Movie Interface

Seth McCrudden and Jimmy Butcher

Abstract

This project proposal presents the requirements and the design for planning a movie interface that allows the user to store and access information about the user's movie collection that is stored in a database using a system that provides easy to understand and user friendly features for performing searches of the movie collection and for entering information for movies to add to the collection.

1. Introduction

The movie interface is designed to give the user a fast, convenient, and easy way to search their movie database at any time and to add information for new movies in the user's collection.

The screen interface gives the user the ability to do searches based upon the title, the actors, the director, the release date, the time length, the rating, the production company, and the genre and to print the information for any movie or list of movies.

This interface will provide a straight-forward, attractive visual screen presentation with the goal of giving the user a simple-to-understand interface to provide a way to manage a movie collection database.

1.1. Background

The system will be implemented as a Windows Presentation Foundation (WPF) (.NET Framework) console application using the C# programming language. The design of the movie database interface will provide easy access to movie information that will function through a user interface (UI) screen.

1.2. Impacts

A well thought out movie database interface system can offer a means of allowing the user to create searches based on the title, the actors, the director, the release date, the time length, the rating, the production company, and the genre.

1.3. Challenges

How to best implement allowing a user to do a search using multiple filters. How to implement having search suggestions pop-up when a user begins to type in a search.

2. Scope

The movie database interface has an attractive, easy to understand appearance that provides a simple and straightforward screen clearly showing the available search and entry options. The system has a way to enter movie information into a database to store a collection of movies. The database will store the title, the actors, the director, the release date, the time length, the rating, the production company, and the genre. It also allows the user to perform a search for movies based upon any of these attributes.

2.1. Requirements

The movie database interface should have an aesthetically appealing screen interface that is simple, easy to understand, and provides the following capabilities and features: The ability to input the information pertaining to movies contained in a user's collection. The capabilities to search for movies based upon particular search criteria.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Enter movie information in collection database	User	Med	1
2	Search database for movie by title	User	Med	1
3	Search database for movies by actor	User	Med	1

TABLE 1. USE CASE TABLE

2.1.1. Functional.

- System will allow a user to enter the title of a movie to store in the collection database.
- System will allow a user to enter the actors in a movie to store in the collection database.
- System will allow a user to enter the director of a movie to store in the collection database.
- System will allow a user to enter the release date of a movie to store in the collection database.
- System will allow a user to enter the time length of a movie to store in the collection database.
- System will allow a user to enter the rating of a movie to store in the collection database.
- System will allow a user to enter the production company of a movie to store in the collection database.
- System will allow a user to enter the genre of a movie to store in the collection database.
- System will allow a user to search the collection database for a movie by title.
- System will allow a user to search a collection database for movies by actor.
- System will allow a user to search a collection database for movies by director.
- System will allow a user to search a collection database for movies by release date.
- System will allow a user to search a collection database for movies by time length.
- System will allow a user to search a collection database for movies by rating.
- System will allow a user to search a collection database for movies by time production company.
- System will allow a user to search a collection database for movies by genre.

2.1.2. Non-Functional.

- User interface will be easy to understand.
- User interface will be attractive.
- User interface will be well organized.
- User interface will not be too busy.
- System will be testable.
- System will be reliable.

2.2. Use Cases

Use Case Number: 1
Use Case Name: Enter movie information
Description: A user enters information for a movie in the user's collection.

1) User enters the movie title.
2) User enters the movie actors.
3) User enters the movie director.
4) User enters the movie release date.
5) User enters the movie time length.
6) User enters the movie rating
7) User enters the movie production company.

Termination Outcome: The user has entered the information of a movie in the user's collection into the database.

Use Case Number: 2
Use Case Name: Search for a movie by title
Description: A user searches for the movie collection for a particular title.

1) User enters a movie title in the search box.
2) User clicks the search button.

Termination Outcome: The user views the information returned from the database for the movie title that was searched for.

Use Case Number: 3
Use Case Name: Search for movies by actor
Description: A user searches for movies by actor.

1) User enters the name of an actor in the search box.
2) User clicks the search button.

Termination Outcome: The user views the movies starring a particular actor.

2.3. Interface Mockups

Figure 1. Movie Interface: Enter Movie Information

Figure 2. Movie Interface: Search Title

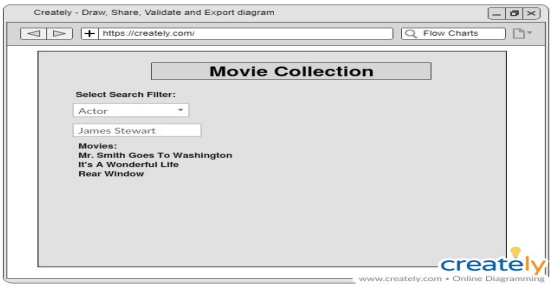

Figure 3. Movie Interface: Search Actor

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure **??**.

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?