Luke Hebert
Dr. Andrew Chen
CSIS 492
4 May 2021

Media Backlog Management Application

Introduction

The goal of this project was to create a desktop GUI (Graphical User Interface) application that would manage a user's media backlog. To do this, the user provides the amount of time they want to spend watching either a show or movie and fills in other optional criteria, such as genre, director, actors, and more. From there, the app returns recommendations based on the search criteria. The backlog of items is provided by the user via a simple search-by-title functionality. This search queries IMDb, a website dedicated to aggregating information about film, TV series, and other visual media, and returns a list of results. The user is able to see each result's title, year released, and poster, which they can then select with a simple click. A variety of information about the title is then added to the user's personal database, which can then be used later to when trying to find something to watch. For example, say a user has 25 minutes of free time during their lunch break and wants to watch something funny to fill their free time. They would enter their 20-minute time limit and that they want to watch a comedy, and the app would suggest something from their backlog.

Initial Idea and Tool Selection

The idea for the project came from the many, many film suggestions my friends make to each other. Keeping track of these endless suggestions is a near impossible feat, made even more difficult by the diminishing amount of free time I've had this past year. I've also had many experiences similar to the aforementioned example of having an odd amount of free time and nothing to do in that time. The app seemed to be a perfect way of addressing those two issues and, in addition, could also address the problem of "doom scrolling*" (see foot note) through streaming apps without finding anything to watch by making suggestions for shows that you may actually be interested in. To sum up the purpose of the app in a single phrase: To optimize content consumption based on the ever-changing balance between work and play.

With the initial details in mind, the first step of my development process involved selecting the optimal tools – optimal in terms of pure functionality, but also optimal in that they would suit my skillset at the time while still allowing me to expand my skills. This led me to select C# as my primary programming language. It was a language I had experience with due to taking CSIS 336 (C# Programming) at MSUM and a language which I'll be using in an upcoming internship. Of course, C# can't be discussed without mentioning Visual Studio, one of the most comprehensive and dynamic development environments in its class. I decided to use Visual Studio's Windows Forms, a graphical class library, to construct the GUI elements of the app, as it was similar to prior tools I had worked with in CSIS 335 (GUI Programming). With that, I was left with a solid foundation to build from.

The next step in tool selection was finding a method of retrieving data about video media. I knew IMDb.com was far and above the most expansive source of that data, but being able to access that data in a quick and efficient manner was a different story. After research, the best candidate was the "Movie Database (IMDb Alternative)" API. Despite its title of being an "alternative", it still accesses IMDb data. It provided a method to search for a movie by title, which returned multiple results, and a method to get more in-depth information on each movie via searching by its IMDb ID. Finally, to keep the data

* slang - the act of mindlessly scrolling through a site/app with no direction, often wasting significant amounts of time

persistent between sessions, I used MSUM's MySQL database on the Puff server since I'll be working with MySQL and other Oracle tools this summer.

After selecting the tools I would use, I decided to plan out a development process that used sprints, a concept used by the Agile programming methodology. Each sprint consisted of a list of tasks to be completed in a time frame. The first three focused on a crucial element of the app, with the first dealing with a prototype of the GUI, the second dealing with the IMDb API, and the third dealing with the MySQL database connection. From there, the remaining sprints involved combining everything and fine tuning it from there.

Design Philosophy and Inspiration

The beginning stages of this project involved the gathering of a wealth of sources and references. I had assumed these sources would be instrumental in the design of my app. However, the reality was that only a few were actually used. The most influential was Wilbert Galitz's *The Essential Guide to User Interface Design: an Introduction to GUI Design Principals and Techniques*. In particular, his coverage of the Human Action Cycle's role in interaction with GUIs (Galitz 72). The process could be summed up as follows:

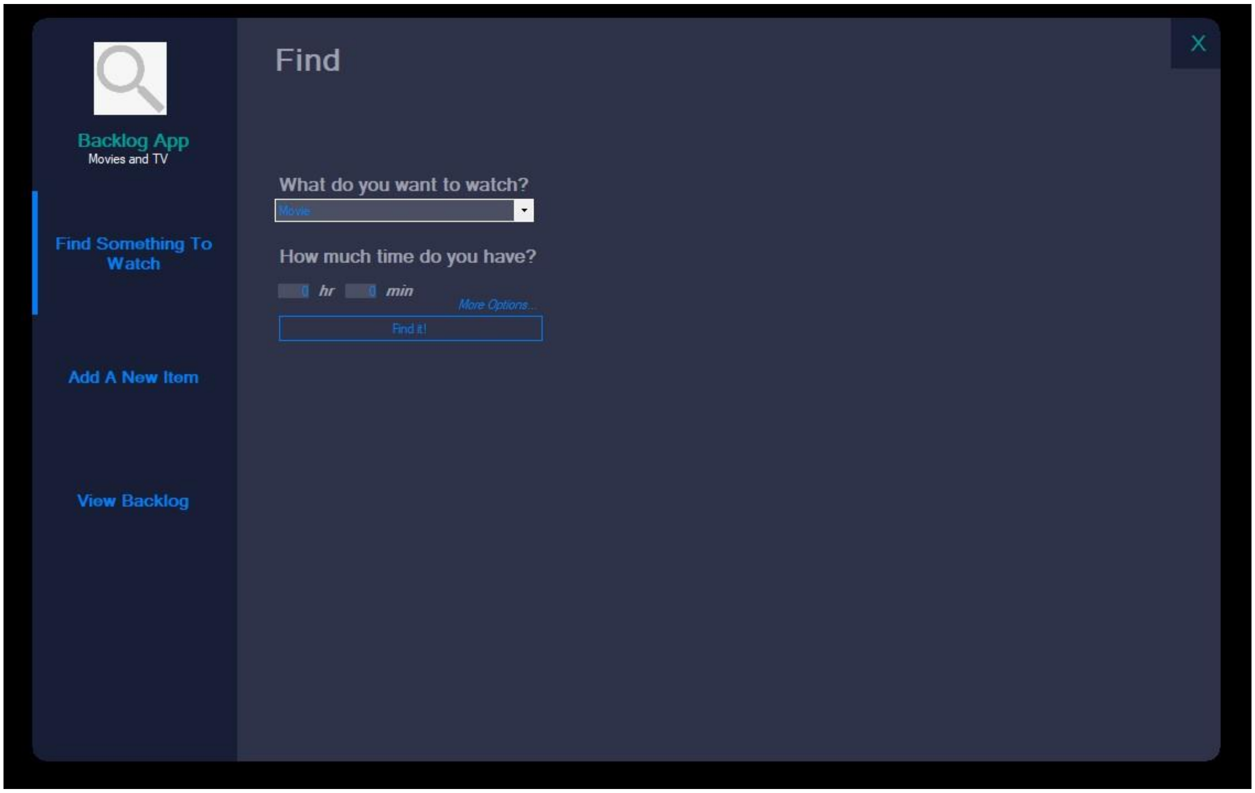
1. A user sets out with a goal in mind and formulates a plan to achieve said goal
2. The user executes their plan
3. The user then evaluates the output of their actions

To me, there were two major takeaways from Galitz's explanation of the cycle: the app would need to be easy to navigate and would have to make user's actions feel responsive. For this app be easy to navigate and for it to have any practical use, the time to get from point A to point B would need to be streamlined as much as possible without losing functionality. The word that stuck with me throughout development was "lightweight", very much like a mobile app. It would need to be easy to pick up, get what was needed (a suggestion for what to watch or adding a new item), and drop. In addition to that, each interaction a user would have with an element would need to provide feedback that was also quick. This ranged from subtle details like buttons changing color when the mouse was over them to displaying messages letting the user know their desired action had been completed.

Throughout the development of the app, there were two references I would often make. The first was to the IMDb mobile app. It's something I use somewhat frequently to get information about movies and it's an experience I would consider "lightweight" in that it's quick and easy to use. However, it still provides a wealth of information about films, TV shows, actors, and more. When using it, I'd often finding myself thinking about elements of the app and how many steps it took to complete my task. On the other hand, I also have experience with an app at my former job doing data entry. It was an app built for entering and viewing data about properties. Despite doing it's job very well, it's the exact opposite of what I hand in mind for my app. It required long-term engagement from the user in the form of entering items for nearly every field. This isn't to say that app was poorly designed – far from that, in fact. They simply serve different purposes. While working on my app, I would always try to take a step back and consider where it was in relation to the IMDb app and the app I used to use at work. From there, I would adjust my strategy and proceed.

The final product was an app that performed as I had specified in my initial design document. The basic search screen allowed the user to enter their free time and what they wanted to watch. Then, based on the user's criteria, the results would be returned See below for some screenshots of the app. An expanded search criteria could also be used, which allowed for searching by year, director, actor (anyone in the main cast), genre, and minimum score from either IMDb's rating system or Metacritic's. Items are added to the backlog via the "Add a New Item" page. Here, users type the title of the item they want to add and results are returned. Each item has its poster displayed when the mouse hovers over it. To add to the backlog, users simply double click the item.

Search screen – standard view

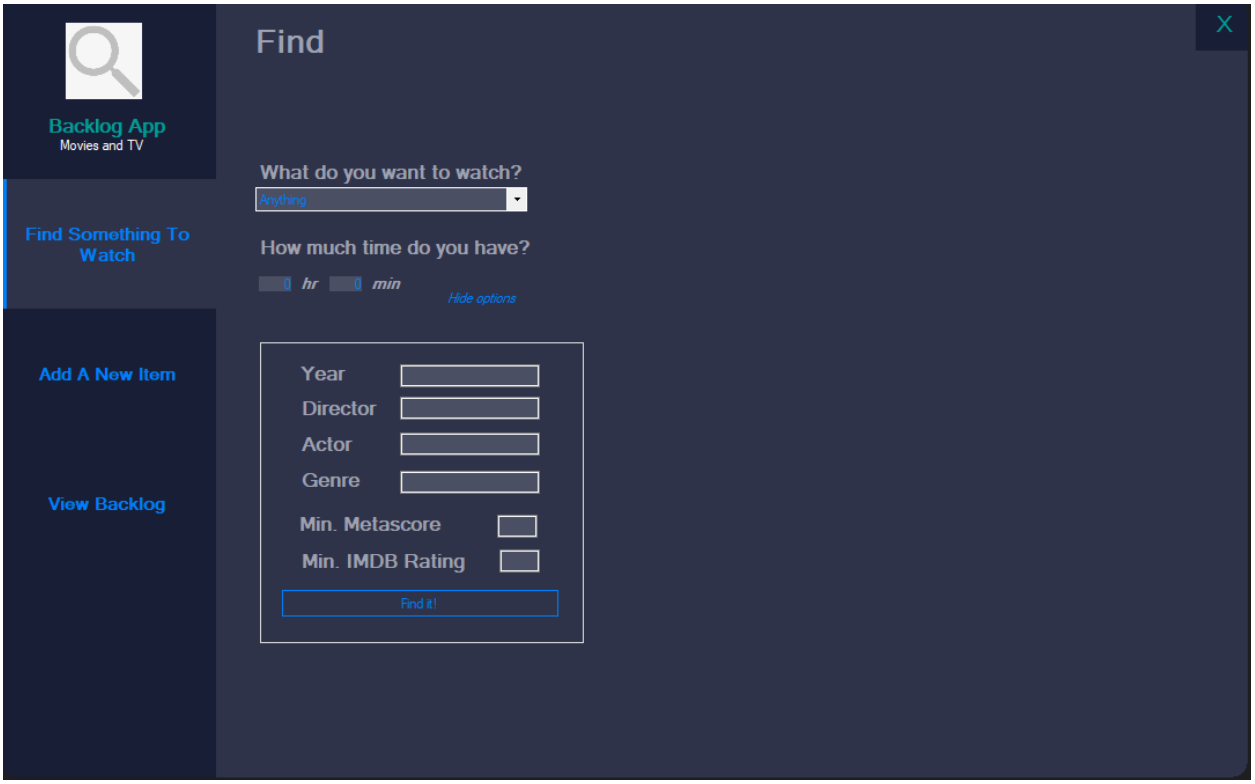


Find

What do you want to watch?
Movie

How much time do you have?
0 hr 0 min

Search Screen – Expanded Criteria



Find

What do you want to watch? Anything

How much time do you have? 0 hr 0 min

Year

Director

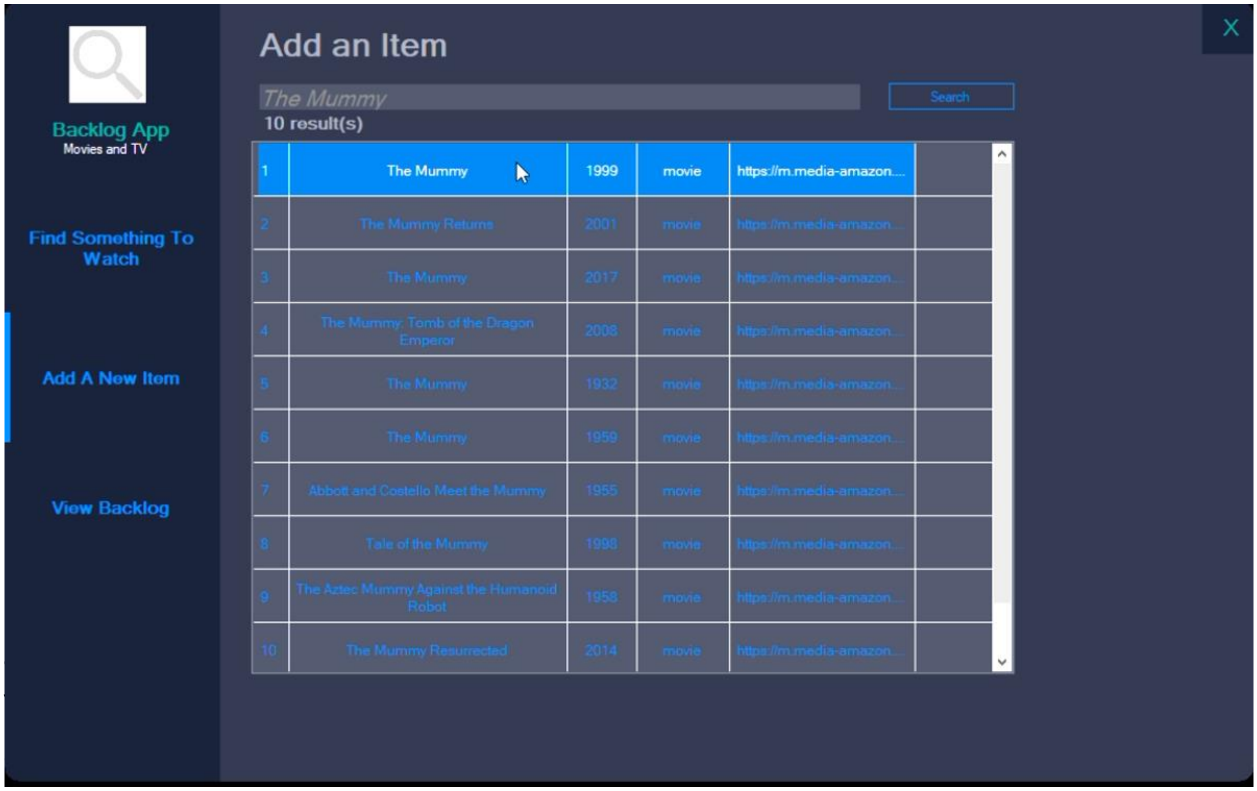
Actor

Genre

Min. Metascore

Min. IMDB Rating

Add a New Item Screen – Note: unable to capture the poster being displayed



Add an Item

The Mummy

10 result(s)

1	The Mummy	1999	movie	https://m.media-amazon....
2	The Mummy Returns	2001	movie	https://m.media-amazon....
3	The Mummy	2017	movie	https://m.media-amazon....
4	The Mummy: Tomb of the Dragon Emperor	2008	movie	https://m.media-amazon....
5	The Mummy	1932	movie	https://m.media-amazon....
6	The Mummy	1959	movie	https://m.media-amazon....
7	Abbott and Costello Meet the Mummy	1955	movie	https://m.media-amazon....
8	Tale of the Mummy	1998	movie	https://m.media-amazon....
9	The Aztec Mummy Against the Humanoid Robot	1958	movie	https://m.media-amazon....
10	The Mummy Resurrected	2014	movie	https://m.media-amazon....

Cuts, Contemplation, and Conclusion

Early in development, it was clear I needed to narrow the focus of the project. For this app to fit my idea of “lightweight”, it could only focus on video media. I had intended for it also to support books and video games. However, due to the nature of those mediums not having a clear cut runtime, they simply wouldn’t fit and were cut from the project. I feel they would require their own, separate app to be effectively managed and my goal for the app was a more seamless experience. Going back to the idea of comparing the app with the IMDb mobile app and the property app, if I had focused on more than video media would make it fall more in line with the property app. Again, the property app is not poorly designed, but is simply not in line with the idea of a “lightweight” experience. Another shortcoming with my development was a lack of testing. This was due to exercising caution with regard to COVID. I think if I was able to test the app with other people and receive their input, the app could have been improved.

In its current state, the app was built to the spec I had initially set out to fulfill, with the exception of the book and game portion. However, I do think there is room to grow for this app. This is somewhat intentional, as there were some concepts I wasn’t comfortable with using at the time of development but would still like to learn. One goal I have is to make a version using Xamarin, an app platform which allows for the creation of mobile apps with C#. Given my comparison to mobile apps throughout development, I feel like that would be a logical next step. Of course, there are also improvements that could be made with the current version. One idea was to create a more intricate algorithm for the order in which results from the backlog are returned based on certain criteria. Another was to create a sign-in system for different users and a user rating system to hone in on what a user prefers to watch and if anything in their backlog is similar. If the app were to be released as a commercial product, it would make sense to add gamified elements, such as achievements or a leveling system, to encourage repeat user engagement. I’d also like to make it into a single executable program down the line, making it more like an “end-product” of a traditional development cycle.

I feel I was able to exercise a majority of the knowledge I’ve accumulated over my past two years at MSUM. I developed an app from initial idea to a working prototype, with an emphasis on creating a user experience that is as lightweight and functional as possible. Though development wasn’t always smooth and there are certainly areas to improve, this has been a project I’ve wanted to make for a while and it’s something I’ll build on in the future for my own personal project.

Bibliography

Galitz, Wilbert O. *The Essential Guide to User Interface Design: an Introduction to GUI Design Principles and Techniques*. John Wiley, 2007.

Relevant Links

API used:

https://rapidapi.com/rapidapi/api/movie-database-imdb-alternative

Slides and video demo from Student Academic Conference:

https://www.slideshare.net/LukeHebert7/sac-slides-lukehebert/LukeHebert7/sac-slides-lukehebert