Chance Gray

Carsten Hood

Connor Fear

Matt Hegarty

Changes to Original Design

* Our graphical display has the same design and concept as we first outlined, but with more features. We found it practical and not significantly more difficult to display all of a movie’s information beneath its poster, rather than just the poster and title. Our display is also able to show 1, 4, 9, or 16 movies in a square grid taking up the right half of the screen. The user is able to choose how many movies they want to see (by updating the numbers of rows) and navigate through them with a previous and next button appearing when there are multiple pages of movies. The total number of resulting movies from the search, the current page, and the number of total pages are displayed in boxes at the bottom of the screen.
* FLTK did not allow us to auto-fill in boxes so we could not show information how we originally planned; we are now just displaying it all and having the user to enter in the information themselves for editing. The search by actor button now not only displays a picture of the actor, but also lists all the movies they are in. When the edit button is selected, all the current buttons on the screen are hidden and a new menu appears that prompts the user for editing. When update is pushed in this “edit mode,” the new information is saved and stored. When done is pushed in edit mode, all the current buttons are hidden and we are taken back to our original screen.
* We also allow the user to search by each piece of information in the database specifically, such as a search by title, search by year, search by actor, and search by tags. There is a “General Search” that will search for everything that has the text boxes filled out, and then display the results
* There is a display all button that displays every movie in the current database. The clear display will clear the right side of the screen regardless of the number of movies showing.
* A new struct called Movie\_display was added to the program. It contains pointers to all the Text and Image objects we are displaying on the screen for a single movie. We realized the need for this struct in order to keep track of all of the movie’s poster and information so that it can be removed when the search results are no longer being displayed.
* In the Actor struct we combined the two strings for an actor’s first and last names into a vector of names to allow for more flexibility with the number of names an actor can have. This also increases the power of our search function over having a single string, allowing the user to search for an actor’s first, middle or last name, rather than necessitating a full name.
* Our project is separated into three files: Data.h has movie and actor structs along with the read and write functions, Search.h has search() and specific search functions such as search\_by\_actor(), Main.cpp has the majority of the code with the Movie\_display struct along with the Movie\_window class and definitons.