Rapid Application Development

Project

AT2

|  |  |
| --- | --- |
| Team B |  |
| Zara Duncanson | P229768 |
| David Perry | 30010019 |
| Alan Pedersen | P225139 |
|  |  |

Date: 21/05/2020

Contents

[Table of Figures ii](#_Toc42084758)

[Table of Tables ii](#_Toc42084759)

[Analysis and Setup 1](#_Toc42084760)

[Meeting Agenda and Minutes 1](#_Toc42084761)

[Introduction 1](#_Toc42084762)

[Source Control 5](#_Toc42084763)

[Project Management 5](#_Toc42084764)

[Software Development Test Plan 2](#_Toc42084765)

[Introduction 2](#_Toc42084766)

[Test Methodology 2](#_Toc42084767)

[Test Deliverables 3](#_Toc42084768)

[Resource and Environment Needs 3](#_Toc42084769)

[Analysis Documentation 2](#_Toc42084770)

[CITE Business Rules for Software Development 2](#_Toc42084771)

[CITE Coding Standards 3](#_Toc42084772)

[CITE Quality Assurance 3](#_Toc42084773)

[ACME Entertainment Pty Ltd Requirements 4](#_Toc42084774)

[Sprint 1 2](#_Toc42084775)

[Sprint 1 project management plan 2](#_Toc42084776)

[Multi-Platform Report 3](#_Toc42084777)

[Adaptive 3](#_Toc42084778)

[Responsive 3](#_Toc42084779)

[Analysis 4](#_Toc42084780)

[Choice 4](#_Toc42084781)

[References 5](#_Toc42084782)

[Development and Testing 1](#_Toc42084783)

[Code Review 1](#_Toc42084784)

[Functional Testing 2](#_Toc42084785)

[Adaptive Web Design 7](#_Toc42084786)

[Sprint 1 Review Meeting 12](#_Toc42084787)

# Table of Figures

[Figure 1: GitHub repository snapshot 5](#_Toc42079862)

[Figure 2: sprint 1 Gantt chart 2](#_Toc42079863)

[Figure 3: sprint 1 project plan 2](#_Toc42079864)

[Figure 4: The difference between Adaptive and Responsive web design (Merlin, 2018). 3](#_Toc42079865)

[Figure 5: the search page 2](#_Toc42079866)

[Figure 6: searching by title 2](#_Toc42079867)

[Figure 7: searching by title and year from 3](#_Toc42079868)

[Figure 8: searching by title and year to 3](#_Toc42079869)

[Figure 9: searching by title, year from and year to 4](#_Toc42079870)

[Figure 10: searching by Title and date ranges 4](#_Toc42079871)

[Figure 11: including Genre values in the search 5](#_Toc42079872)

[Figure 12: searching using all constraints 5](#_Toc42079873)

[Figure 13: The Top 10 page 6](#_Toc42079874)

[Figure 14: search screen arrangement 7](#_Toc42079875)

[Figure 15: 2 x 2 pattern search screen 8](#_Toc42079876)

[Figure 16: search screen arranged in a column 9](#_Toc42079877)

[Figure 17: top 10 chart small version 10](#_Toc42079878)

[Figure 18: top 10 chart medium version 10](#_Toc42079879)

[Figure 19: top 10 chart large version 11](#_Toc42079880)

# Table of Tables

[Table 1: ACME search requirements 4](#_Toc42079881)

[Table 2: application file list 1](#_Toc42079882)

[Table 3: top 10 chart sizes 9](#_Toc42079883)

# Analysis and Setup

## Meeting Agenda and Minutes

## Source Control

GitHub was selected to manage source and version control. A repository was created with the following address:

<https://github.com/AlanPedersen/RAD_project_movie_database>

A snapshot of the initial repository is presented below:

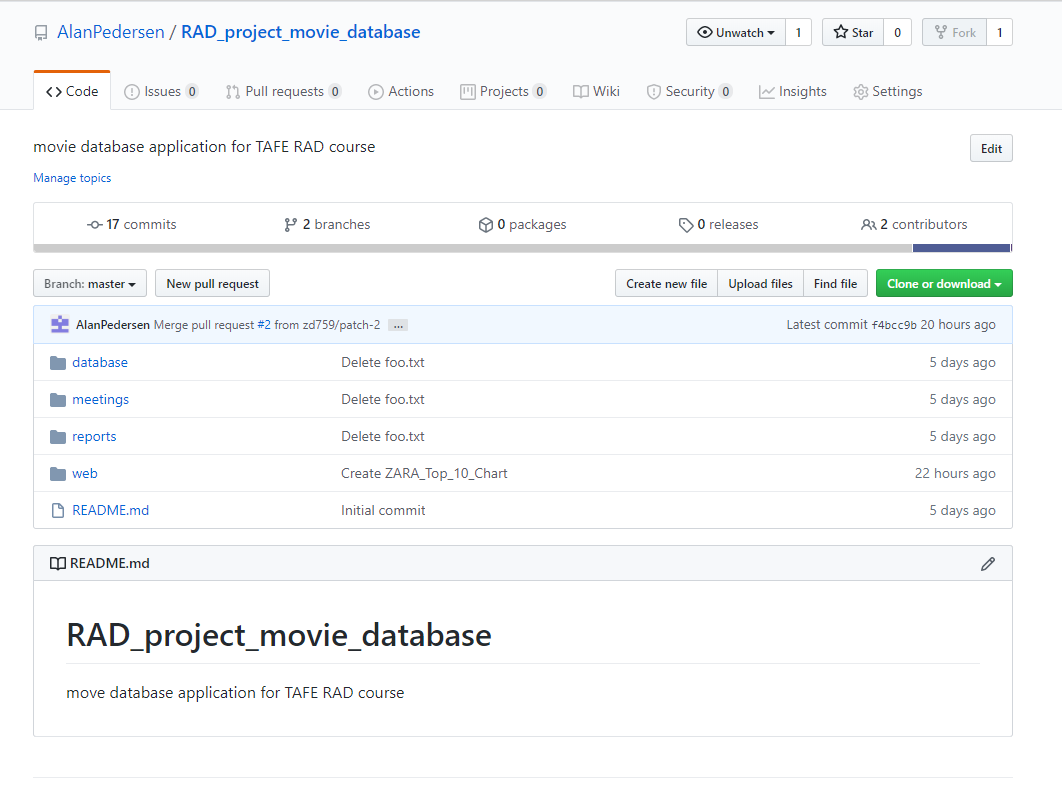


Figure : GitHub repository snapshot

## Project Management

Project Libre was adopted for project management.

# Sprint 1

## Sprint 1 project management plan

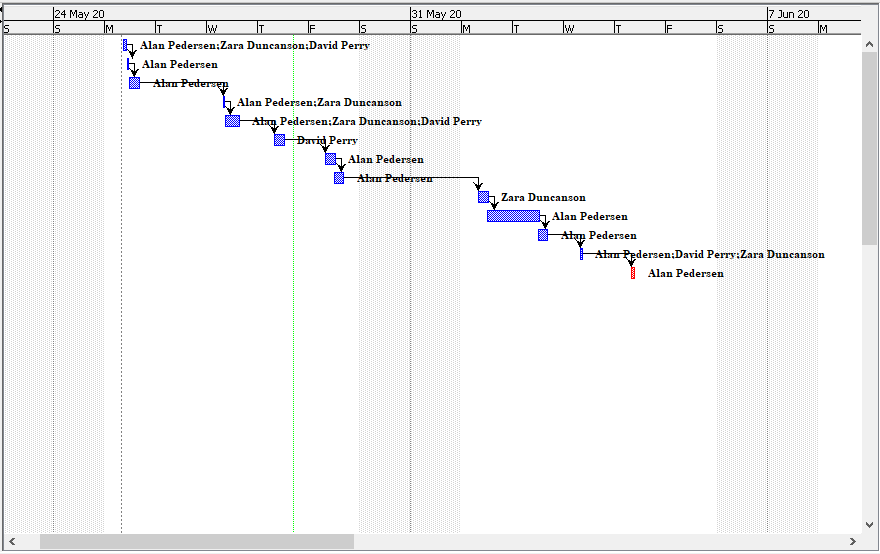


Figure : sprint 1 Gantt chart

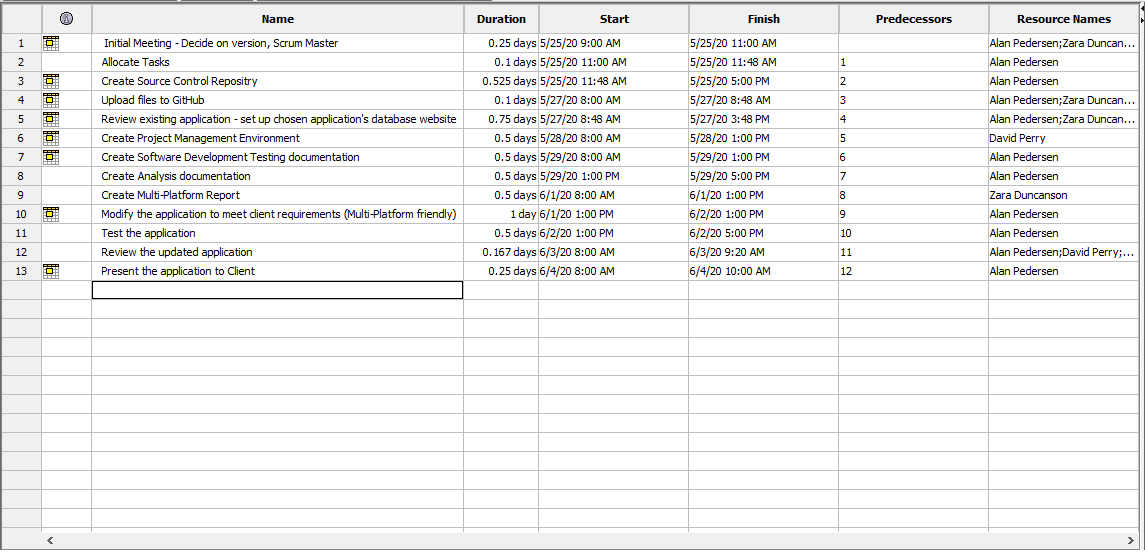


Figure : sprint 1 project plan

## Development and Testing

### Code Review

The following files are used by the application

|  |  |
| --- | --- |
| File Name | Description |
| bootstrap.min.css | Bootstrap css formatting code file |
| connect.pdo.php | Open a connection to the database |
| demo.css | Default css formatting |
| index.php | Default web application entry page |
| movie\_list\_genre.php | Read the genre codes from the database and create the selection list for the search form. |
| movie\_list\_rating.php | Read the rating codes from the database and create the selection list for the search form. |
| movie\_list\_scr.php | Create the SQL query from the form data to use when selecting movie records from the database. Run the query and format the results into a HTML table for display |
| movie\_top\_10\_google\_data\_scr.php | Get the list of top 10 searched for movies from the database. Format data so it is suitable for google charts |
| movie\_year\_limits.php | Get the minimum and maximum year values from the database |
| SearchMovies.php | Display the form to search for movies and display the results of the search. |
| Top10.php | Display a chart of the top 10 most searched for movies |

Table : application file list

Each file was reviewed manually and with the php code sniffer application.

### Functional Testing

The following screen dump shows the query page when first loaded, the Title field is preselected in the Sort By list:

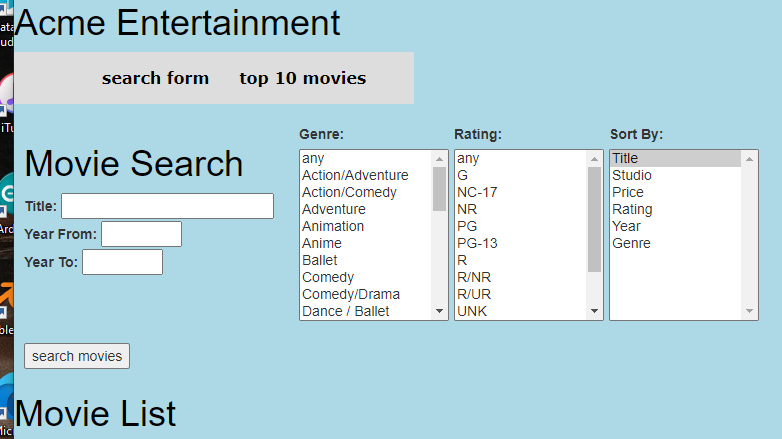


Figure : the search page

A search was conducted using the term “sing” in the title box, 18 records were returned all the titles containing “sing” within them:

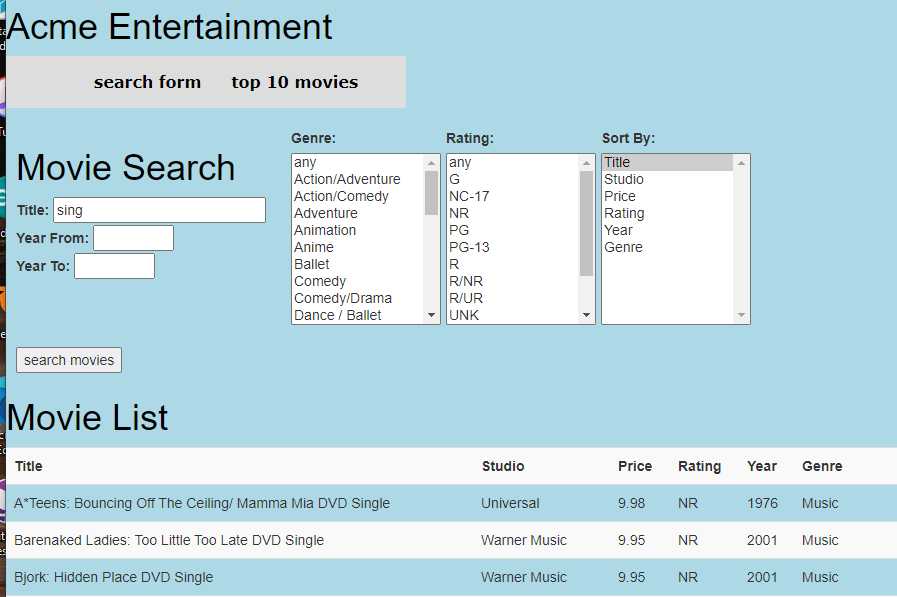


Figure : searching by title

A Year From value of 1970 was set and the results ordered by Year, only movies with a Year greater than 1970 are included in the report:

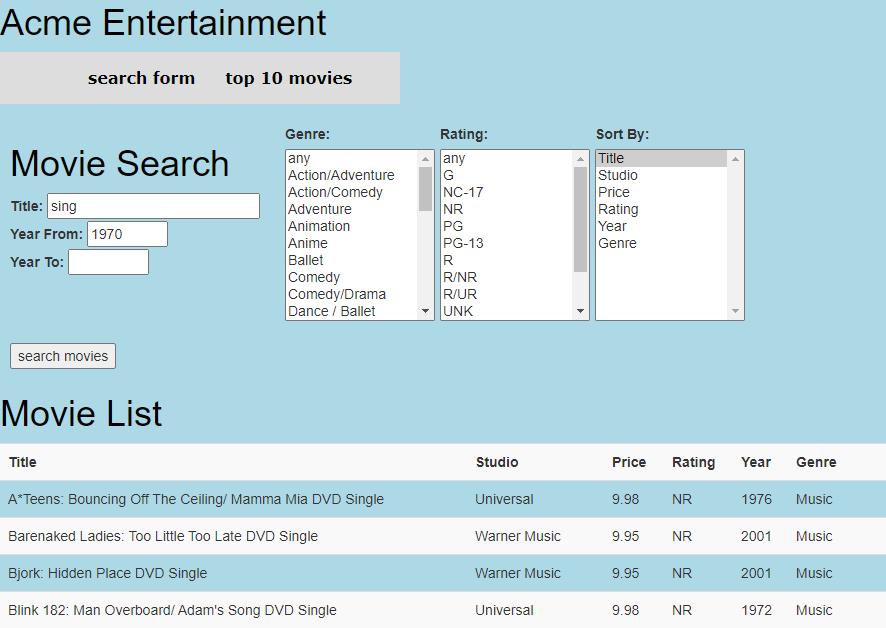


Figure : searching by title and year from

A Year To value of 1990 was set and the results ordered by Year, only movies with a Year less than 1990 are included in the report:

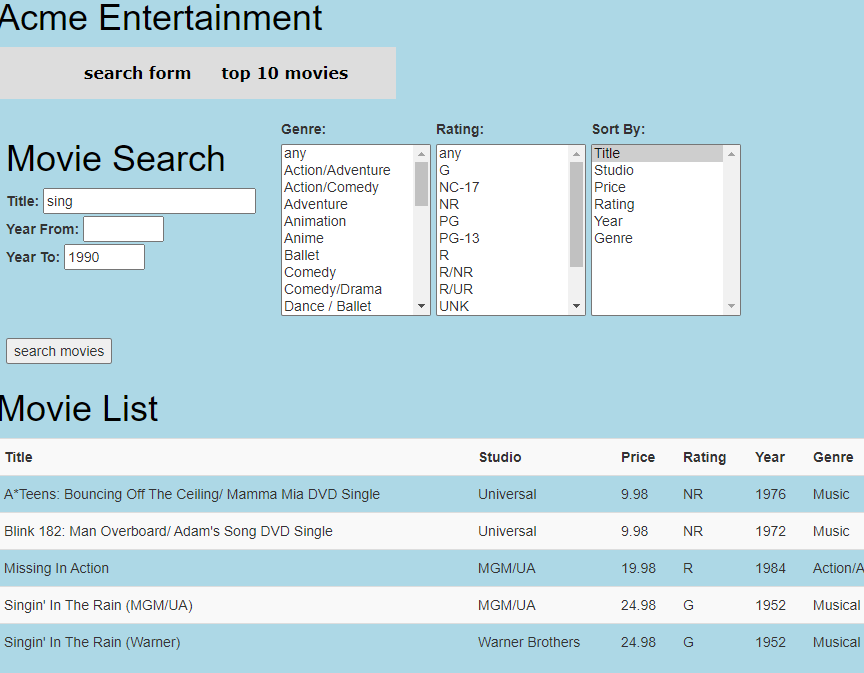


Figure : searching by title and year to

A Year From value of 1970 and a Year To value of 1990 was set and the results ordered by Year, only movies with a Year between 1970 and 1990 are included in the report:

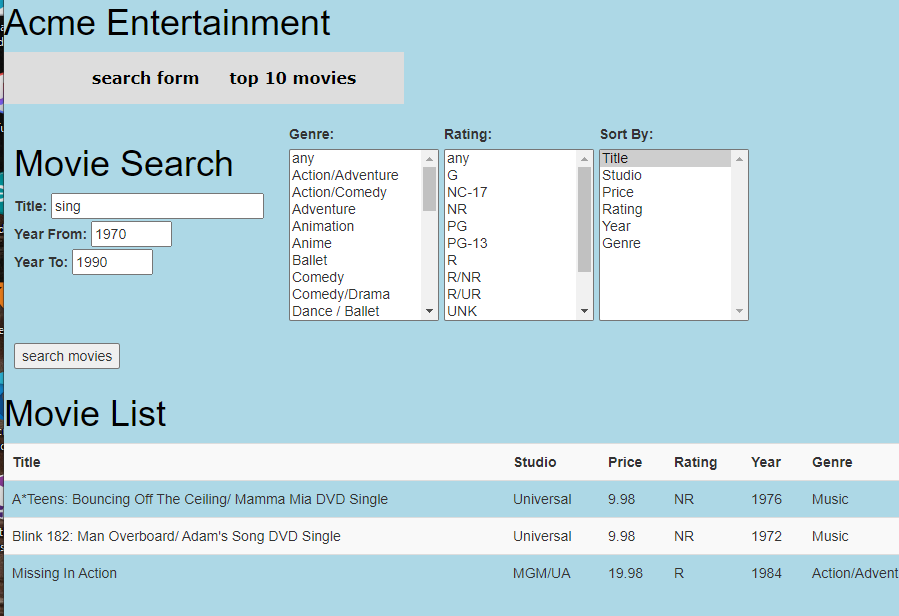


Figure : searching by title, year from and year to

The year values were swapped to find movies outside of the selected range:

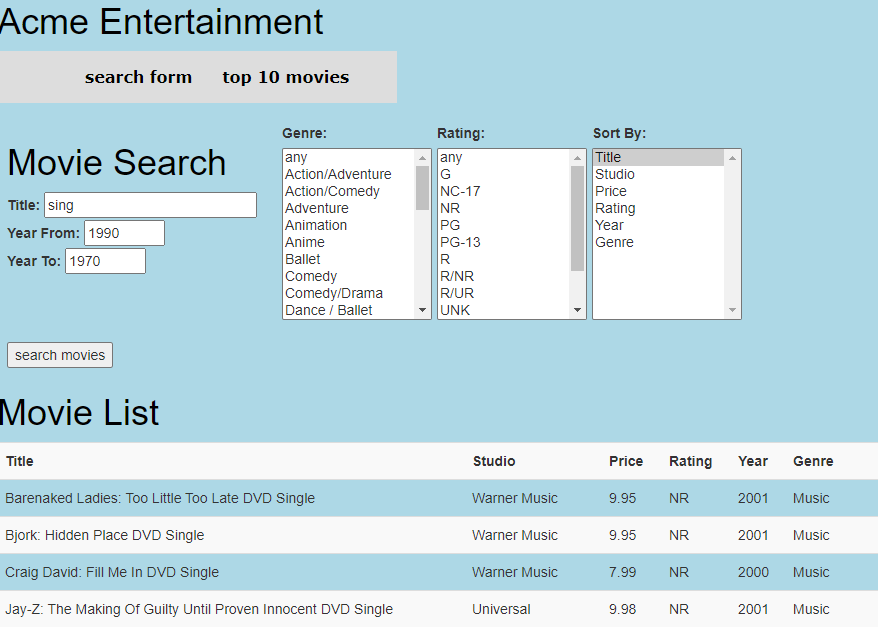


Figure : searching by Title and date ranges

The Music, Musical and Comedy Genres were added to the search criteria:

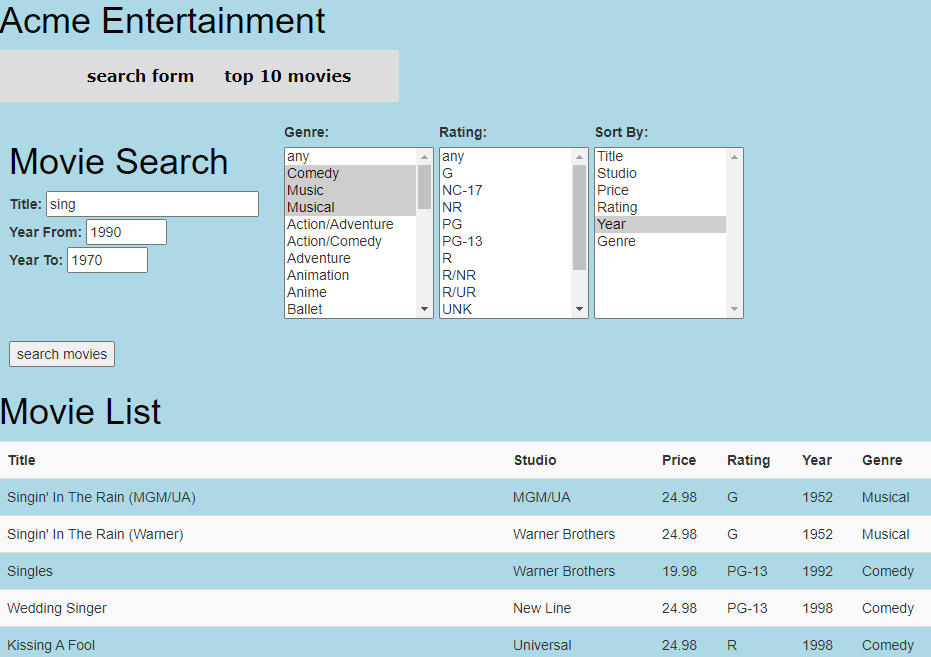


Figure : including Genre values in the search

The rating was constrained to G, PG and PG-13 and sorted by Rating:

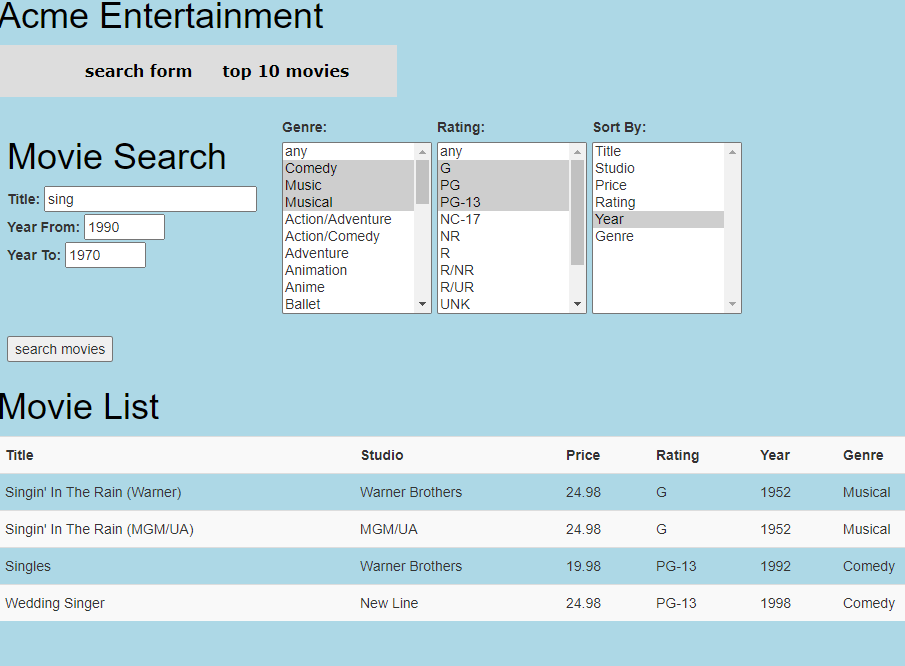


Figure : searching using all constraints

All the search elements were tested individually and in combination, the results have not been included for brevity.

A screen capture of the Top 10 page is presented below:

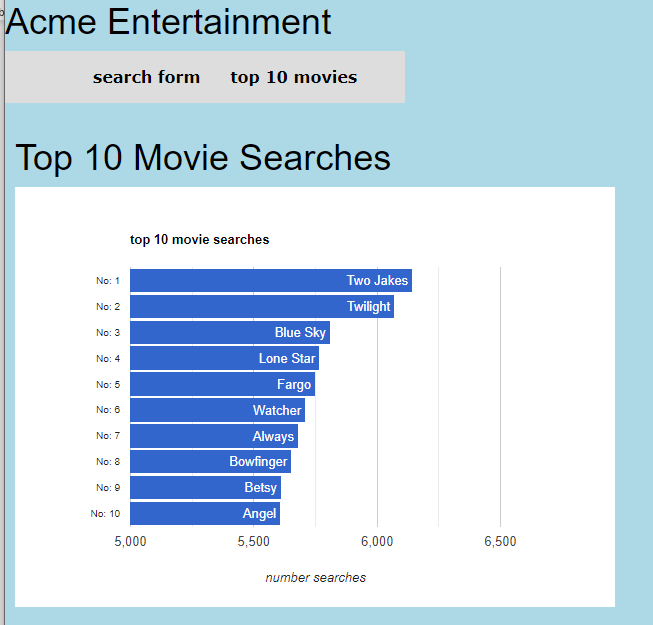


Figure : The Top 10 page

### Adaptive Web Design

A different approach was adopted for the two web pages.

The elements of the search page can be divided up into a series of logical blocks. The title and year search inputs, the three selection lists and the report area. These blocks can be rearranged independently to make the best use of the available screen space. The figure below shows the different blocks:

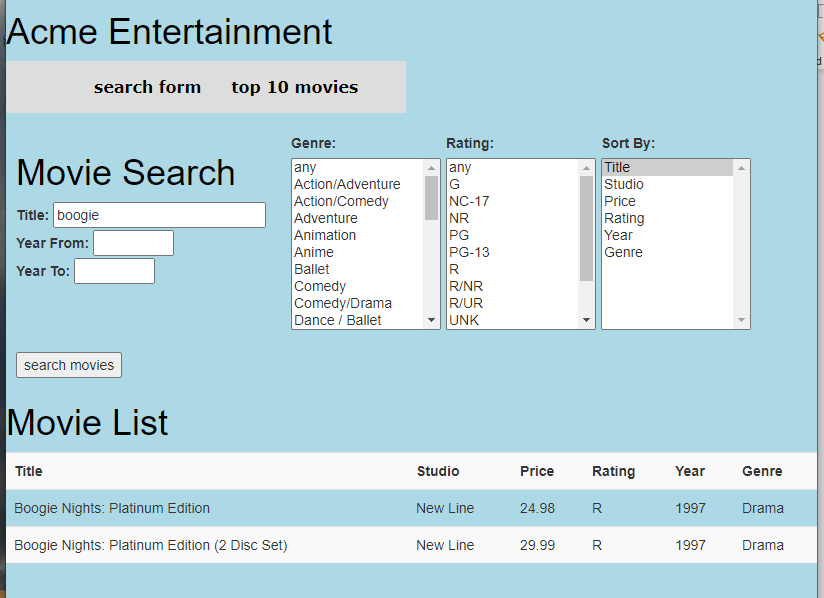


Figure : search screen arrangement

Three screen widths were targeted, less than 600 pixels, between 600 and 800 pixels and greater than 800 pixels.

For a screen width of greater than 800 pixels the search controls are arranged on a single line as shown in Figure 5 above.

For a screen width between 600 and 800 pixels the search controls are arranged into a 2 x 2 pattern, as shown in Figure 6 below:

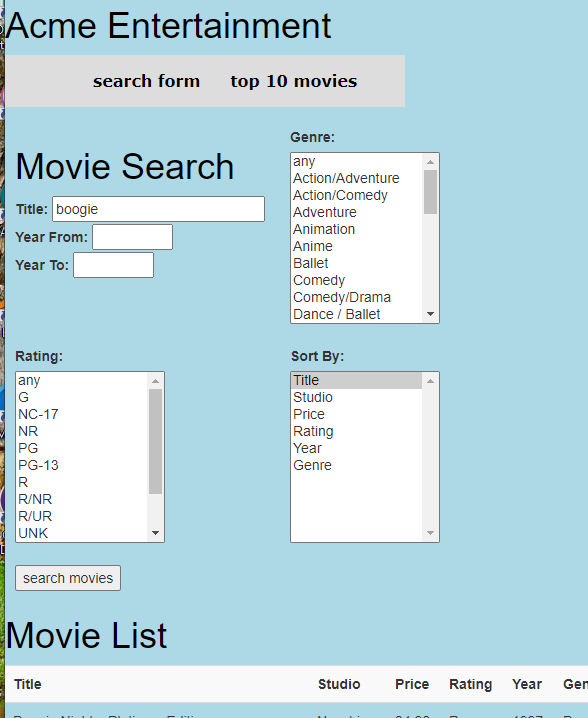


Figure : 2 x 2 pattern search screen

For a screen width of less than 600 pixels the search controls are arranged into a single column, as shown in Figure 7 below:

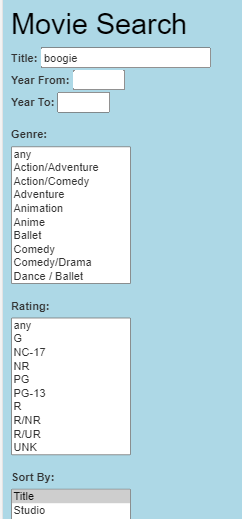


Figure : search screen arranged in a column

The top 10 search page displays a chart of the ten most searched for movies. The page creates three version of the chart each targeted at a range of screen widths. The versions are summarised in the following table:

|  |  |  |
| --- | --- | --- |
| Chart Size | Chart Dimensions | Screen Widths |
| Small | 325px x 220px | < 400px |
| Medium | 400px x 280px | >= 400px < 600px |
| Large | 600px x 420px | >= 600px |

Table : top 10 chart sizes

A screen dump of the small chart is shown below:

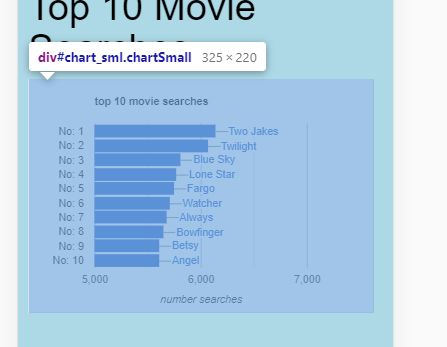


Figure : top 10 chart small version

A screen dump of the medium chart is shown below:

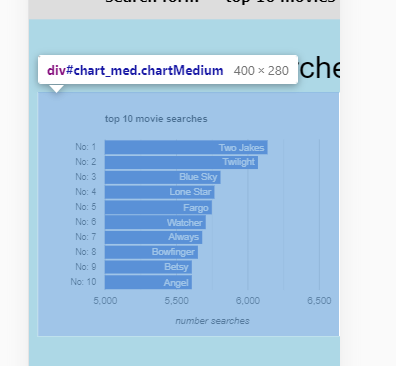


Figure : top 10 chart medium version

A screen dump of the large chart is shown below:

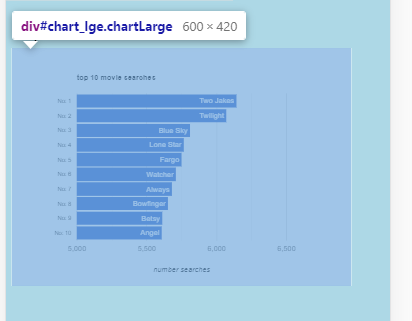


Figure : top 10 chart large version

## Sprint 1 Review Meeting