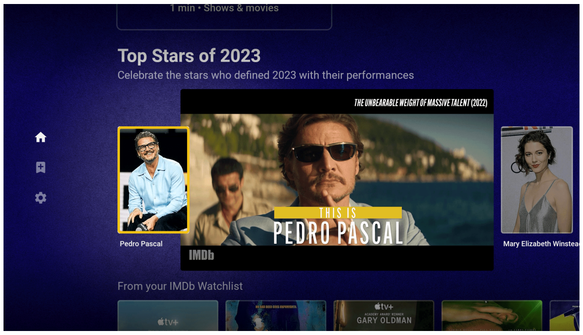

Figure 1: Names widget row example

3.2 Requirements

Each item in the names widget row must contain: 1) a featured photo, ideally of the featured person's headshot, 2) the person's name; and 3) a featured promotional video, oftentimes an interview or movie trailer. The items to be displayed in the row must be specified using a configuration file in JSON format. The user must be able to scroll between items, and whenever an item is focused its associated promotional video must play automatically, with the ability to disable this in the settings menu. The user must also be able to change the video to full screen using the play/pause button on their remote, and the user must be able to mute all audio in the settings menu.

3.3 Implementation

All the data being used for the names widget was retrieved directly from the IMDb internal database using GraphQL endpoints. A new GraphQL query was created, which used the widget configuration file to retrieve all the necessary information for each item in the widget, including the link to the primary photo, person's name, promotional video link, and other metadata (the maturity rating for the video, the video quality, etc.). This query was called in a repository class, as per Android app architecture guidelines.

This data is then passed to the user interface layer through a controller class, which is where all the business logic is contained, including performing all the necessary validation and formatting, placing all the data into an easily accessible object, and determining which item in the row (and therefore photo and video) should be displayed at any given time. Finally, this was all displayed to the view using a view holder class which bound all the data retrieved from the controller using a custom UI component.

3.4 Challenges

Because existing rows for movie and TV series titles were already implemented, they were used as a reference, making the process for this project relatively streamlined, albeit with some small challenges. One such challenge was integrating the new feature piece-by-piece without breaking the current build of the app as a whole. Because this was one of the first major features to be released for the app that required many new classes and changes to existing classes, we had no system in place for its integration, which caused issues when trying to release new builds while the feature was still in development. These issues were solved by adding a special case in the main view class to not attempt to render any rows that were of the new widget type, until the feature was completed. Another issue was that rather than being a fixed size like movie or TV show posters, the headshots that were used for individual items in the row were not a fixed aspect ratio. This caused some pictures to be blurry or incorrectly cropped, while others looked fine. This was solved by creating a new scaling function to ensure that all the photos were scaled correctly.

4. RESULTS

My team and I completed UI testing for my feature in the last weeks of my internship, and it was released in a UAT build of the app after passing UI tests for all Amazon FireTV devices. The feature was ultimately released into production and is available on the IMDb What to Watch app for all FireTV devices; however, as my internship ended prior to the production release of my feature, I was unable to see the expected results of the feature such as increased user interaction and user retention. However, as a result of my work on the application, notably the refactoring of the existing class systems to be more flexible with different forms of data and a more extensive set of unit tests for certain reusable components, my team testified that my work had made the codebase much more maintainable and future proofed.

5. CONCLUSION

The names widget feature for the IMDb What to Watch app provides users with a novel, effective way of discovering new media during a time in which finding what to watch is harder than ever. As the application grows in scope and popularity, it is my hope that this feature will help many IMDb users discover their new favorite actor or director. During the creation of this feature, I learned a great deal about Agile methodology, Android architecture, and overall development on a recently released and quickly-evolving project.

6. FUTURE WORK

One potential feature to be added to the names widget is Amazon Alexa support, which would allow Alexa Show devices to display names widget lists. Another feature would be dynamic list support, in which a given user’s lists could be generated through their watch preferences, or watchlist as opposed to being curated and used for all users. Finally, Roku support would allow the What to Watch app to reach a much wider audience, although due to the different operating systems this would require a great deal of developer effort, as Roku OS is not Android based.

REFERENCES

Amt-Lab. (2021). *How Streaming Services Use Algorithms—AMT Lab @ CMU*. Retrieved February 23, 2024, from https://amt-lab.org/blog/2021/8/algorithms-in-streaming-services

PCMag. (n.d.) *Welcome Back, Cable TV: Americans Think There Are Too Many Streaming Services.* Retrieved February 23, 2024, from https://www.pcmag.com/news/welcome-back-cable-tv-americans-think-there-are-too-many-streaming-services