Blake Topham, Erol Anil, Nathan Tai

Movie Database Project

RAD Project- Team Elite

# Contents

[Contents i](#_Toc74767558)

[Figures v](#_Toc74767559)

[Tables v](#_Toc74767560)

[Source Control Screenshot 1](#_Toc74767561)

[Analysis Report 1](#_Toc74767562)

[Introduction: 1](#_Toc74767563)

[CITE business rules for software development: 1](#_Toc74767564)

[CITE Managed Services Quality Assurance: 2](#_Toc74767565)

[Full-cycle QA Testing. 2](#_Toc74767566)

[Document and Code Reviews 2](#_Toc74767567)

[Defect Tracking 2](#_Toc74767568)

[Configuration Management 2](#_Toc74767569)

[Process Monitoring 2](#_Toc74767570)

[Risk Management 2](#_Toc74767571)

[Acme Entertainment Pty Ltd development requirements: 3](#_Toc74767572)

[Create a working prototype of a movie database and webpage frontend. 3](#_Toc74767573)

[Create a multi-platform report to determine if we should use an adaptive or responsive approach for multi-platform support. 3](#_Toc74767574)

[Implement the chosen approach into the prototype. 3](#_Toc74767575)

[Team Meeting 4](#_Toc74767576)

[Client Meeting 1 5](#_Toc74767577)

[Consistency Validation Report 6](#_Toc74767578)

[Adaptive Design 2](#_Toc74767579)

[Description 2](#_Toc74767580)

[Advantages 2](#_Toc74767581)

[Disadvantages 2](#_Toc74767582)

[Responsive Design 2](#_Toc74767583)

[Description 2](#_Toc74767584)

[Advantages 2](#_Toc74767585)

[Disadvantages 2](#_Toc74767586)

[Project Selection 3](#_Toc74767587)

[Navigation Chart 3](#_Toc74767588)

[Software Testing Plan 4](#_Toc74767589)

[Introduction 4](#_Toc74767590)

[Scope 4](#_Toc74767591)

[In Scope 4](#_Toc74767592)

[Out of Scope 4](#_Toc74767593)

[Quality Objective 5](#_Toc74767594)

[Objectives 5](#_Toc74767595)

[CITE MS QA Standards 5](#_Toc74767596)

[Roles and Responsibilities 6](#_Toc74767597)

[Scrum Master 6](#_Toc74767598)

[Configuration Manager 6](#_Toc74767599)

[Developer 6](#_Toc74767600)

[Test Methodology 6](#_Toc74767601)

[Overview 6](#_Toc74767602)

[Test Levels 6](#_Toc74767603)

[Test Tables 7](#_Toc74767604)

[Screenshots 9](#_Toc74767605)

[Bug Triage 1 14](#_Toc74767606)

[Suspension Criteria and Resumption Requirements 15](#_Toc74767607)

[Test Completeness 15](#_Toc74767608)

[Test Deliverables 15](#_Toc74767609)

[Resource and Environment Needs 16](#_Toc74767610)

[Testing Tools 16](#_Toc74767611)

[Test Environment 16](#_Toc74767612)

[Glossary 16](#_Toc74767613)

[Source Control Snapshot 17](#_Toc74767614)

[GitHub Repository 17](#_Toc74767615)

[Performance Report 18](#_Toc74767616)

[Definition 18](#_Toc74767617)

[Goal 18](#_Toc74767618)

[Assumptions 18](#_Toc74767619)

[Process Description 18](#_Toc74767620)

[Optimizer Tools 20](#_Toc74767621)

[PHPLOC 20](#_Toc74767622)

[Data Need Analysis 21](#_Toc74767623)

[Collecting 21](#_Toc74767624)

[Aggregating 21](#_Toc74767625)

[Visualizing 21](#_Toc74767626)

[Interpreting 22](#_Toc74767627)

[Analyzing 22](#_Toc74767628)

[Reporting 22](#_Toc74767629)

[Performance of Results 23](#_Toc74767630)

[Performance of Top 10 Movies 23](#_Toc74767631)

[Performance of Home Page 24](#_Toc74767632)

[Software Review Plan 25](#_Toc74767633)

[Current Architecture 25](#_Toc74767634)

[Software Quality Issues 26](#_Toc74767635)

[Modularity 26](#_Toc74767636)

[Portability 26](#_Toc74767637)

[Efficiency 26](#_Toc74767638)

[Extensibility 26](#_Toc74767639)

[Readability 26](#_Toc74767640)

[Proposed Improvements 26](#_Toc74767641)

[Suggestions/Recommendations (Optional) 26](#_Toc74767642)

[Testing 27](#_Toc74767643)

[Scope 27](#_Toc74767644)

[Test Table 28](#_Toc74767645)

[Screenshots 30](#_Toc74767646)

[Bug Triage 2 40](#_Toc74767647)

[Team Meeting 2 41](#_Toc74767648)

[Client Meeting 2 43](#_Toc74767649)

[Source Control Snapshot 44](#_Toc74767650)

[Optimization Report 45](#_Toc74767651)

[Description 45](#_Toc74767652)

[Need for Optimization 45](#_Toc74767653)

[Case Studies 45](#_Toc74767654)

[Best Practices 45](#_Toc74767655)

[Performance Tuning Steps Followed 46](#_Toc74767656)

[Performance Tuning Steps Followed 47](#_Toc74767657)

[Previous Observations 48](#_Toc74767658)

[Network Analytics 49](#_Toc74767659)

[Memory Status 50](#_Toc74767660)

[Security 51](#_Toc74767661)

[Warnings / Error Logging 52](#_Toc74767662)

[Current Performance Extract Report 53](#_Toc74767663)

[Updated Software Testing Plan 54](#_Toc74767664)

[Reflections 54](#_Toc74767665)

[Testing Overview 54](#_Toc74767666)

[Scope 54](#_Toc74767667)

[Test Table 55](#_Toc74767668)

[Screenshots 57](#_Toc74767669)

[Team Meeting 3 70](#_Toc74767670)

[Client Meeting 3 72](#_Toc74767671)

[Bug Triage 3 73](#_Toc74767672)

# Figures

[Figure 1- GitHub Screenshot 1](#_Toc74769252)

[Figure 2- GANTT Chart 1](#_Toc74769253)

[Figure 3- GANTT Chart (continued) 2](#_Toc74769254)

[Figure 4- Test Case 1 failed 19](#_Toc74769255)

[Figure 5- Name search 19](#_Toc74769256)

[Figure 6- Name, Genre 19](#_Toc74769257)

[Figure 7-Name, Genre, Rating 20](#_Toc74769258)

[Figure 8- Name, Genre, Rating, Year 20](#_Toc74769259)

[Figure 9- Top ten 21](#_Toc74769260)

[Figure 10- Runs on PC 21](#_Toc74769261)

[Figure 11- Runs on iPad 22](#_Toc74769262)

[Figure 12- Runs on Galaxy S9 23](#_Toc74769263)

[Figure 13- GitHub repository 27](#_Toc74769264)

[Figure 14- GANTT chart sprint two 27](#_Toc74769265)

[Figure 15- Search by name 39](#_Toc74769266)

[Figure 16- Name, Genre 40](#_Toc74769267)

[Figure 17-Name, Genre, Rating 41](#_Toc74769268)

[Figure 18- Name, Genre, Rating, Year 42](#_Toc74769269)

[Figure 19- Top ten 43](#_Toc74769270)

[Figure 20- Runs on PC 43](#_Toc74769271)

[Figure 21- Runs on iPad 44](#_Toc74769272)

[Figure 22- Runs on S5 45](#_Toc74769273)

[Figure 23- User unsubscribes, database not yet reflected 46](#_Toc74769274)

[Figure 24- User unsubscribes, admin email 47](#_Toc74769275)

[Figure 25- Admin unsubscribes user 48](#_Toc74769276)

[Figure 26- GitHub screenshot 53](#_Toc74769277)

[Figure 27- GANTT chart 54](#_Toc74769278)

[Figure 28- Search by name 67](#_Toc74769279)

[Figure 29- Name, Genre 68](#_Toc74769280)

[Figure 30-Name, Genre, Rating 69](#_Toc74769281)

[Figure 31- Name, Genre, Rating, Year 70](#_Toc74769282)

[Figure 32- Top ten 71](#_Toc74769283)

[Figure 33- Runs on PC 71](#_Toc74769284)

[Figure 34- Runs on iPad 72](#_Toc74769285)

[Figure 35- Runs on S5 73](#_Toc74769286)

[Figure 36- User unsubscribes, database not yet reflected 74](#_Toc74769287)

[Figure 37- User unsubscribes, admin email 75](#_Toc74769288)

[Figure 38- Admin unsubscribes user 76](#_Toc74769289)

[Figure 39- Admin logs in 77](#_Toc74769290)

[Figure 40- User leaves a score 78](#_Toc74769291)

[Figure 41- Ten highest-rated 79](#_Toc74769292)

# Tables

[Table 1- Bug Report 27/05/2021 14](#_Toc74767713)

[Table 2- Timeline of Test Deliverables 15](#_Toc74767714)

Sprint One

# Source Control Screenshot(S1)

Link: <https://github.com/profiteroles/RAD>

Screenshot:

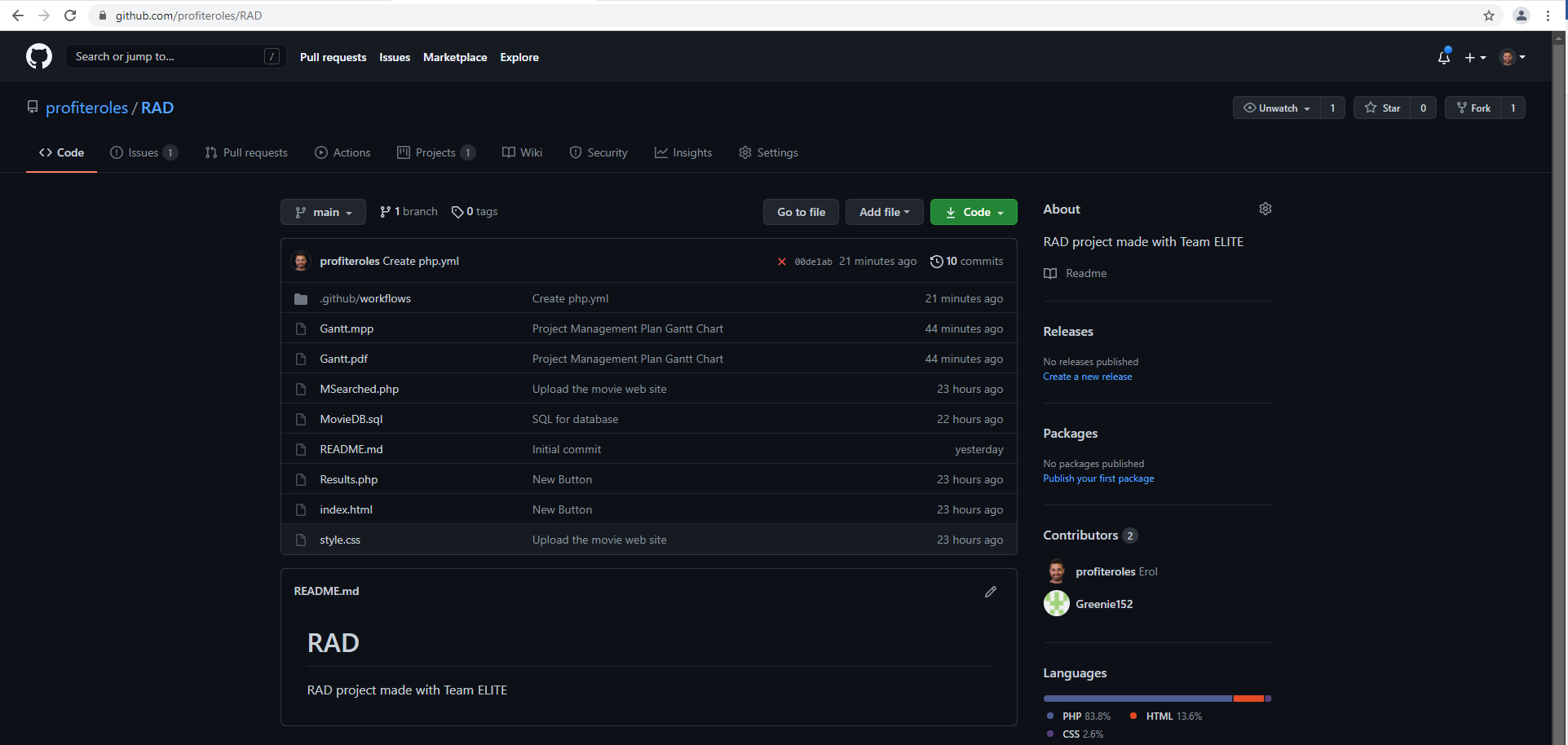


Figure 1- GitHub Screenshot

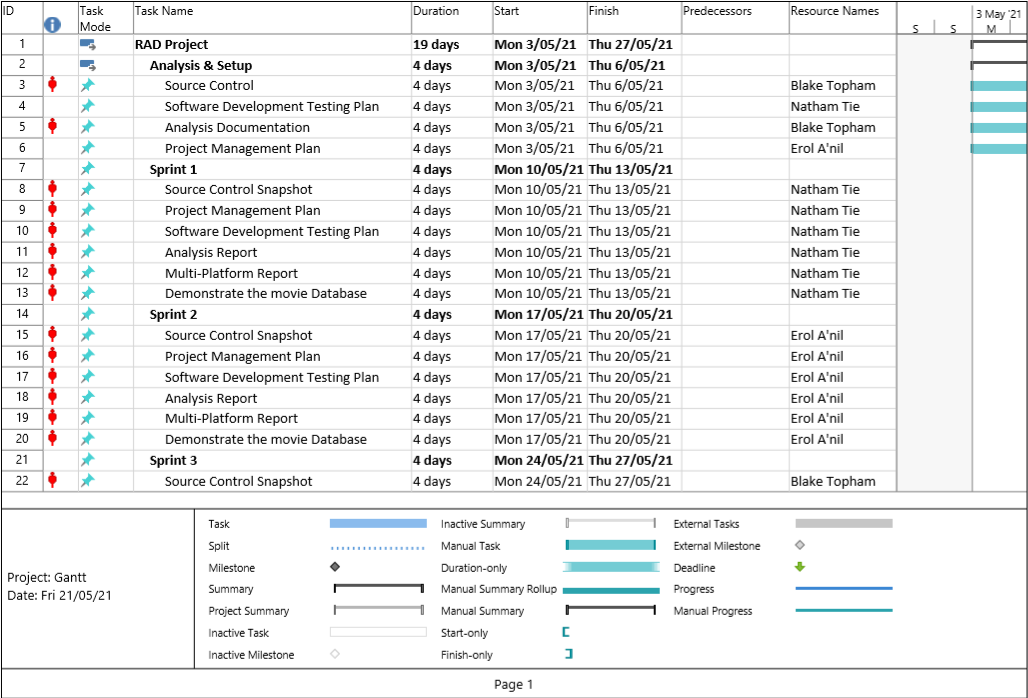


Figure 2- GANTT Chart

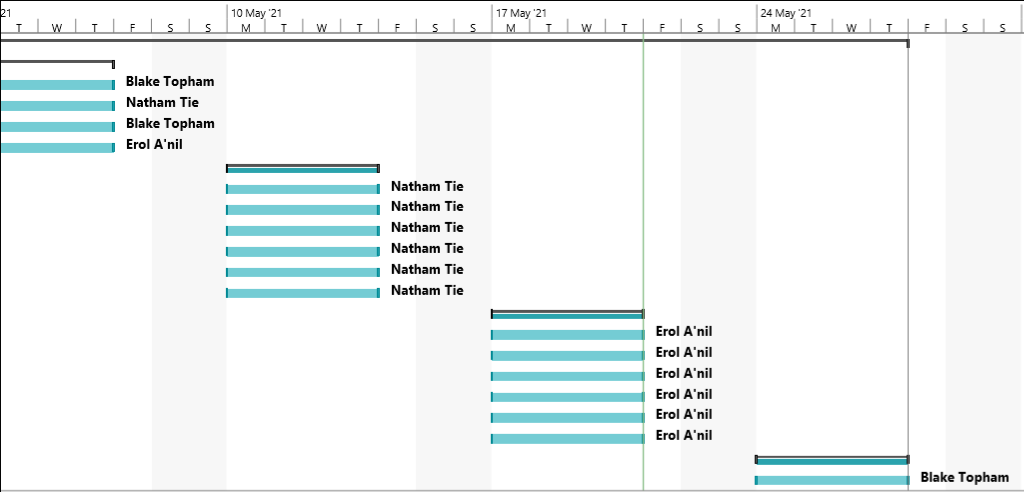


Figure 3- GANTT Chart (continued)

# Analysis Report(S1)

## Introduction:

The purpose of this document is to analyze and outline the business rules and QA practices of our development house (CITE managed services) as well as the development requirements outlined by our client.

## CITE business rules for software development:

These are the rules of our development house that we must adhere to throughout development.

For a more in depth version of these rules the company website is located at this url: https://www.citems.com.au/

But at a most basic level we adhere to these core tenants:

• We always put the client’s needs and interests first and ensure that everything meets our client’s standards and requirements.

• We always allow for ourselves be held accountable for anything we make’s successful operation.

• We see our staff and reputation as our best assets and operate in an ethical manner in any situation to maintain them.

• We aim for professionalism and the highest possible quality in everything we do to ensure all our work is of the highest possible standard.

• We are always open to innovation when it comes to making the best possible product for our client.

• We ensure that our development teams work together as a group to ensure that our client’s needs are met.

We believe that if we adhere to these tenants and adhere to or primary coding standards that are followed company-wide that we will always work to create the best possible products for our clients’ needs and interests.

As an aside while our coding standards are different for each coding language we use all are projects are designed to adhere to the ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes software standard.

## CITE Managed Services Quality Assurance:

This is the outline we must follow when performing testing and quality assurance.

While the full details of CITE’s quality assurance practices can be found at:

https://www.citems.com.au/services/application-development/quality-management/?fbclid=IwAR1yYGMEAfMhdPAQyn3YjIVx9xm99XPEEyWZPT9CLpHRFQfyydZbFa1nyPw

It can be summarized into a few key points:

### Full-cycle QA Testing.

This is performing tests for the sake of quality assurance and fault prevention each development sprint and/or major code change to ensure that the resulting program is always operating at its best.

### Document and Code Reviews

Our documents and code are checked by those not doing the core development to ensure that they are up to quality standards from a separate person’s perspective

### Defect Tracking

This is the process of keeping track of all known flaws and defects in the program/product and ensuring they are all given attention where needed and not allowed to just sit there.

### Configuration Management

This is the process of handling changes in a manner that both allows for rollback and tracking and helps allow for development to go smoothly by keeping track of each version of the program to what works

### Process Monitoring

This is the act of keeping track of each process the program in performing to ensure that everything operates and flows as intended.

### Risk Management

This is analyzing data and software risks and assessing their risk level through a mix of likelihood and impact and implementing prevention methods in order of importance.

## Acme Entertainment Pty Ltd development requirements:

These are the requirements from the client that we must meet in development and for this sprint are as follows:

### Create a working prototype of a movie database and webpage frontend.

This webpage must allow for searching of the movie database using a variety of different parameters and have the ability to display the 10 most searched movies in the database.

### Create a multi-platform report to determine if we should use an adaptive or responsive approach for multi-platform support.

This report must include the merits of each approach as well as the chosen approach and reasons for the decision.

### Implement the chosen approach into the prototype.

Once the chosen approach is implemented testing should be performed to ensure that it is working as intended.

### Team Meeting

#### Meeting Minutes

27/05/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 03/06/2021, 9:00AM, Murdoch Campus

**1. Team Role Assignment**

**EROL: Source Control Snapshot**

• Create GitHub, fork to rest of team

• Create Trello account, link with team

• Add tasks to Trello board

• Create documentation with screenshots/evidence of both, add to master document

**EROL: Project Management Plan**

• GANTT Chart for SPRINT ONE (week six)

o Coding

o Documentation

• Include deadlines and milestones

• Include who does what task and by when

• Create Master Document based on agreed template

**BLAKE: Analysis Report**

• CITE business rules for software development

• CITE Managed Services Quality Assurance

• Acme Entertainment Pty Ltd development requirements

**BLAKE: Multi-Platform Report**

• Adaptive vs Responsive

**NATHAN: Software Development Testing Plan**

• Incorporates the QA standards of CITE (from analysis report?)

• What will be tested and at what stages

• Testing types, methods

**EROL: Demonstrate the Movie Database Application**

• Read and understand all documentation

• Read and understand all code

• Demonstrate application on three different sized digital devices

### Client Meeting 1

#### Meeting Minutes

27/05/2021

Present: Client, Blake Topham, Erol Anil, Nathan Tai

Next meeting: 03/06/2021, 9:30AM, Murdoch Campus

**1. Project Requirements**

* Project requirements discussed and documented.
* Project planning commenced.
* Prototype requested.
* Requirements noted and agreed.

**2. Quality Assurance Standards**

* Quality Assurance standards set forth and agreed upon (CITEMS QA standards).

Available at:

https://www.citems.com.au/services/application-development/quality-management/

## Consistency Validation Report

Overall the only data being tested for consistency validation is the database itself. The current measures we have in place in case of failure are that we have the original SQL file used to create the database stored in multiple locations such as personal data storage and a GitHub repository.

As such if the file on the server is corrupted or damaged in any way the database can be easily restored. This will lose any updates made since the last time this backup was renewed but will restore the database to working order.

A table showing how the data will be in a multitude of events is included below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event no** | **description** | **Main file status** | **Backup status** | **Explanation** |
| 1 | Main file is corrupted or deleted | Main file recoverable from backup | Unchanged | Problem can be resolved easily however site will not operate in the meantime |
| 2 | One backup lost | unchanged | Backup can be restored either from second backup instance or generated again from main file | This will have absolutely no effect other than the need to re-generate a backup |
| 3 | All backups lost | unchanged | All backups gone but can be re-generated from main file fairly painlessly and restored | This will be a dangerous situation however it will be easy to resolve as it will simply need the generation of a new backup and it being placed in all the places it needs to be |
| 4 | Main file and all backups lost | Lost | Lost | This would be a catastrophic situation however the chance of it happening are insanely small as it would require the simultaneous deletion or corruption of multiple files stored entirely separately from one another |

## Adaptive Design

### Description

Adaptive design refers to the creation of websites that operate within static thresholds for display resolution, known as the viewing context. The viewing context of the target device is assessed and a set of stylings are applied accordingly to the elements of the web page.

### Advantages

The main advantage of adaptive design is the final presentation of the webpage’s display is much more predictable during development. This is due to the elements being sized according to a selected fixed layout, determined by the screen’s resolution.

Adaptive designs can be made applicable to older devices that do not support responsive designs.

### Disadvantages

Adaptive design is limited to the static definitions of resolutions defined by the developer. As the webpage is loaded, and the resolution is assessed, a pre-determined layout is selected appropriately. Adaptive design has limitations here bound by the definitions of the developer.

Once a webpage is loading, adaptive layout selection is no longer applied. For example, in modern devices where it is possible to scale the resolution of the web browser on screen, the webpage will not adapt its design to a modified screen resolution after the page has loaded.

## Responsive Design

### Description

Responsive web design refers to the creation of websites using dynamically sized elements that are responsive to their parent containers, and ultimately the display resolution.

### Advantages

Responsive design is extremely flexible in its rendering approach. The designs are fluid are able to display responsively to any viewing context, in contrast to adaptive design’s fixed layout approach.

Statistics show that more than half of the global internet traffic (52+%) is from a mobile device (Hosting Tribunal, 2020). As mobile devices continue to be produced in various shapes and sizes, responsive designs gain an advantage of dynamically adapting to these devices.

Responsive designs achieve higher site rankings by search engines like Google or Yahoo!, therefore leading to more web traffic.

Responsive designs can reduce the need, and therefore the cost of, developing a separate mobile-friendly version of the website.

### Disadvantages

Site navigation can become more challenging to keep consist with responsive designs. As the screen size decreases, so too do the elements portraying the navigation paths.

Responsive web designs can also take longer to load than adaptive designs. This is due to their relative nature where the elements are sized in relation to the size of the parent elements. Longer loading times can often leading to a bad browsing experience on the user’s behalf.

Older browsers, or older devices who are limited with software, may not be compatible with responsive designs and will therefore show incorrectly.

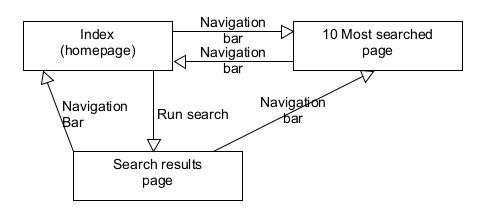
## Project Selection

For this project we have chosen a Responsive design.

Responsive design will provide an overall better look and feel (user experience) to the finished product and result in a higher Search Engine Optimization ranking. The majority of elements will be designed to be responsively sized and placed.

### Navigation Chart

The following illustrates the navigation pathways of the webpage prototype in all directions:



# Software Testing Plan(S1)

## Introduction

This software test plan will utilize an Agile testing methodology which forms part of the overall Agile SDLC model used for developing the project. Given the simplistic nature of the project, static testing may not be necessary. The program may also be tested manually; as there is no need for numerous, repetitive actions to test the relatively few required functions. The main test metrics used will be error count and error frequency.

The requirements of this project suggest the employment of a Behavioral Testing Strategy which involves the use of Black Box tests on the application. Testing will focus primarily on key functionality from the user’s perspective. Project scope does not specify requirements pertaining to user experience.

## Scope

### In Scope

* User authentication
* Display database
* Search database
  + With any combination of parameters
* Top ten movies
* Operation on three different sized devices
  + PC
  + Tablet/Laptop
  + Phone

### Out of Scope

* User experience
* Login security
* User account recovery
* Thumbnails with theatrical posters/cover art
* User reviews

## Quality Objective

The quality objectives will connect measures of software quality to tangible and achievable goals for the development team. It is through these objectives that software quality assurance will be performed.

### Objectives

1. Ensure functional requirements are met
2. Test all relevant code pathways
3. Ensure multi-device compatibility
4. Ensure program is free of errors that:
   1. *Break vital functionality*
   2. *Cause crashes/ freezing*
   3. *Significantly affect reasonable usability*
   4. *Threaten data integrity*

### CITE MS QA Standards

The following are the tasks and objectives of CITE Managed Services Quality Assurance System which will guide the quality assurance standards of the project, as detailed on the CITEMS Website:

* Elaboration and implementation of procedures and regulations for software development process based on industry standards and best practices;
* Product lifecycle monitoring to ensure compliance with established processes and guidelines
* Product quality verification and validation to ensure that it complies with clients’ business needs and expectations;
* Establishment of an effective collaboration between all project team members.

## Roles and Responsibilities

### Scrum Master

The Scrum Master will set forth the testing plan. They will delegate and share testing duties among the team. They will determine which code pathways/scenarios are to be tested, and their priority. They will also outline the test methodology, including test levels.

### Configuration Manager

The Configuration Manager will be responsible for approving changes committed, and monitoring the effects of these changes. They will identify any issues caused by fixes/additions to the project and determine when they occurred, and who was responsible. They will also use this information to assist in the reproduction of successful configurations.

### Developer

Developer will produce a Bug Report detailing any new bugs with each iteration, and assist the team in determining the priority of each bug based on Severity, Frequency, and Risk.

## Test Methodology

### Overview

The Agile test methodology will be used for this project. This methodology is part of the Agile SDLC model used for the project’s development. Each software iteration is built and then tested thoroughly before moving on to the next iteration, ensuring no recurring issues will carry over between development cycles.

### Test Levels

Given the relatively simple nature of the project, some of these testing levels may be executed concurrently to eliminate redundant efforts e.g. Unit + Integration, System + Acceptance.

#### Unit Testing

Each unit of the program is tested independently to ensure proper function in accordance with project requirements.

#### Integration Testing

Units are tested when integrated with each other. Interactions between units are scrutinized and verified, and errors in these interactions are identified.

#### System Testing

The complete program is tested with all units fully integrated. Determines program’s ability to function as a whole.

#### Acceptance Testing

Determines whether a program is acceptable for the end user. Finished program is tested against business requirements.

## Test Tables

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Report S1 T1 | | | | | | | | |
| Project Name: RAD Sprint One | | | Test Type: Black Box | | | | | |
| Description: Movie Search Application | | | Date: 29/05/2021 | | | | | |
| Developer(s): Blake Topham, Erol Anil, Nathan Tai | | | Tester: Nathan Tai | | | | | |
| Test Case # | Test Case Name | Test Steps | | Test Data | | Expected Results | Evidence | Pass / Fail |
| 1 | Search by Name | Enter search terms | | Search term | | Matching results displayed | Figure 4 | Fail |
| 2 | Search by Name, Genre | Enter search terms | | Search term | | Matching results displayed | N/A | N/A |
| 3 | Search by Name, Genre, Rating | Enter search terms | | Search term | | Matching results displayed | N/A | N/A |
| 4 | Search by Name, Genre, Rating, Year | Enter search terms | | Search term | | Matching results displayed | N/A | N/A |
| 5 | Top Ten Movies | Click “10 Most Searched” button | | 10 movies with highest search count | | 10 movies with highest search count displayed | N/A | N/A |
| 6 | Runs on: Desktop PC | Run page in desktop browser | | Page elements | | Page runs | N/A | N/A |
| 7 | Runs on:  iPad | Run page in device simulation mode | | Page elements | | Page runs | N/A | N/A |
| 8 | Runs on: Galaxy S5 | Run page in device simulation mode | | Page elements | | Page runs | N/A | N/A |
| Test Completeness Criteria Achieved:  1. All test cases carried out successfully  2. All bugs fixed | | | | | Suspension Criteria Met:  1. Vital functionality broken  2. Error directly prevents further testing  3. Errors too many or too often  4. Error makes further testing redundant | | | |
| Testing Complete | | | | | Testing Suspended | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Report S1 T2 | | | | | | | | |
| Project Name: RAD Sprint One | | | Test Type: Black Box | | | | | |
| Description: Movie Search Application | | | Date: 29/05/2021 | | | | | |
| Developer(s): Blake Topham, Erol Anil, Nathan Tai | | | Tester: Nathan Tai | | | | | |
| Test Case # | Test Case Name | Test Steps | | Test Data | | Expected Results | Evidence | Pass / Fail |
| 1 | Search by Name | Enter search terms | | Search term | | Matching results displayed | Figure 5 | Pass |
| 2 | Search by Name, Genre | Enter search terms | | Search term | | Matching results displayed | Figure 6 | Pass |
| 3 | Search by Name, Genre, Rating | Enter search terms | | Search term | | Matching results displayed | Figure 7 | Pass |
| 4 | Search by Name, Genre, Rating, Year | Enter search terms | | Search term | | Matching results displayed | Figure 8 | Pass |
| 5 | Top Ten Movies | Click “10 Most Searched” button | | 10 movies with highest search count | | 10 movies with highest search count displayed | Figure 9 | Pass |
| 6 | Runs on: Desktop PC | Run page in desktop browser | | Page elements | | Page runs | Figure 10 | Pass |
| 7 | Runs on:  iPad | Run page in device simulation mode | | Page elements | | Page runs | Figure 11 | Pass |
| 8 | Runs on: Galaxy S5 | Run page in device simulation mode | | Page elements | | Page runs | Figure 12 | Pass |
| Test Completeness Criteria Achieved:  1. All test cases carried out successfully  2. All bugs fixed | | | | | Suspension Criteria Met:  1. Vital functionality broken  2. Error directly prevents further testing  3. Errors too many or too often  4. Error makes further testing redundant | | | |
| Testing Complete | | | | | Testing Suspended | | | |

### Screenshots

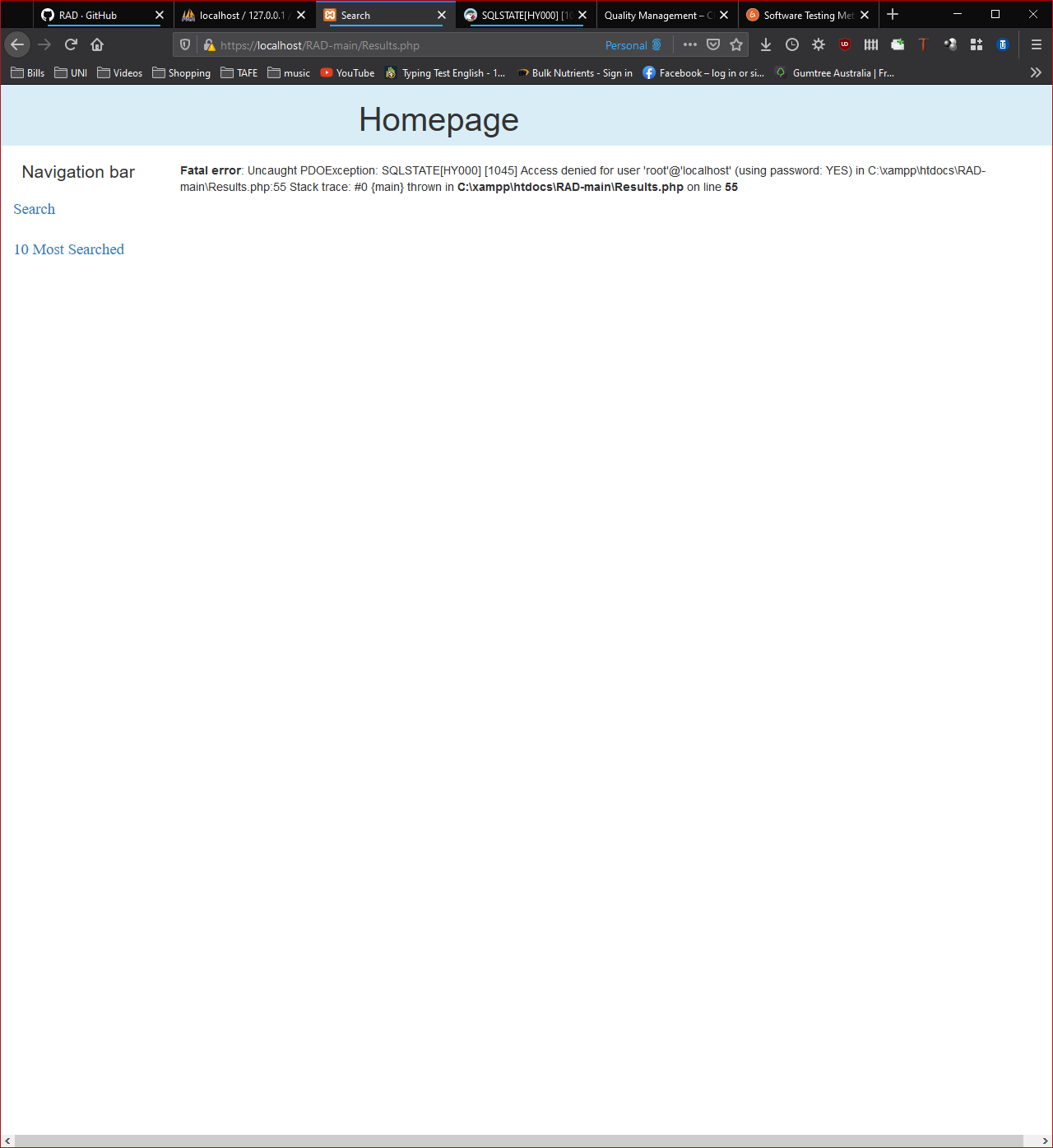


Figure 4- Test Case 1 failed

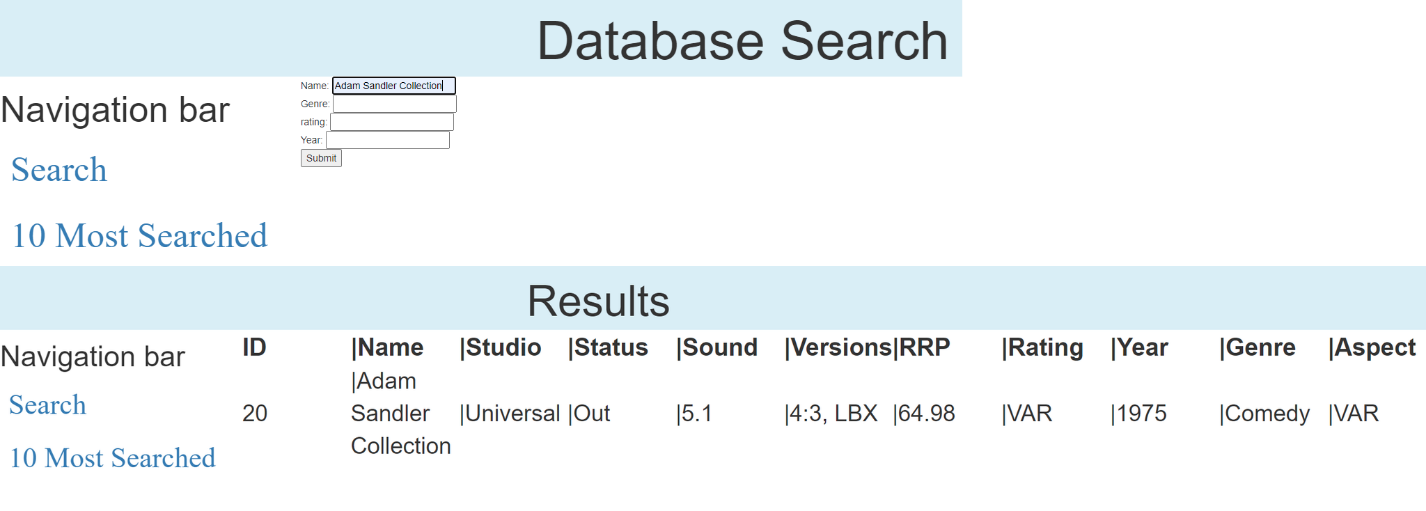


Figure 5- Name search

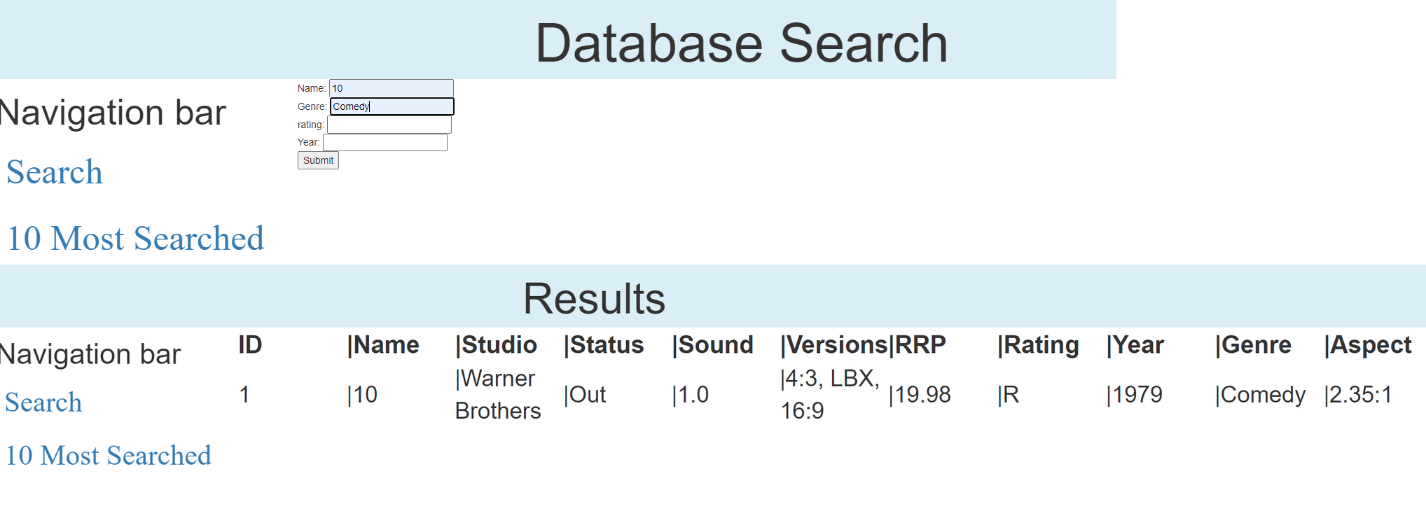


Figure 6- Name, Genre

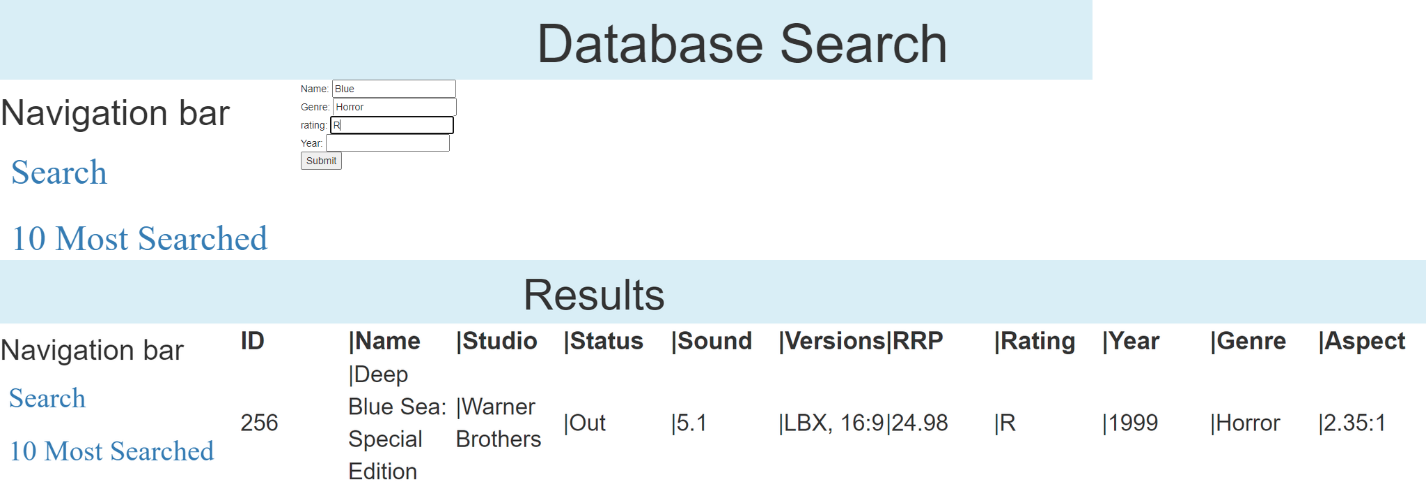


Figure 7-Name, Genre, Rating

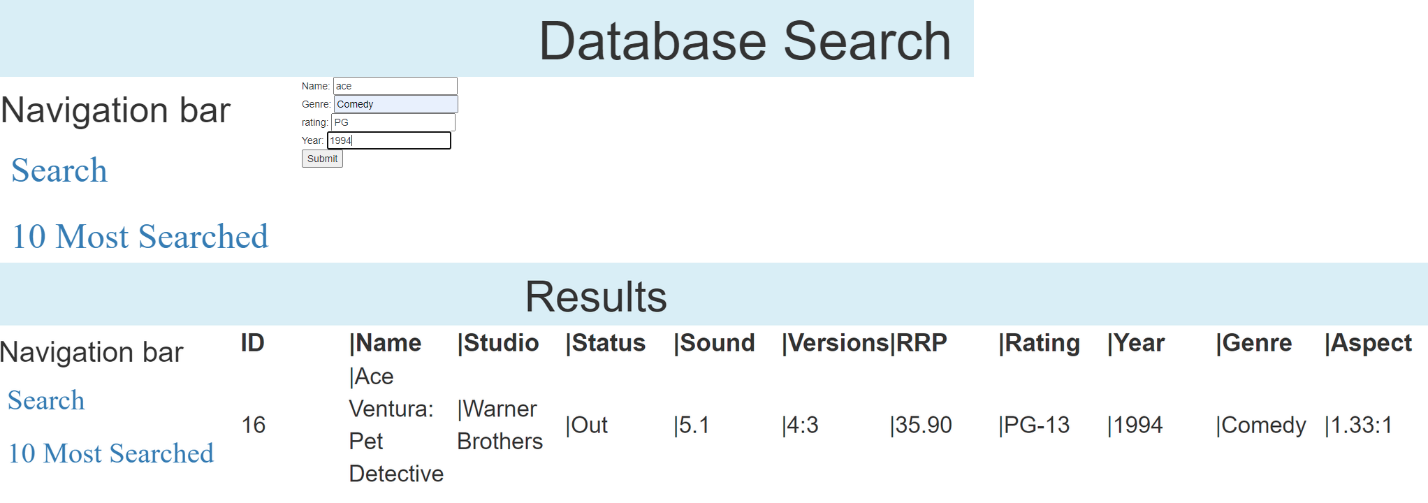


Figure 8- Name, Genre, Rating, Year

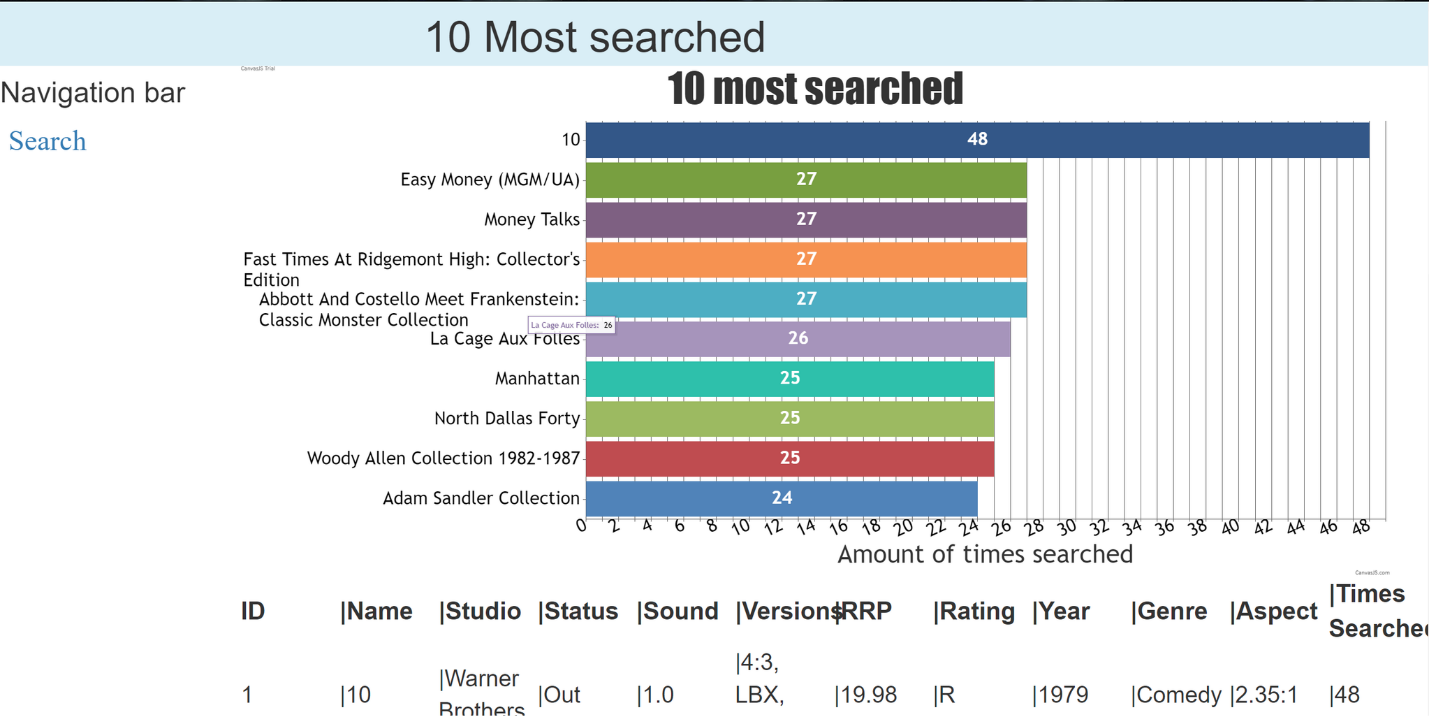


Figure 9- Top ten

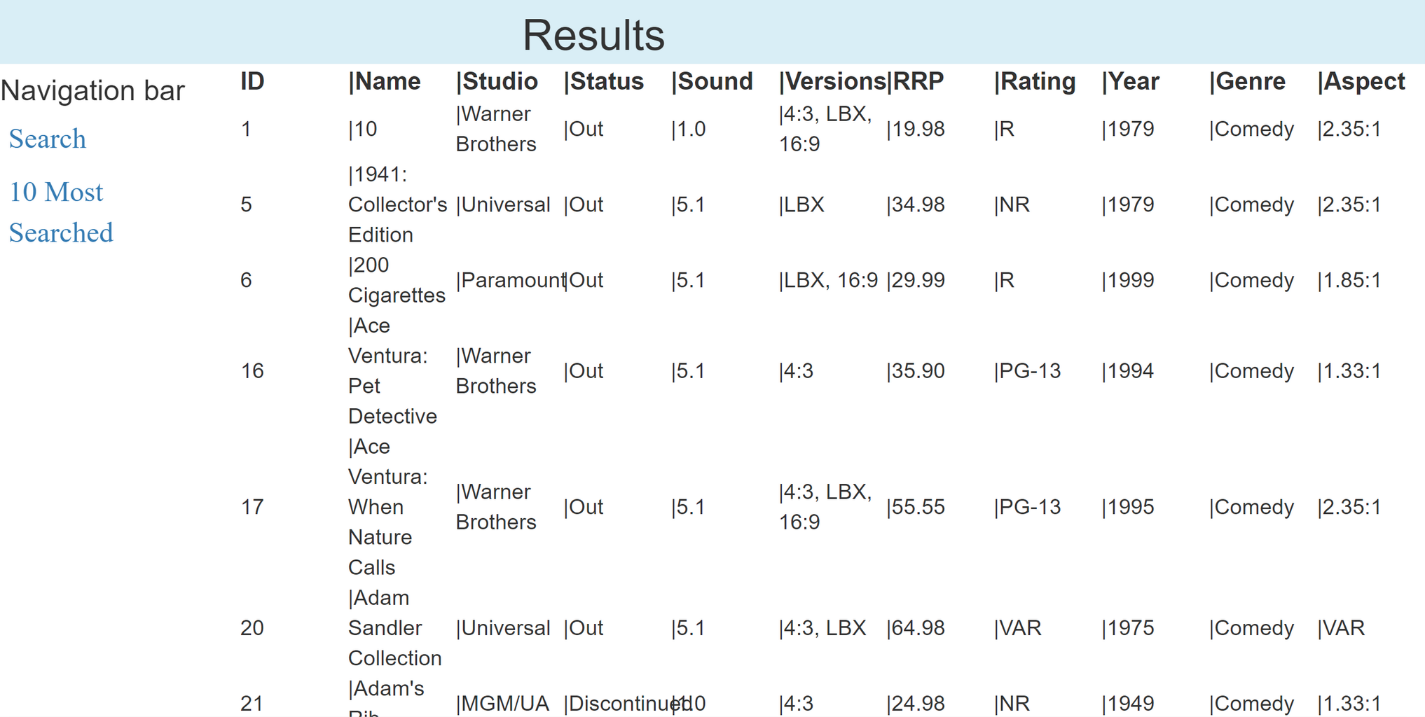


Figure 10- Runs on PC



Figure 11- Runs on iPad

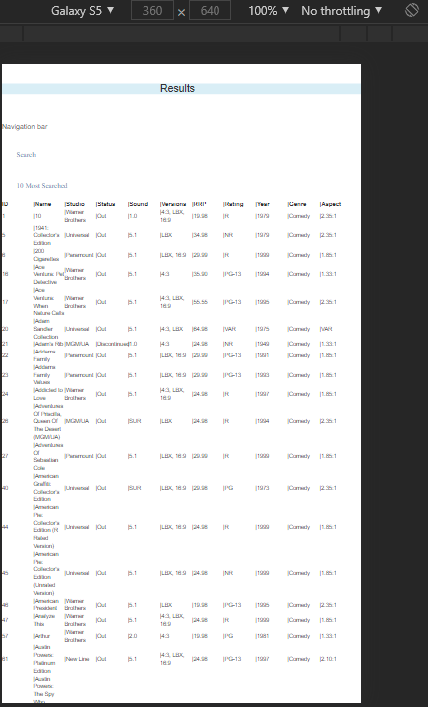


Figure 12- Runs on Galaxy S9

### Bug Triage 1

#### Meeting Minutes

27/05/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 03/06/2021, 9:00AM, Murdoch Campus

**1. Bug Report**

One bug identified and discussed. Database connection broken, in need of modification to fit new server.

**2. Triage (Bug Priority)**

Sole bug takes top priority by default.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **DESCRIPTION** | **SEVERITY** | **FREQUENCY** | **RISK** | **PRIORITY** |
| 1 | Database connection fails | **HIGH** | **HIGH** | **HIGH** | **HIGH** |

Table 1- Bug Report 27/05/2021

### Suspension Criteria and Resumption Requirements

#### Suspension Criteria

Testing will be suspended if:

1. Any vital functionality is broken
2. An error directly prevents further testing
3. Error(s) considered too numerous or too frequent
4. An error is discovered that renders further testing redundant

#### Resumption Requirements

Testing will resume when:

1. Vital functions repaired
2. Errors preventing testing are resolved
3. Errors are reduced to an acceptable level
4. Testing is no longer obstructed by dependencies

### Test Completeness

The criteria to determine that testing is complete, are as follows:

1. All test cases carried out successfully
2. All bugs fixed
   1. OR mitigated, with plans for full resolution in next iteration

### Test Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **SPRINT ONE** | **SPRINT TWO** | **SPRINT THREE** | **HANDOVER** |
| Test Strategy | Test Cases | Test Cases | Client Signoff |
| Test Cases | Bug Report | Bug Report |  |
| Bug Report | Bug Triage | Bug Triage |  |
| Bug Triage |  |  |  |

Table 2- Timeline of Test Deliverables

## Resource and Environment Needs

### Testing Tools

1. Linux Server
2. Movie Database (Populated)

### Test Environment

PC REQUIREMENTS

* **Hardware**
  + Intel Pentium 4 CPU or later
  + 4GB RAM
  + Networking capabilities (LAN, Wi-Fi)
* **Software**
  + OS: Windows 7 or later
  + Any leading web browser

MOBILE DEVICE REQUIREMENTS

* Android 5.0 Lollipop or later

## Glossary

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| Bug | An error within a computer program that causes undesired or unexpected actions |
| GitHub | Online service providing version control for software development |
| LAN | Local Area Network |
| OS | Operating System; software which gives computers a user friendly interface, and acts as a foundation for running programs |
| RAM | Random Access Memory. Programs are loaded into this short term storage, allowing fast access to any part of the program |
| Wi-Fi | Wireless Local Area Network.  The name Wi-Fi has no further meaning, and was never officially a shortened form of "Wireless Fidelity” |

Sprint Two

# Source Control Snapshot(S2)

## GitHub Repository

GitHub Link:

<https://github.com/profiteroles/RAD>



Figure 13- GitHub repository

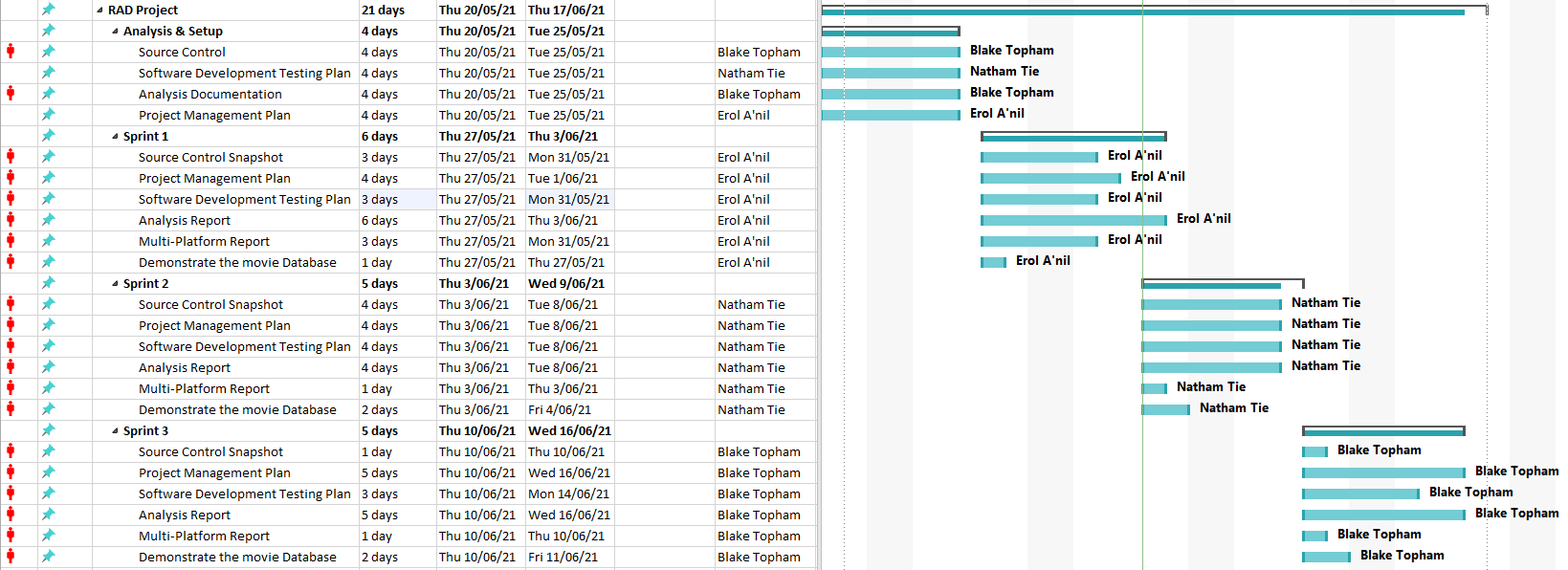


Figure 14- GANTT chart sprint two

# Performance Report(S2)

### Definition

Performance report is a documented assessment of performance and capacity of an application or system that is complex, time consuming and expensive to build. Performance test helps to reduce the risk of down time on multi-user interface by conducting tests that use “load” to reveal errors and limitations in the application.

The steps in a performance test are:

* Discovery
* Modelling
* Developing Scripts
* Execution of tests

### Goal

1. To clarify metrics and factors that the pages operate on
2. State of assumptions
3. Process description
4. Improvement opportunities

### Assumptions

These assumptions should be revised by the team closely related to the business and specific part of the application.

Software and Hardware

* CPU
* Network Connection
* Hard Drive
* Memory
* Version of Operating System
* Version of Software
  + Web Server
  + Database
  + Applications Server

### Process Description

All the steps in performance testing matter in making good decisions to make a project successful. These steps include, but aren’t limited to:

* Discovery
* Modelling
* Developing Scripts
* Executing Test

### Optimizer Tools

Possible tools for code optimisation/performance improvement would include:

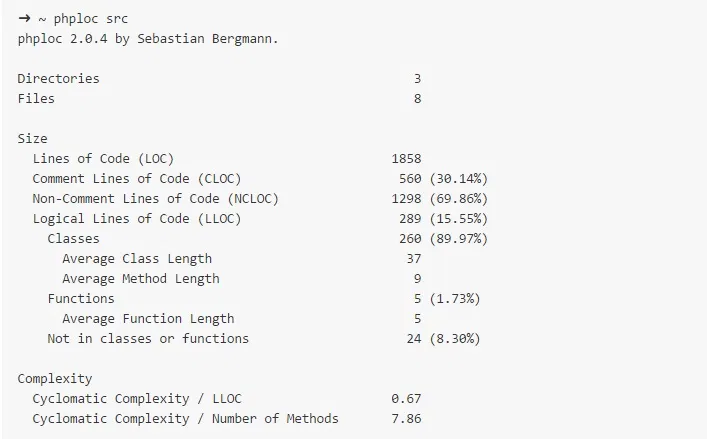
#### PINBA

A MySQL storage engine that monitors the performance of PHP code and detects bottlenecks in real time. Statistics are displayed in user-friendly, human language. Interface is read only.



### PHPLOC

A tool to measure how many lines of code in a PHP file as well as number of classes, files, etc. Ability to generate retroactive reports in CSV format.



## Data Need Analysis

Load testing tools have graphic capability. Graphs are just tools but not an actual report, however graphical data aids visualization to guide the stakeholders in consuming actionable information. Learning pattern recognition can take years to acquire where the need exists to recognize the system performance changes after certain load is surpassed, understanding of the limiting resources etc, is an ongoing and changing process.

A performance tester does the following duties:

* Form hypothesis
* Draw tentative conclusions
* Collect information to determine the information needed for the above
* Prepare key visualizations that provide insight into the performance and bottlenecks
* Support report narratives

To perform the above duties a thorough understanding of the following is very important:

* Architecture
* Hard and Soft Resources
* Garbage Collection Algorithms
* Database Performance
* Message Bus Characteristics
* Auxiliary Components of Systems – in complex systems

The full value of the performance test is unlocked when there is collective information from Developers, Operations, database analysts, help desk techs, business stakeholders and all teammates. Few effective steps to successfully achieve this are:

### Collecting

To weigh up the validity of the performance results its essential to gather-

* + Errors and type of errors
  + Pattern of errors
  + Obtaining error logs from the application

Measurements from every few seconds helps to understand the granularity of the application and help us to spot the trends and transient conditions in application development and testing.

### Aggregating

Measurements with statistics like scatter plots, graphs, data ranges, variance studies to study the data distribution aids in making the report more accurate. Using various levels of granularity provide isolated to collective views of the performance with compared with consistent granularities. This can be an improvement strategy standard.

### Visualizing

Comparison studies are done by using key graphical indicators to help us understand what is happening during the testing of the application.

List of the comparative visualizations are

* Check for validity of results by studying **Load data vs Error**
* Recognizing bottlenecks by checking **Load vs bandwidth throughput**
* Study scaling and scaling behavior by studying **Load Vs Response time**
* **Capacity of Infrastructure** i.e.system resources adequacy is determined by
  + Load vs Server CPU
  + JVM heap Memory
  + Input/output latency
  + Database lock contention

### Interpreting

Evaluating the data and drawing conclusions from hypothesis can be done by

* Quantitative observations – What can be mainly observed in the data?
* Comparing the observations – Where are the consistencies and inconsistencies?
* Developing hypothesis based on observations
* Testing the hypothesis
* Conclude from the hypothesis by validating

### Analyzing

Deciding on actions to be taken by checking if the objectives are met, then determining remediation options at business level, applications level, system requirements and network level, then retest.

Transparency in costs, benefits, and risk is essential, them must be specific and actionable at technical level or at the management level.

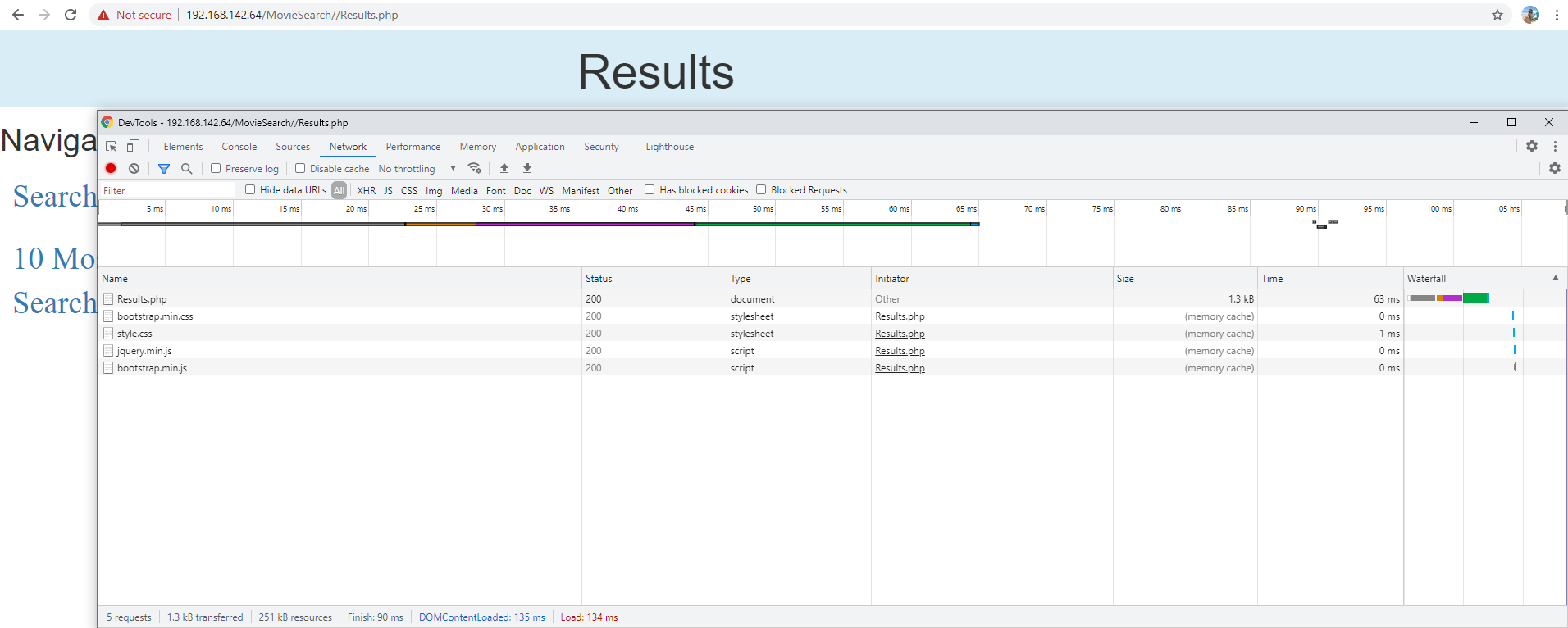
### Reporting

This is done by aggregating and presenting the risks, costs, limitations and recommendations to the stakeholders’ terms in a short elevator summary or a brief narrative. The report has the following sections:

* Executive summary
* Supporting detail
* Documents associated with the test
* Presentation

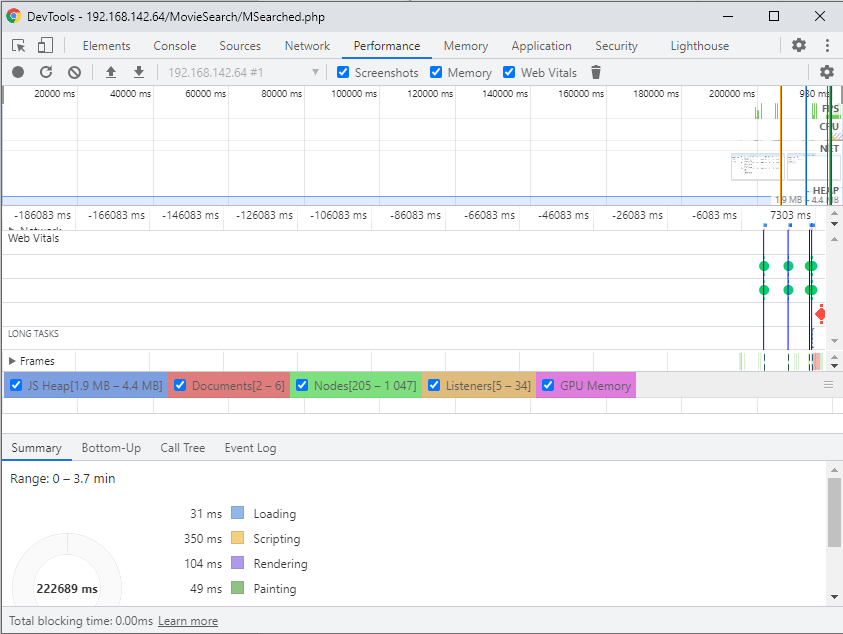
## Performance of Results

Response that we get within 65ms.



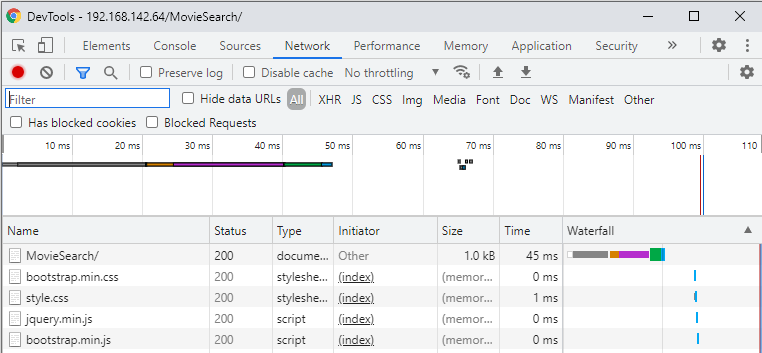
## Performance of Top 10 Movies

The completion of top 10 movies takes up to 980ms which includes the loading of a graphical chart.



## Performance of Home Page

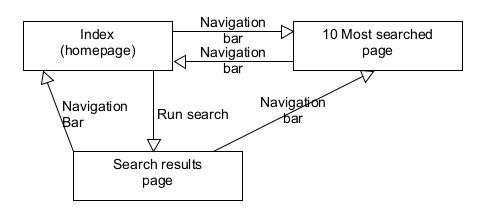
The Home(Index) page loads less than 50ms.



# Software Review Plan(S2)

## Current Architecture

The following navigation chart illustrates the current system structure of the project. The web application is comprised of three pages; Index, Search Results, and Top Ten. The chart demonstrates the relationships and pathways, including their direction.



The Index page is an HTML page which serves as a hub, providing text input fields and buttons. The buttons directly link to each of the two other pages, passing user input data using a “POST” method.

The Results page contains a connection script to the associated database. It generates SQL queries based on search terms posted from the index page, and constructs a table to display all matching entries in the database. A user-friendly message is displayed if no matching results are found.

The Ten Most-Searched Movies page is similar to the search results page. The page contains its own connection script and generates an SQL query to find and return the ten most-searched movies in the database. Movies are displayed in a table identical to that of the results page, and includes a bar chart displaying the titles and search count of each of the top ten movies.

## Software Quality Issues

Quality issues can be grouped in the following categories:

### Modularity

The two results pages (search results and top ten) contain **repeated code**; both of these pages have their own identical connection scripts.

### Portability

The database connection scripts contain **hard coded values** for host, username, password and database name. Given the scope of the project, this may be inconsequential but is worth noting.

### Efficiency

The code appears to contain some **redundant variables** and links, a few at most. Impact may be negligible. Further analysis required.

### Extensibility

In its current state, the codebase is quite small and simple. New additions would not be difficult to implement and integrate.

### Readability

Code is sufficiently commented in plain terms using minimal technical vocabulary. Appropriate indentation is utilized throughout. Code contains minimal nesting. Naming conventions are consistent and meaningful.

## Proposed Improvements

* Remove connection scripts from results pages. Create a dedicated database connection file to be referenced as needed, or place the connection script in a global function.
* Streamline code; check for redundant variables/links and remove if necessary
* Format code with auto-formatting tool

### Suggestions/Recommendations (Optional)

* Create optional form on homepage to override default values for database connection with user-defined values (host, user, password, etc.)
* Place code for results table in a global function

## Testing

Full program will be tested, including previously tested features to **monitor effect of changes** and ensure new additions have not broken existing modules. The testing methodology, strategy, and metrics are consistent with those detailed in the Sprint One testing plan.

### Scope

The new proposed features are detailed in the following revised testing scope:

#### In Scope

* User authentication
* Display database
* Search database
  + With any combination of parameters
* Top ten movies
* Operation on three different sized devices
  + PC
  + Tablet/Laptop
  + Phone

**New features to be tested:**

* User subscription
* Unsubscribe option
* Email user with:
  + Newsletter
  + Newsflash updates
* Admin portal
  + Receives email when user unsubscribes
  + Function to remove user details from database
* User experience (NON FUNCTIONAL)
  + Speed of operation to be reviewed in **Performance Report**

#### Out of Scope

* Login security
* User account recovery
* Thumbnails with theatrical posters/cover art
* User reviews

### Test Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Report S2 T1 | | | | | | | | |
| Project Name: RAD Sprint Two | | | Test Type: Black Box | | | | | |
| Description: Movie Search Application | | | Date: 03/06/2021 | | | | | |
| Developer(s): Blake Topham, Erol Anil, Nathan Tai | | | Tester: Blake Topham | | | | | |
| Test Case # | Test Case Name | Test Steps | | Test Data | | Expected Results | Evidence | Pass? |
| 1 | Search by Name | Enter search terms | | Search term | | Matching results displayed | Figure 15 |  |
| 2 | Search by Name, Genre | Enter search terms | | Search term | | Matching results displayed | Figure 16 |  |
| 3 | Search by Name, Genre, Rating | Enter search terms | | Search term | | Matching results displayed | Figure 17 |  |
| 4 | Search by Name, Genre, Rating, Year | Enter search terms | | Search term | | Matching results displayed | Figure 18 |  |
| 5 | Top Ten Movies | Click “10 Most Searched” button | | 10 movies with highest search count | | 10 movies with highest search count displayed | Figure 19 |  |
| 6 | Runs on: Desktop PC | Run page in desktop browser | | Page elements | | Page runs | Figure 20 |  |
| 7 | Runs on:  iPad | Run page in device simulation mode | | Page elements | | Page runs | Figure 21 |  |
| 8 | Runs on: Galaxy S5 | Run page in device simulation mode | | Page elements | | Page runs | Figure 22 |  |
| 9 | User subscribes | Enter email and click subscribe | | Email address | | User subscribed, receives newsletter/newsflash | Figure 23 |  |
| 10 | User unsubscribes | Enter email and click unsubscribe | | Email address | | Email sent to Admin account for database removal | Figure 24 |  |
| 11 | Admin removes user | Remove user entry from database | | User details(email, etc.) | | User entry removed from database | Figure 25 |  |
| Test Completeness Criteria Achieved:  1. All test cases carried out successfully  2. All bugs fixed | | | | | Suspension Criteria Met:  1. Vital functionality broken  2. Error directly prevents further testing  3. Errors too many or too often  4. Error makes further testing redundant | | | |
| Testing Complete | | | | | Testing Suspended | | | |

### Screenshots

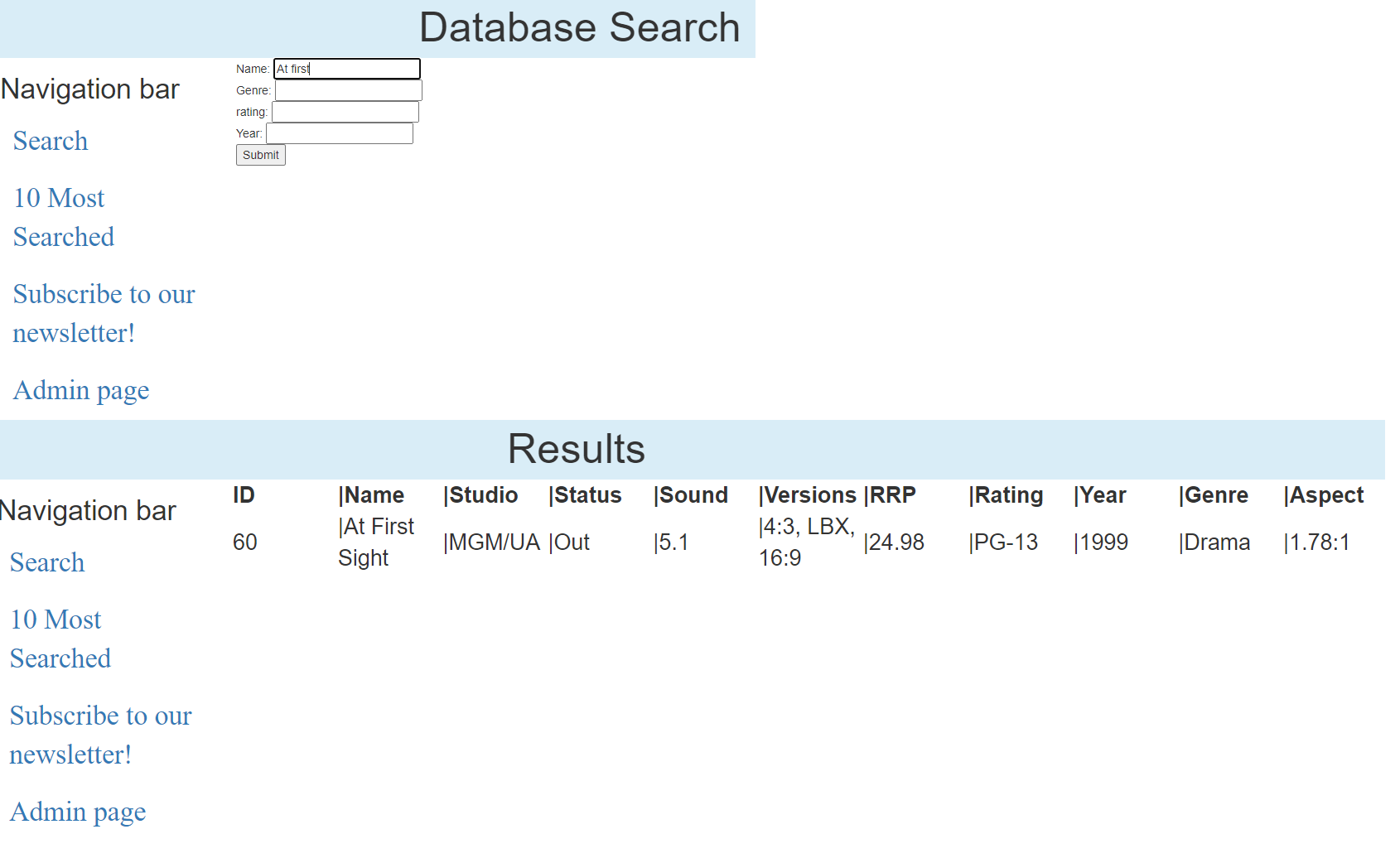


Figure 15- Search by name

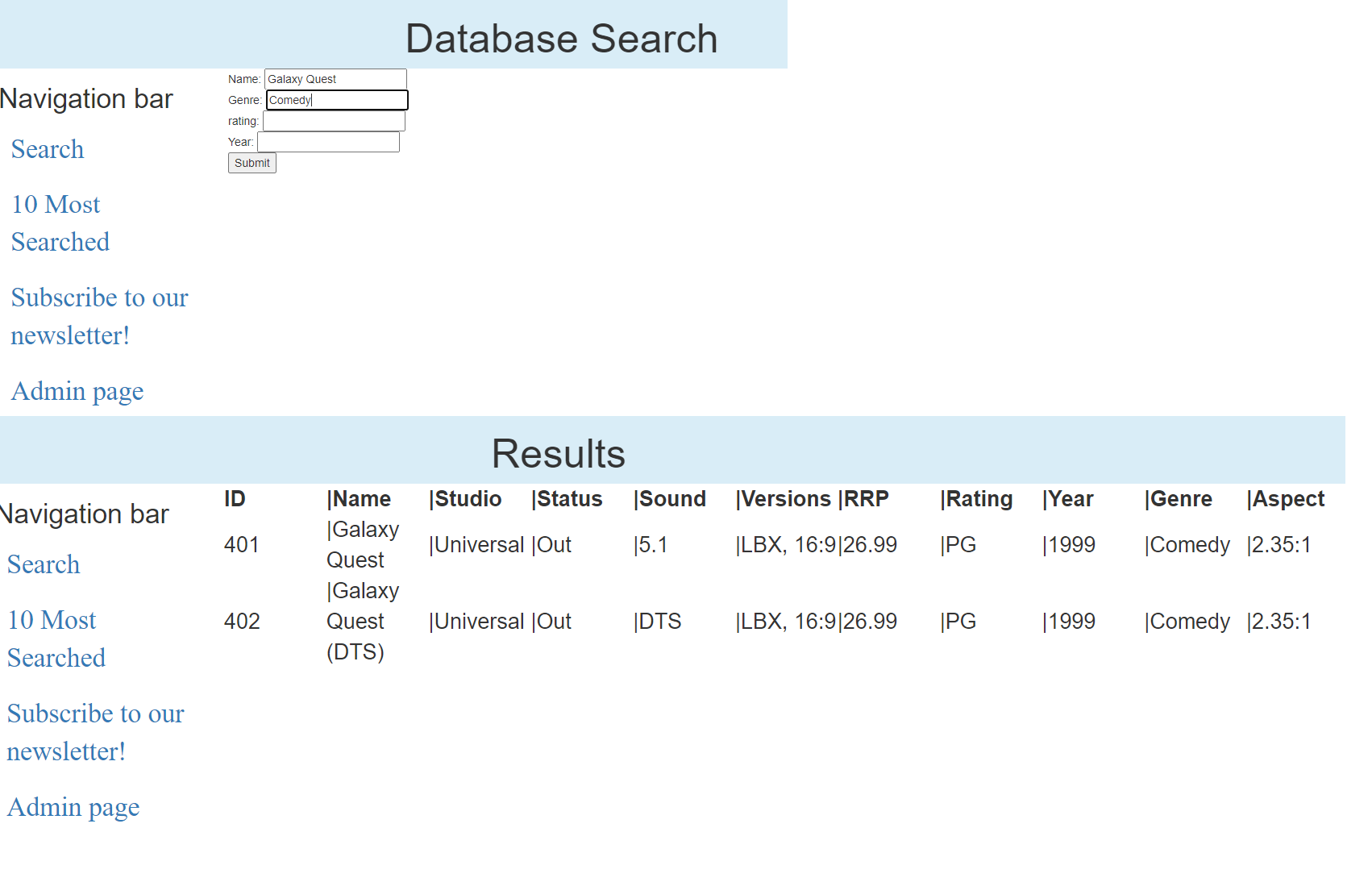


Figure 16- Name, Genre

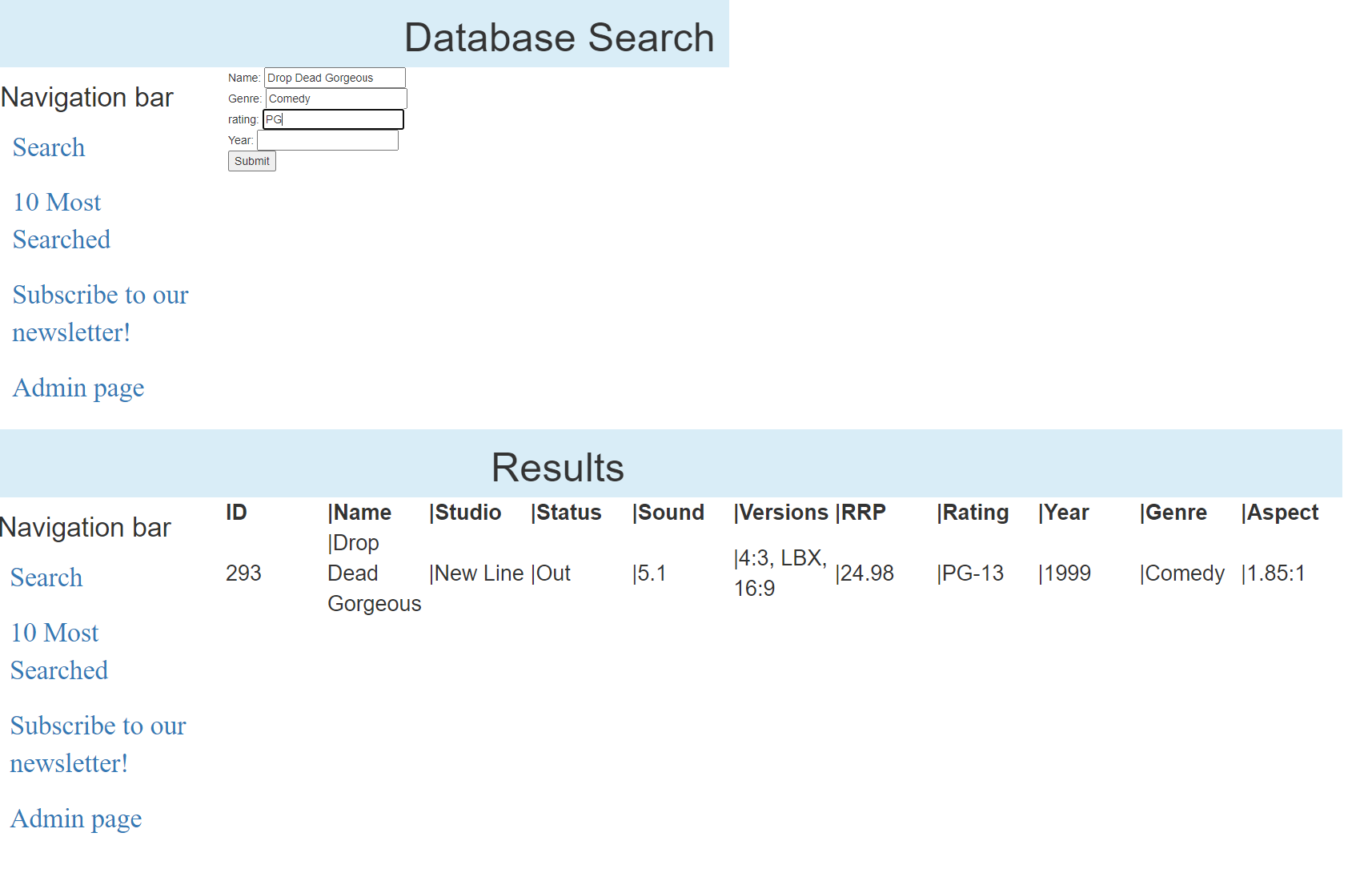


Figure 17-Name, Genre, Rating

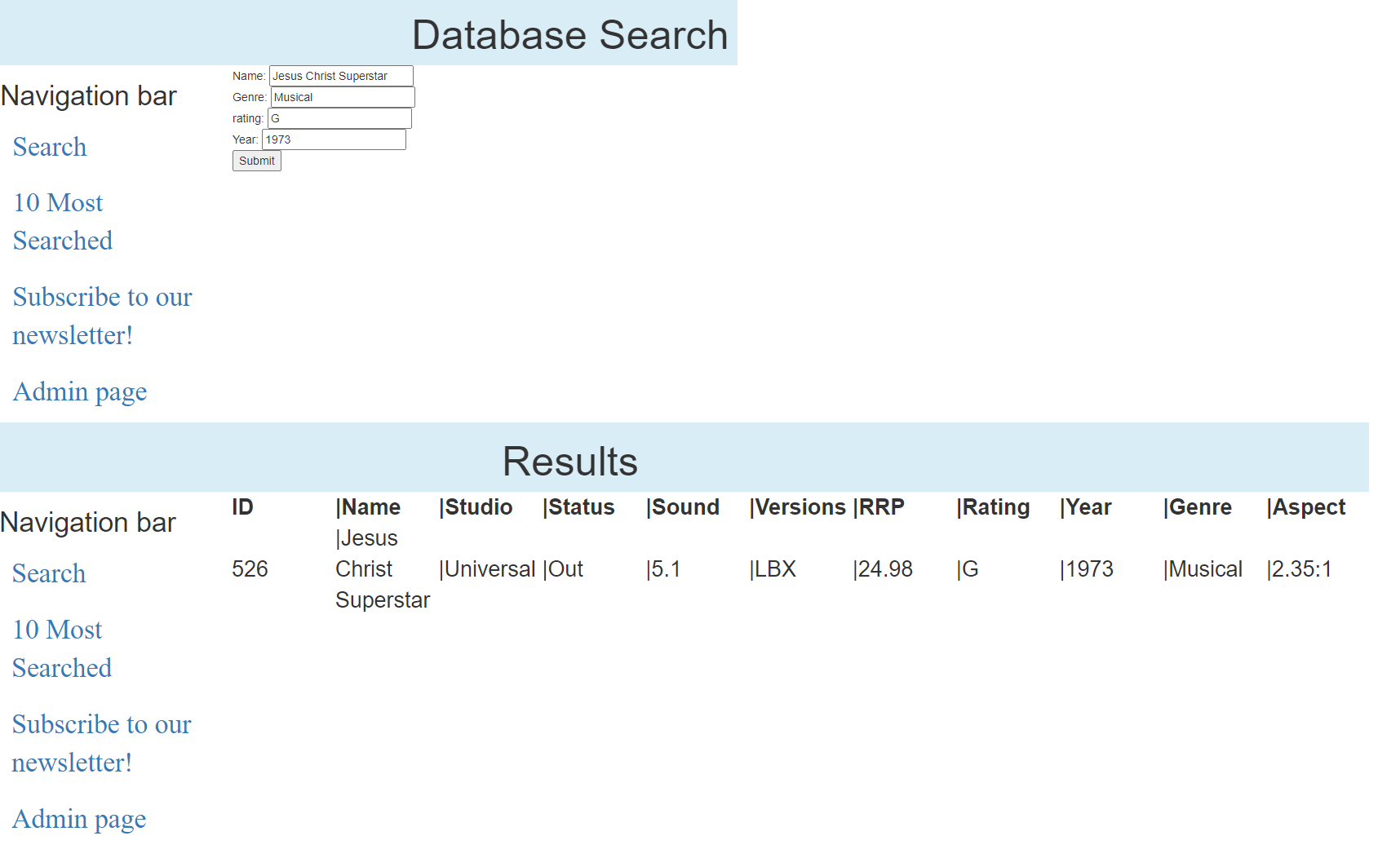


Figure 18- Name, Genre, Rating, Year

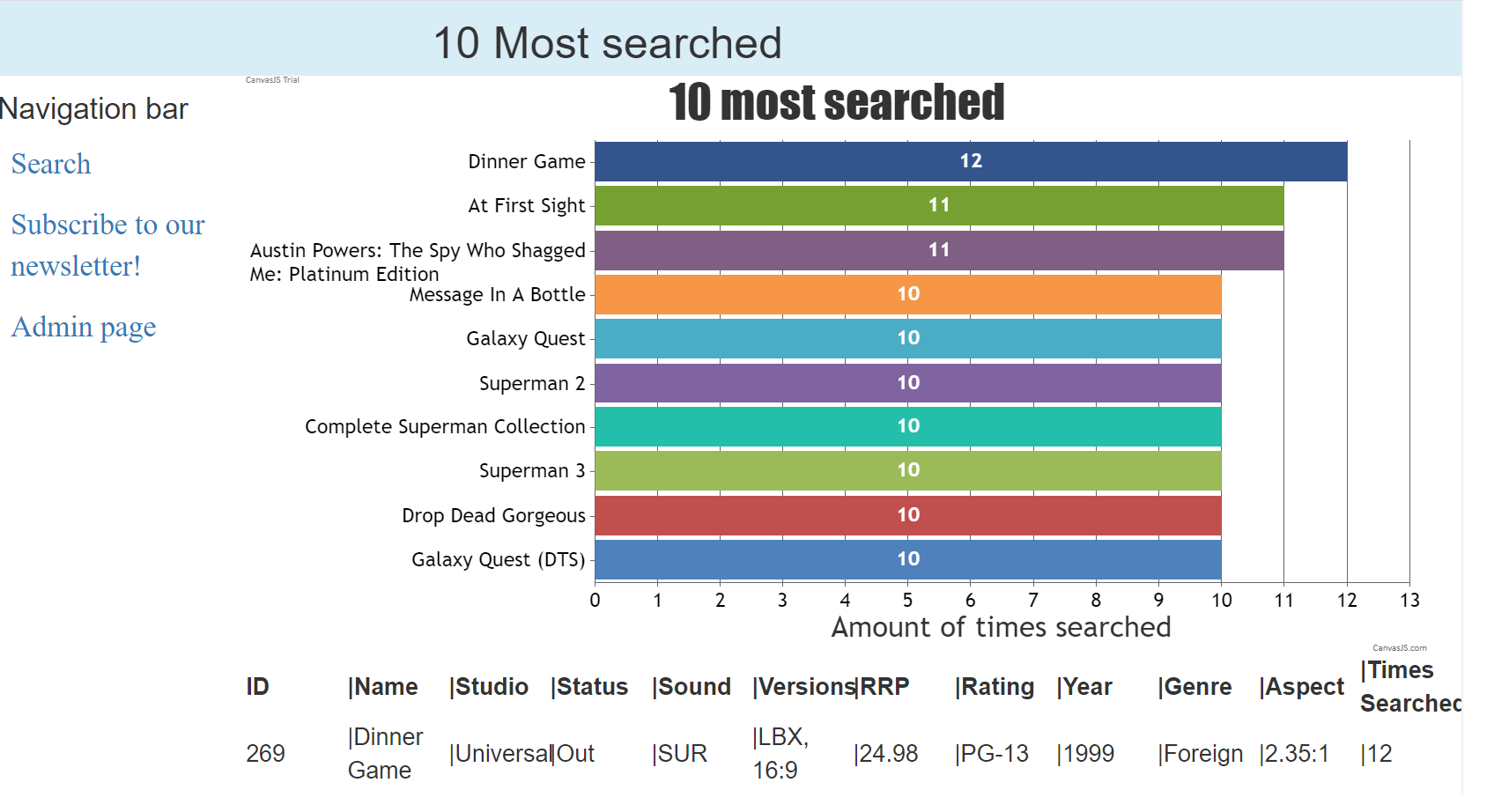


Figure 19- Top ten

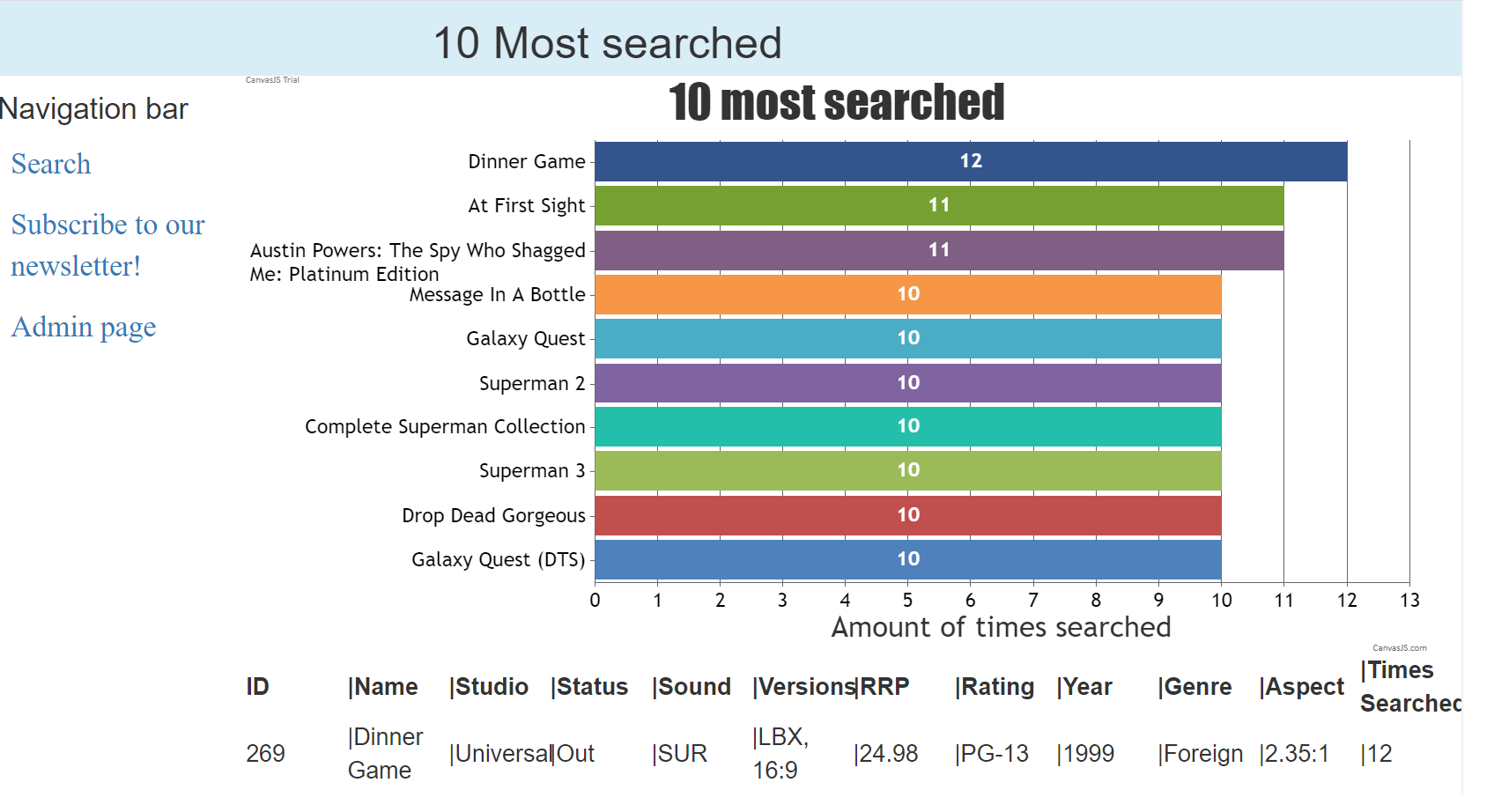


Figure 20- Runs on PC

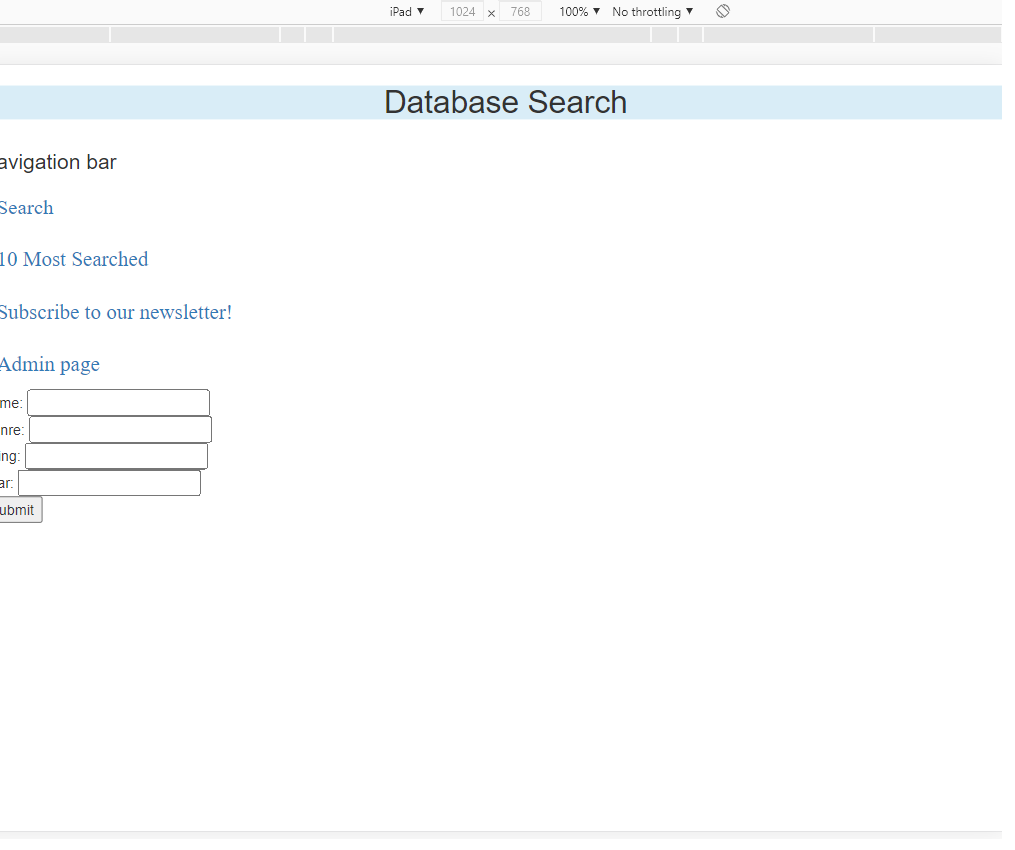


Figure 21- Runs on iPad

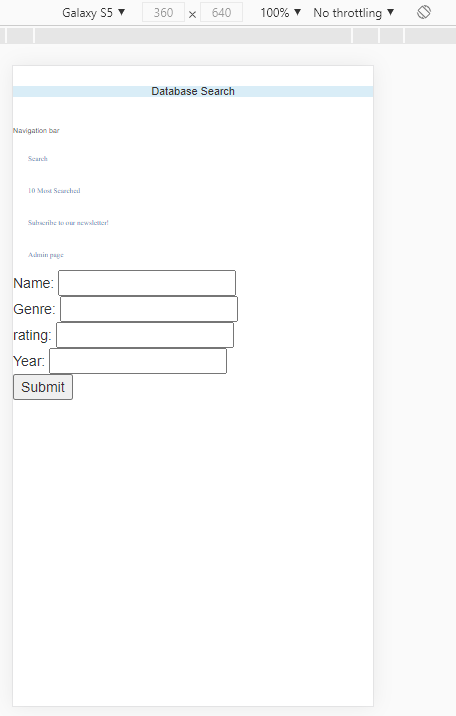


Figure 22- Runs on S5

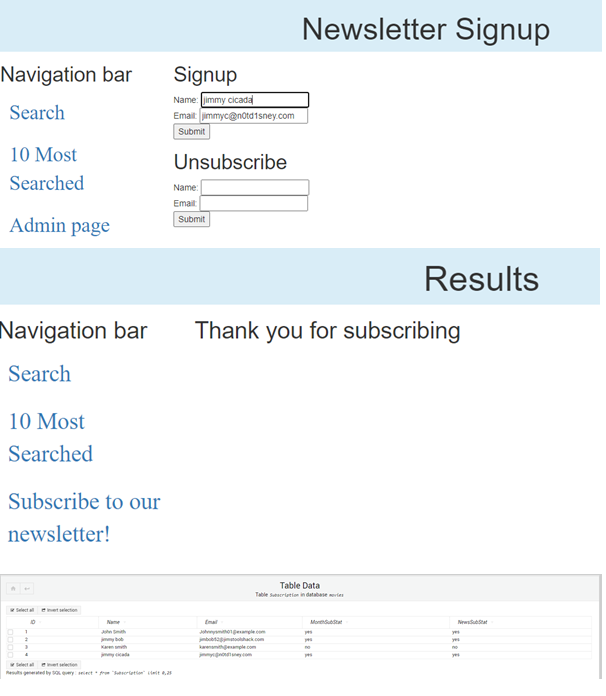


Figure 23- User unsubscribes, database not yet reflected

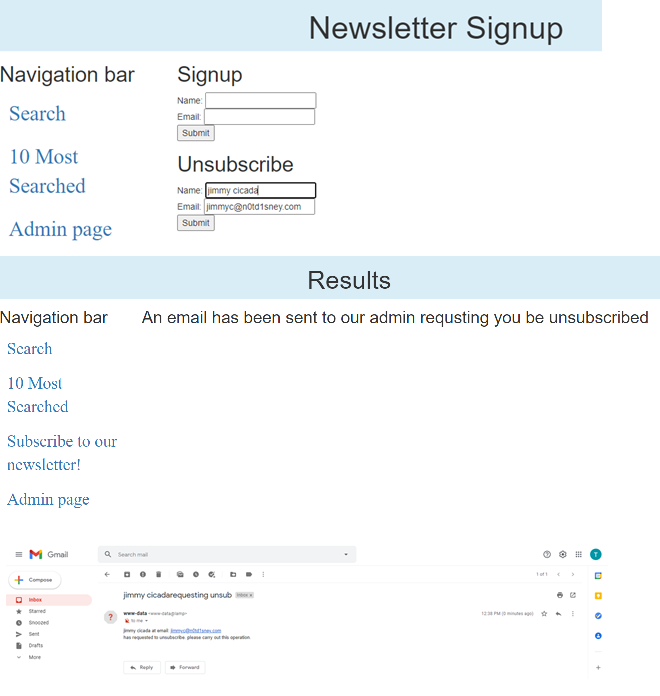


Figure 24- User unsubscribes, admin email

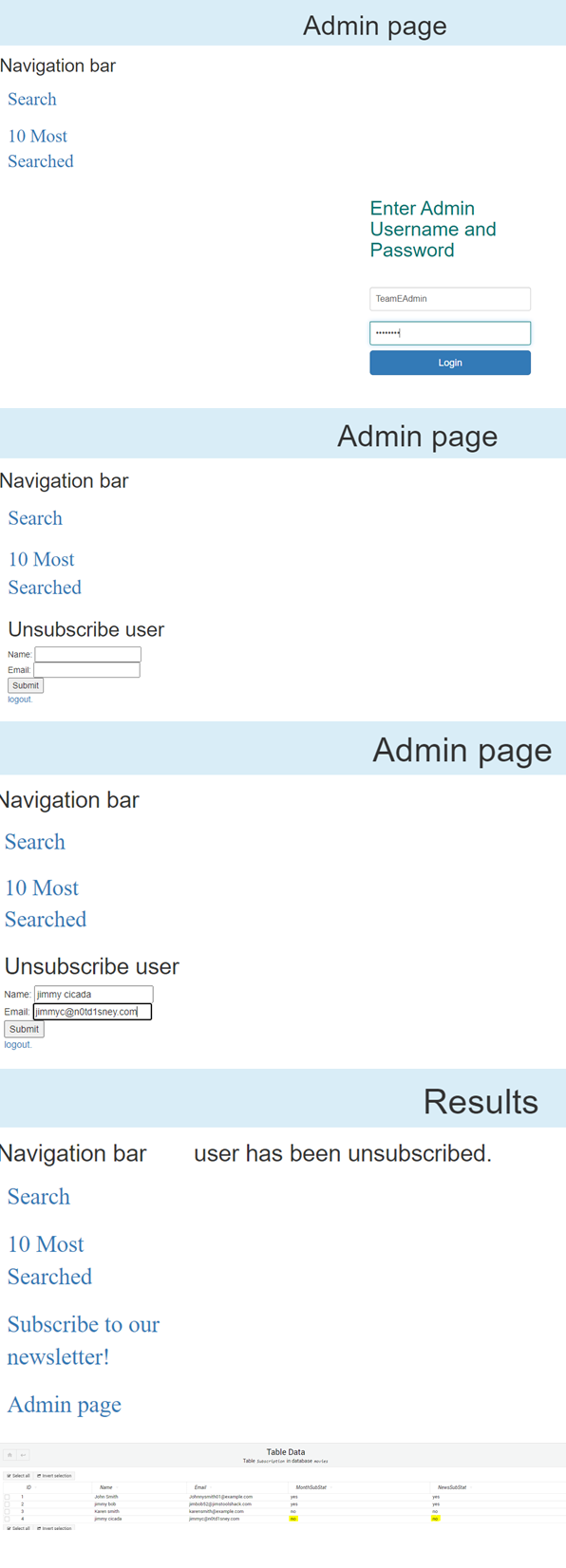


Figure 25- Admin unsubscribes user

### Bug Triage 2

#### Meeting Minutes

08/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:00AM, Murdoch Campus

**1. Bug Report**

One issue identified and discussed. Not necessarily a bug. Unsubscribe emails going straight to spam folder.

**2. Triage (Bug Priority)**

Sole issue takes top priority by default.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **DESCRIPTION** | **SEVERITY** | **FREQUENCY** | **RISK** | **PRIORITY** |
| 2 | Unsubscribe emails going straight to spam folder | **LOW** | **HIGH** | **LOW** | **HIGH** |

### Team Meeting 2

#### Meeting Minutes

03/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:00AM, Murdoch Campus

**1. Team Role Assignment**

**ANIL- Configuration Manager**

Source Control Snapshot

* Update GitHub
* Create screenshots, add to master document

General Version Control

* Manage repositories

Performance Report

GANTT Chart for SPRINT TWO (week seven)

* Timeline
* Include who does what

**BLAKE- Developer**

Development

* Admin page
* Database tables for:
  + Email addresses
  + Login details if required
* Unsubscribe feature
  + Sends email to admin account
  + Do not allow user to directly manipulate database
  + Admin account required to remove database entries
* Separate options for user to select:
  + Monthly newsletter
  + Newsflash notification email

**NATHAN- Scrum Master**

Project Management Plan

* Discuss roles
* Record team meeting minutes
* Allocate tasks
* Produce documentation detailing task allocations

Software Review Plan

* Update requirements based on client meeting
* Record minutes of all meetings (client, task allocations, bug triage)
* Review application against new requirements
* Update testing plan to include new features

Demonstrate the Movie Database Application

* Read and understand all documentation
* Read and understand all code
* Demonstrate application in 3 different sized device environments

### Client Meeting 2

#### Meeting Minutes

03/06/2021

Present: Client, Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:30AM, Murdoch Campus

**1. Review Requirements**

* Project requirements discussed and documented.
* Project planning commenced.
* Additions requested.
  + Subscription feature, choice of:
    - Monthly newsletter
    - Newsflash notifications
* Requirements noted and agreed.

**2. User Experience**

* Performance review requested
  + Speed
  + Usability

Sprint Three

# Source Control Snapshot(S3)

Link: <https://github.com/profiteroles/RAD>

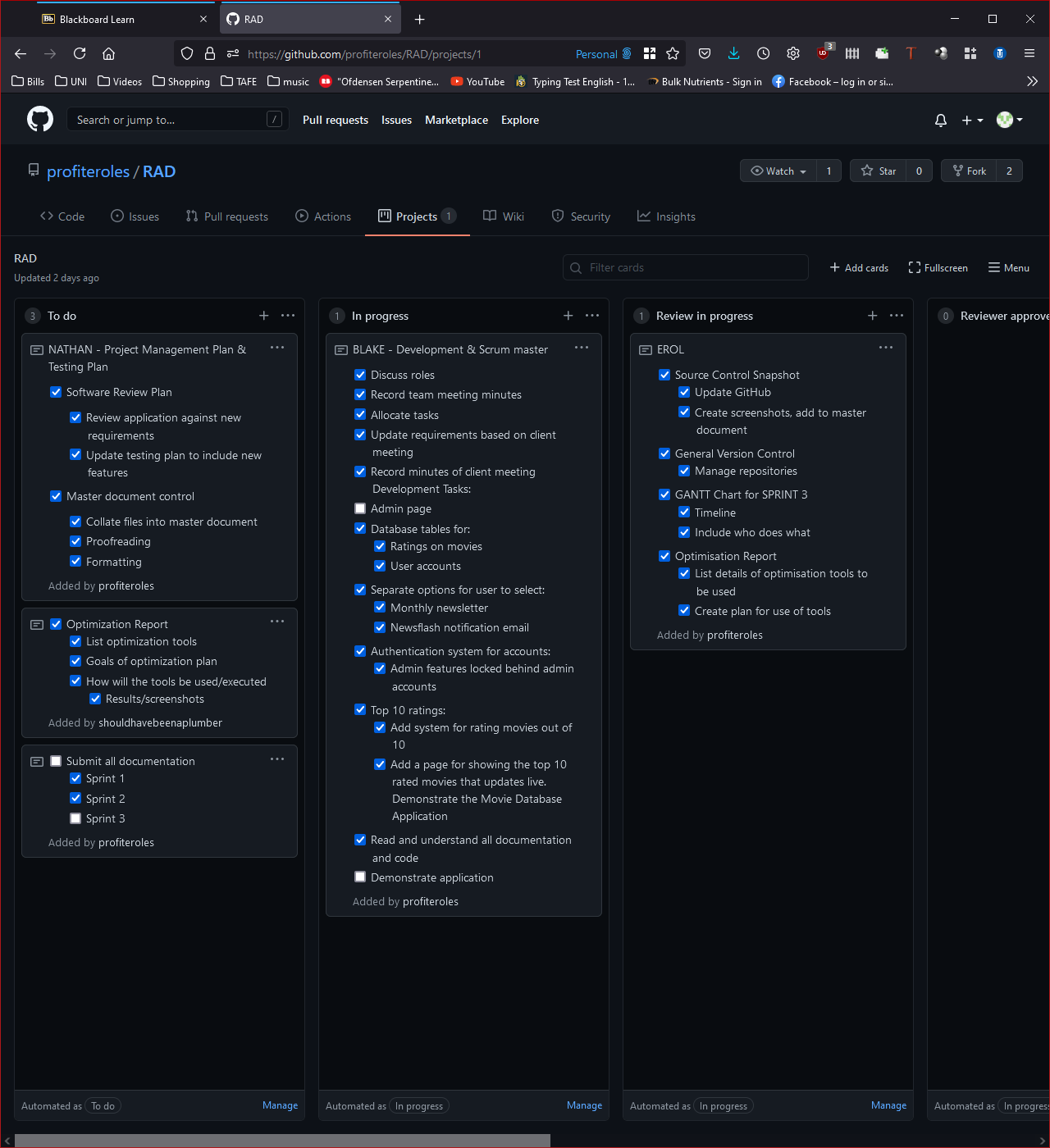


Figure 26- GitHub screenshot

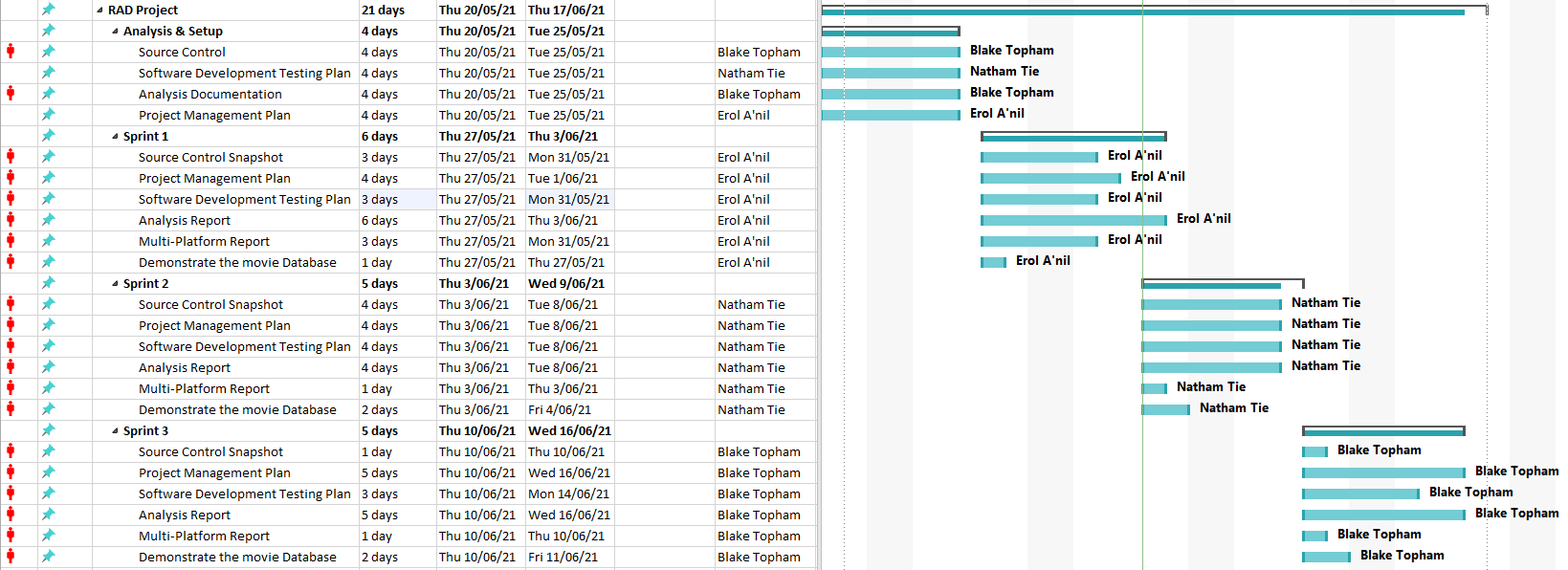


Figure 27- GANTT chart

# Optimization Report(S3)

## Description

Performance optimization measures the performance of the server-side scripts using various tool in different environments. It is performed by running the codes several times and observing the average execution time to create a benchmark.

## Need for Optimization

1. To make applications fast & effective.
2. It appeals the consumers: speedy campaigns imply higher selling speed

## Case Studies

* Bing found that searches that were 2 seconds slower resulted in a 4.3% drop in revenue per user.
* When Mozilla shaved 2.2 seconds off their landing page, Firefox downloads increased 15.4%.
* Shopzilla saw conversion rates increase 7-12% because of their web performance optimization efforts.
* Making Barack Obama’s website 60% faster increased donation conversions 14%

## Best Practices

* Using latest PHP – PHP 8.1.0 Alpha 1
* Using JSON data types instead of XML
* Using caching systems
* Closing the DB connection and limit the DB hit.

## Performance Tuning Steps Followed

To keep a check and eliminate slow loading pages, inaccessible pages, unresponsive links, and many other factors that would limit the applications downside and work fast and efficient for ACME Entertainment, below tuning tips were considered to monitor our coding performance:

1. **Bottlenecks** – Identifying impediments and finding the root cause
2. **Profiling** – What fits ACME Entertainment’s needs. Prefix was looked into as a profiling tool
3. **Code Optimization**- code was optimized so that it utilizes memory, executes more rapidly, and also performs fewer input and output operations
4. **Configure Optimization**- optimize the performance of your application, as well as ensure the reliability and cost-effectiveness of your applications system storage
5. **Distributed Computing** - Increase the potential for parallel execution
6. **Catching Strategy** - reduce the number of database operations and code compilation
7. **Load Balancing** - Load balancing is configured in the plugins using various techniques to prioritize by avoiding session objects (stored in local memory)
8. **Avoiding Client-side** - constantly check and remove or lessen the redirection of your pages
9. **Security** - responsible for processing users request in a web page or in the browser
10. **SQL** - dynamic management views (DMVs) which provides provide data about query stats, executing plans, and many more
11. **Content Delivery Network** - reduce latency, reduce bandwidth consumption, and secure apps while it can also block spammers and other that attack your system
12. **Error handling** - allows you to develop and improve error reporting that best suit the needs

## Performance Tuning Steps Followed

* Latest versions used
* Used double Quotes (“) and Single quotes (‘)
* Avoided Relative Paths in File Inclusion
* Releasing all Resources
* Avoided Unnecessary Use of Global Variables
* Never used Count or Any Other Methods in The Condition Section of a Loop
* Latest versions used
* Used double Quotes (“) and Single quotes (‘)
* Avoided Relative Paths in File Inclusion
* Releasing all Resources
* Avoided Unnecessary Use of Global Variables
* Never used Count or Any Other Methods in The Condition Section of a Loop
* Used ISSET
* Used More Static Methods/Properties
* foreach > for > while implementations

## Previous Observations

#### Performance Extracts

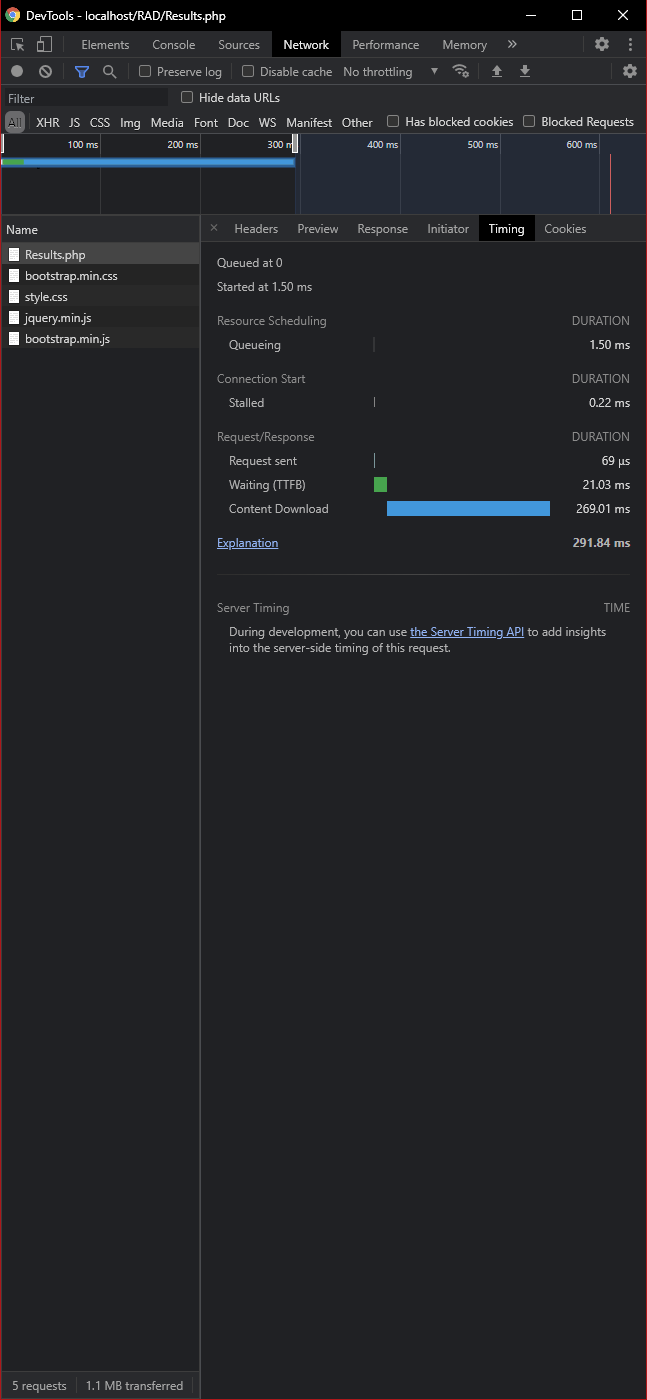
Table

Description automatically generated

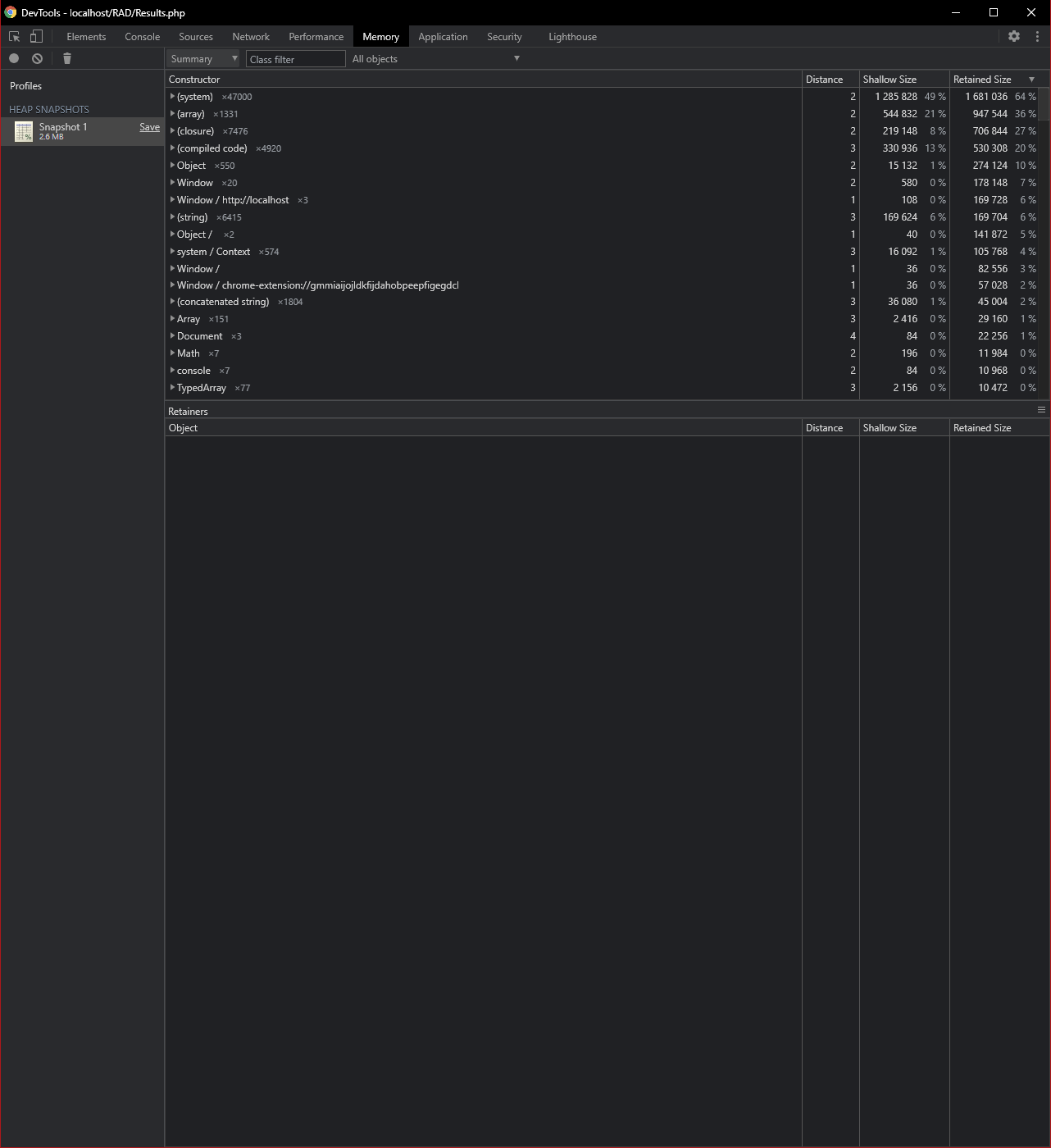
Graphical user interface, application, table

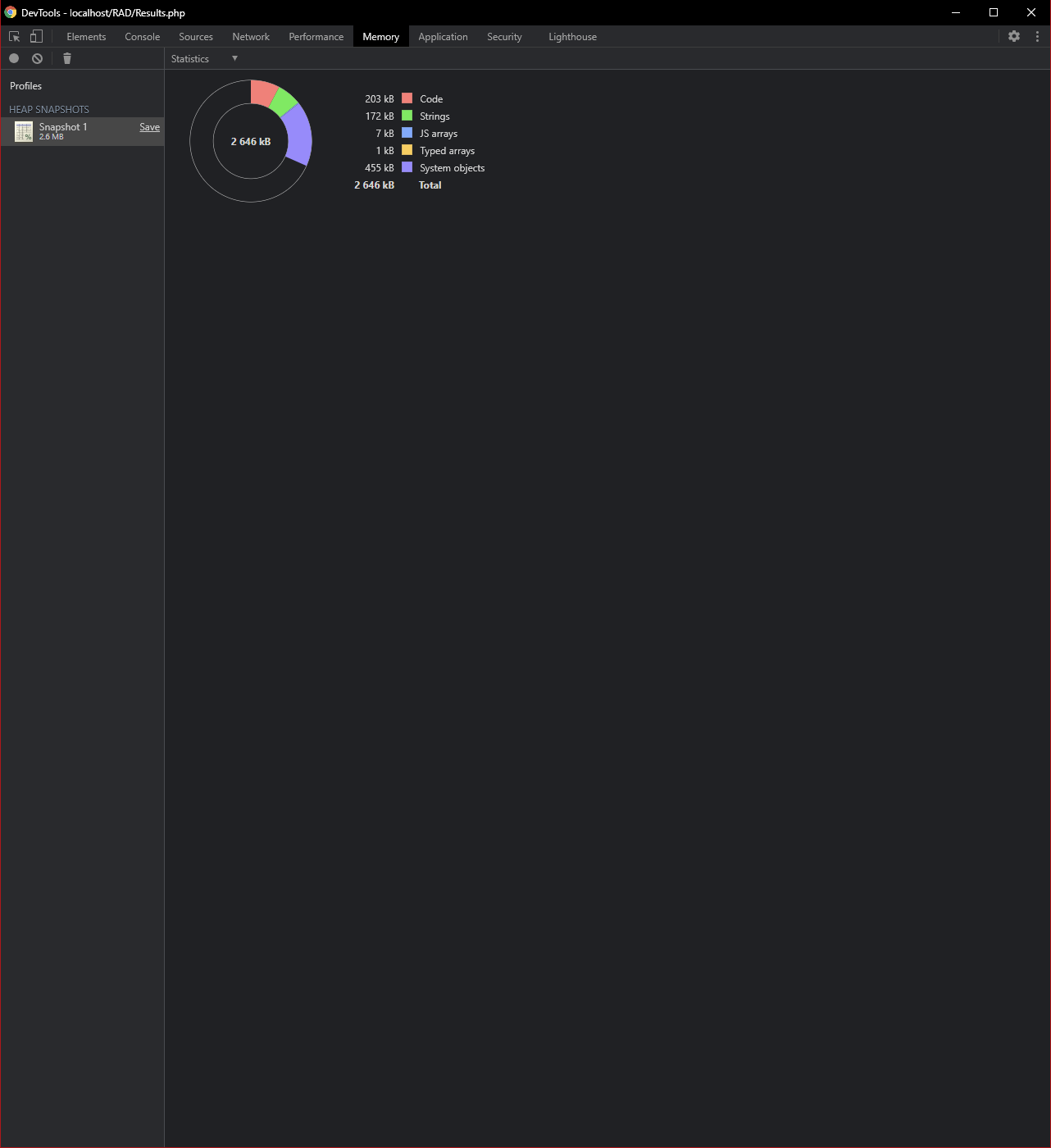
Description automatically generated

## Network Analytics

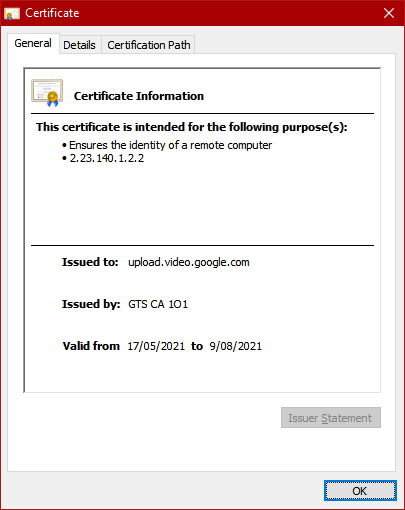


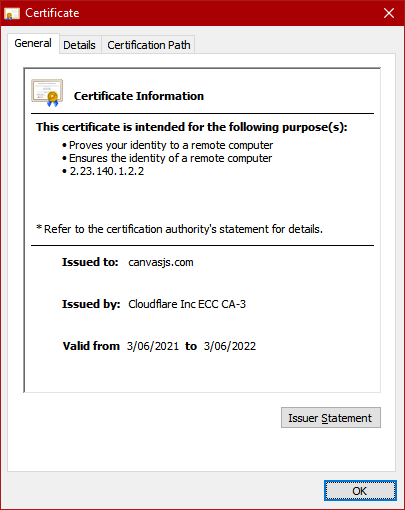
## Memory Status



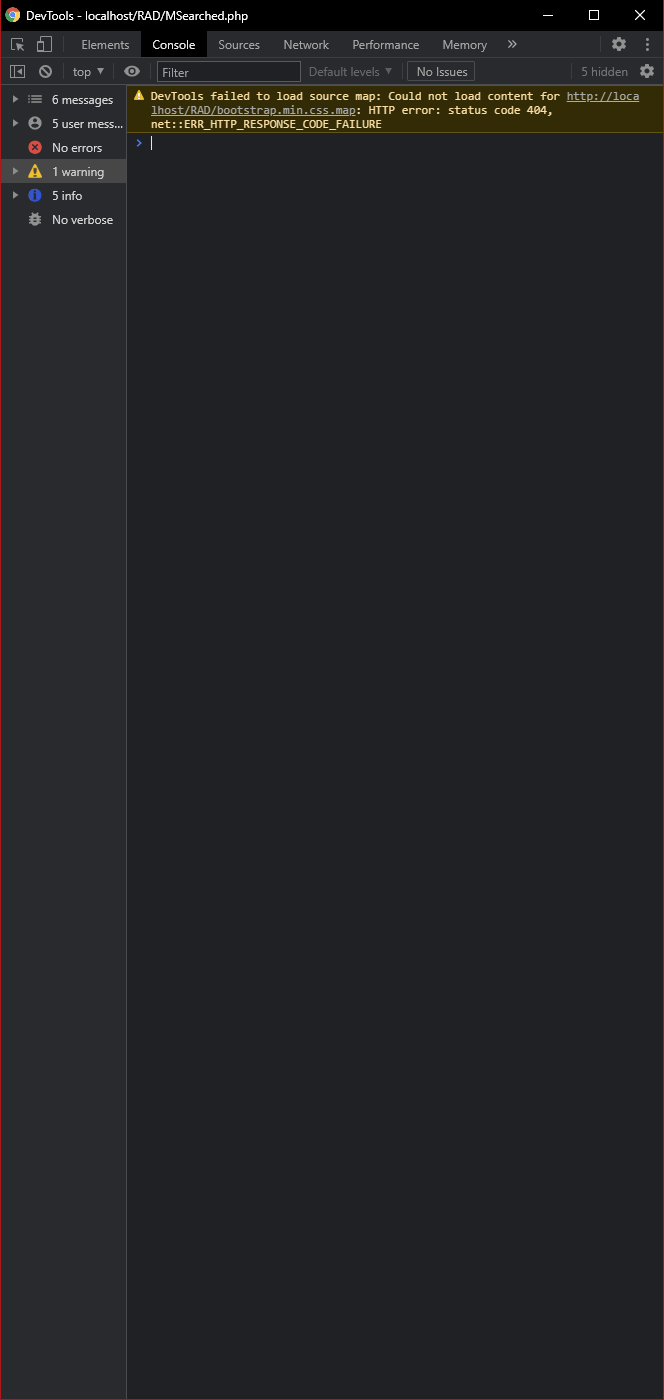


## Security

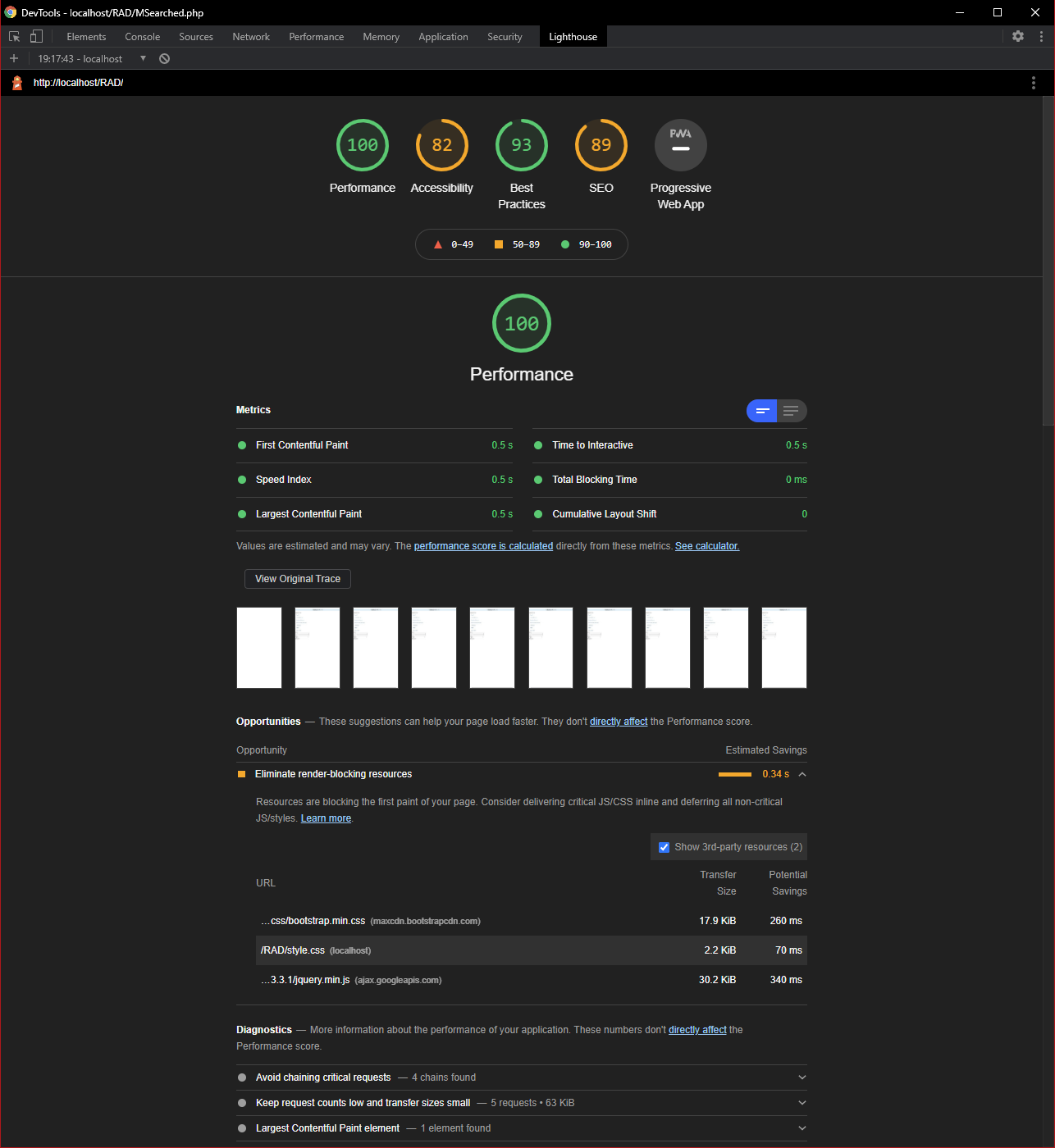




## Warnings / Error Logging



## Current Performance Extract Report



# Updated Software Testing Plan(S3)

## Reflections

Agile testing methodology has thus far been successful in achieving the quality objectives of the project. Increases in the project’s complexity have been well tolerated by the test methodology. Manual black box testing has been sufficient. Test methodology for the final sprint will remain unchanged. Testing will continue to focus primarily on key functionality from a user perspective.

## Testing Overview

Testing will be conducted on the program in its entirety, including all previously-tested features to **monitor effect of changes.** Full testing will ensure new features and fixes have not broken existing modules. The testing methodology, strategy, and metrics are consistent with those detailed in the Sprint One testing plan.

### Scope

The new proposed features are detailed alongside previously developed features in the following revised testing scope:

#### In Scope

* User authentication
* Display database
* Search database
  + With any combination of parameters
* Top ten movies
* Operation on three different sized devices
  + PC
  + Tablet/Laptop
  + Phone
* User subscription
* Unsubscribe option
* Email user with:
  + Newsletter
  + Newsflash updates

**New features to be tested for SPRINT THREE:**

* Admin page
* Authentication system for accounts
* Admin features locked behind admin accounts
* Top 10 ratings
* System for rating movies out of 10
* Users can enter ratings
* Page for showing top 10 rated movies that updates live

#### Out of Scope

* User account recovery
* Thumbnails with theatrical posters/cover art

## Test Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Report S3 T1 | | | | | | | | |
| Project Name: RAD Sprint Three | | | Test Type: White box | | | | | |
| Description: Movie Search Application | | | Date: 11/06/2021 | | | | | |
| Developer(s): Blake Topham, Erol Anil, Nathan Tai | | | Tester: Blake Topham | | | | | |
| Test Case # | Test Case Name | Test Steps | | Test Data | | Expected Results | Evidence | Pass? |
| 1 | Search by Name | Enter search terms | | Search term | | Matching results displayed | Figure 27 |  |
| 2 | Search by Name, Genre | Enter search terms | | Search term | | Matching results displayed | Figure 28 |  |
| 3 | Search by Name, Genre, Rating | Enter search terms | | Search term | | Matching results displayed | Figure 29 |  |
| 4 | Search by Name, Genre, Rating, Year | Enter search terms | | Search term | | Matching results displayed | Figure 30 |  |
| 5 | Top Ten Movies | Click “10 Most Searched” button | | 10 movies with highest search count | | 10 movies with highest search count displayed | Figure 31 |  |
| 6 | Runs on: Desktop PC | Run page in desktop browser | | Page elements | | Page runs | Figure 32 |  |
| 7 | Runs on:  iPad | Run page in device simulation mode | | Page elements | | Page runs | Figure 33 |  |
| 8 | Runs on: Galaxy S5 | Run page in device simulation mode | | Page elements | | Page runs | Figure 34 |  |
| 9 | User subscribes | Enter email and click subscribe | | Email address | | User subscribed, receives newsletter/newsflash | Figure 35 |  |
| 10 | User unsubscribes | Enter email and click unsubscribe | | Email address | | Email sent to Admin account for database removal | Figure 36 |  |
| 11 | Admin removes user | Run admin user unsubscribe command. | | User details(email, etc.) | | User entry set to unsubscribed | Figure 37 |  |
| 12 | Admin login | Enter admin details | | Admin login details | | Admin features enabled | Figure 38 |  |
| 13 | User adds rating | Enter rating as user | | Score | | Rating added to database | Figure 39 |  |
| 14 | Rating auto-updates on page | View any movie search results | | Score | | Rating auto-updates on page | Figure 40 |  |
| Test Completeness Criteria Achieved:  1. All test cases carried out successfully  2. All bugs fixed | | | | | Suspension Criteria Met:  1. Vital functionality broken  2. Error directly prevents further testing  3. Errors too many or too often  4. Error makes further testing redundant | | | |
| Testing Complete | | | | | Testing Suspended | | | |

### Screenshots

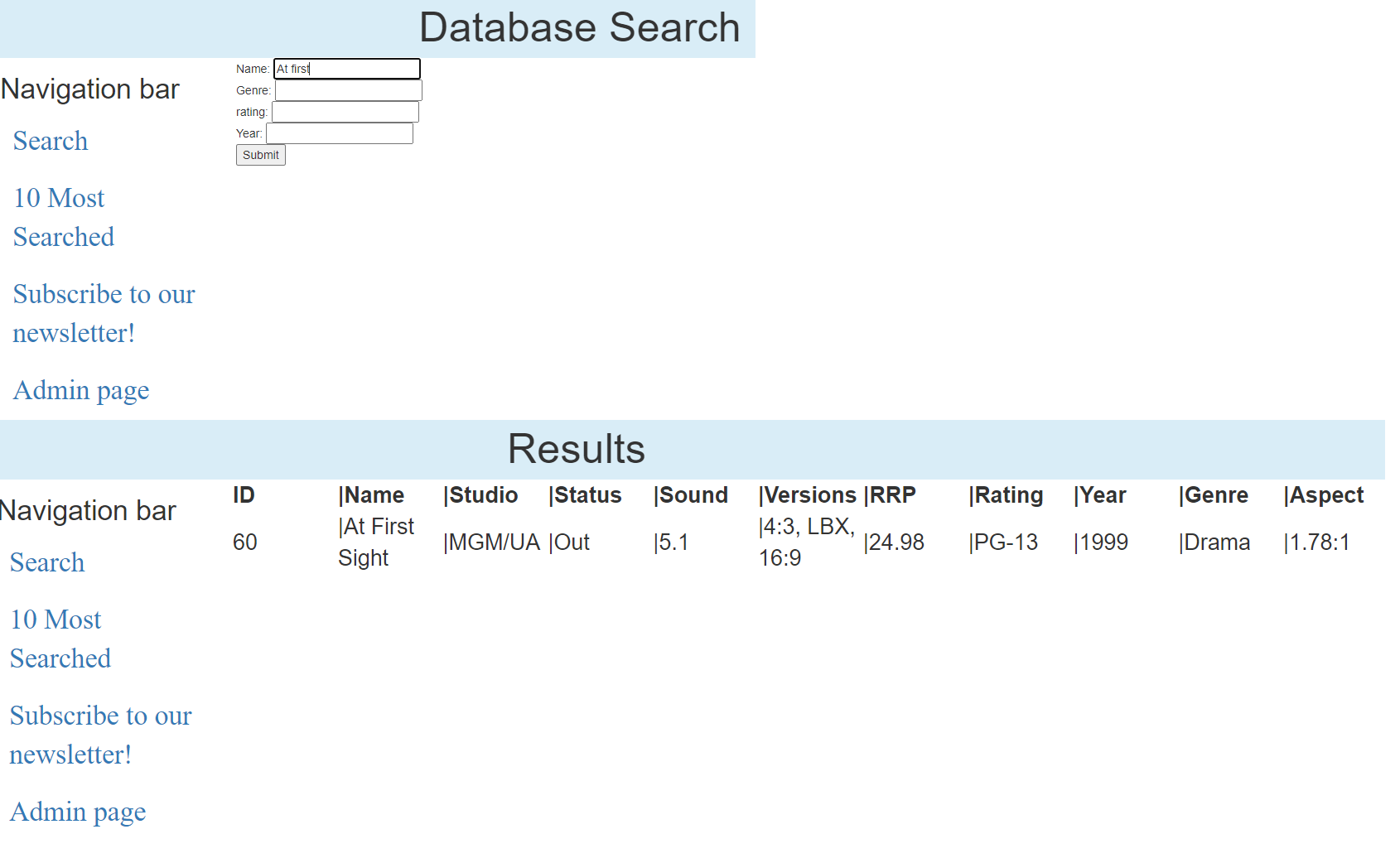


Figure 28- Search by name

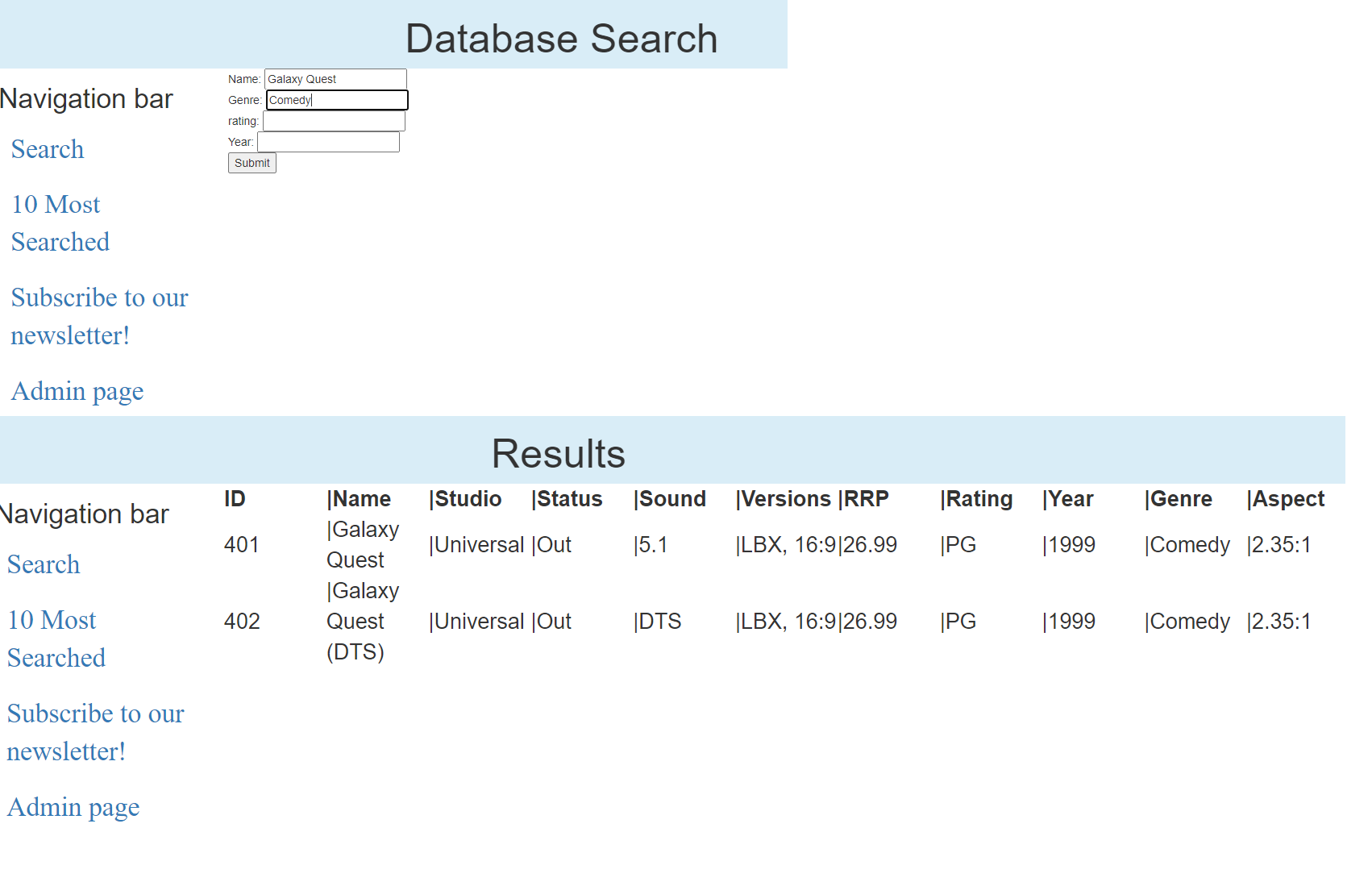


Figure 29- Name, Genre

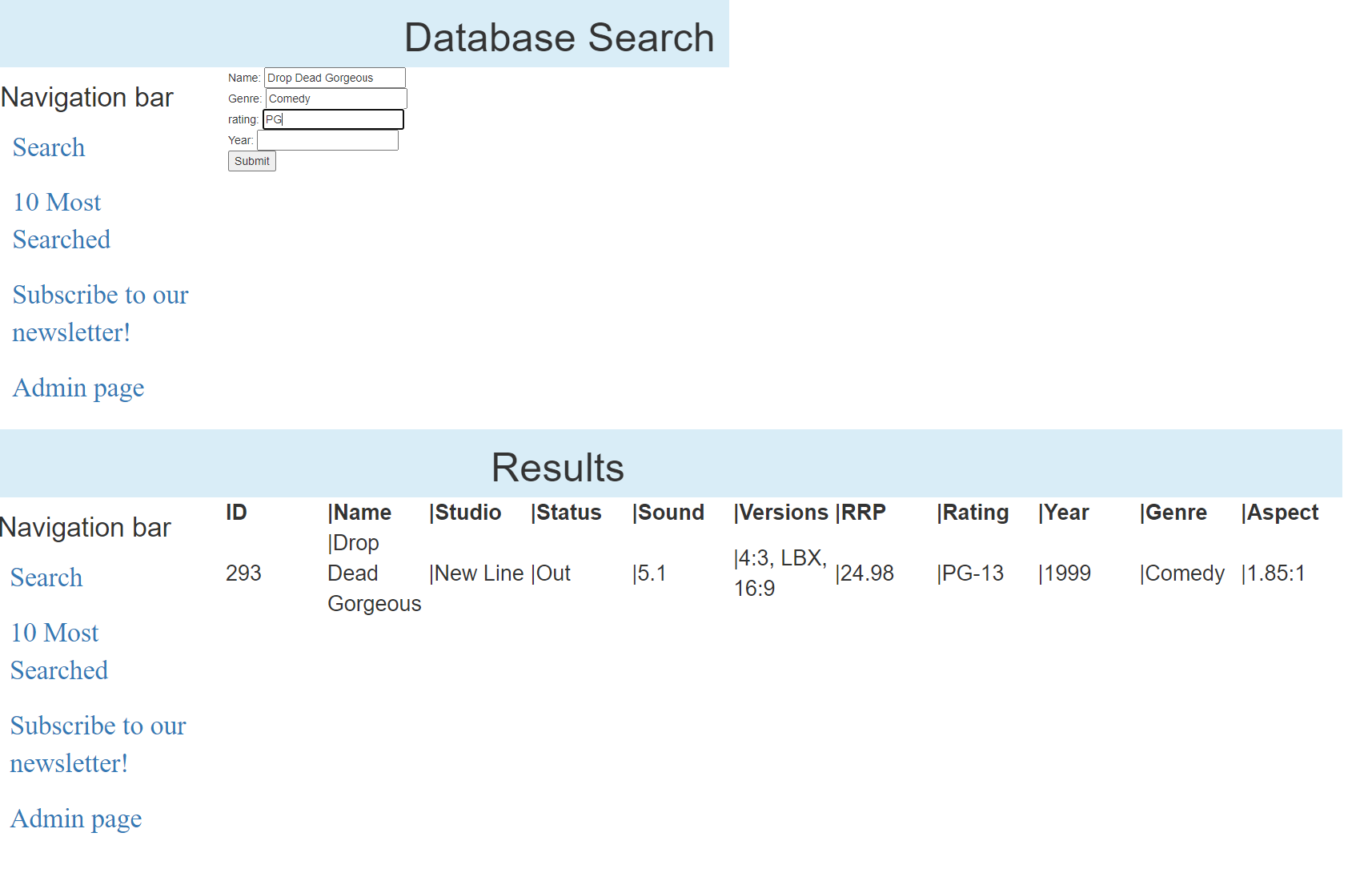


Figure 30-Name, Genre, Rating

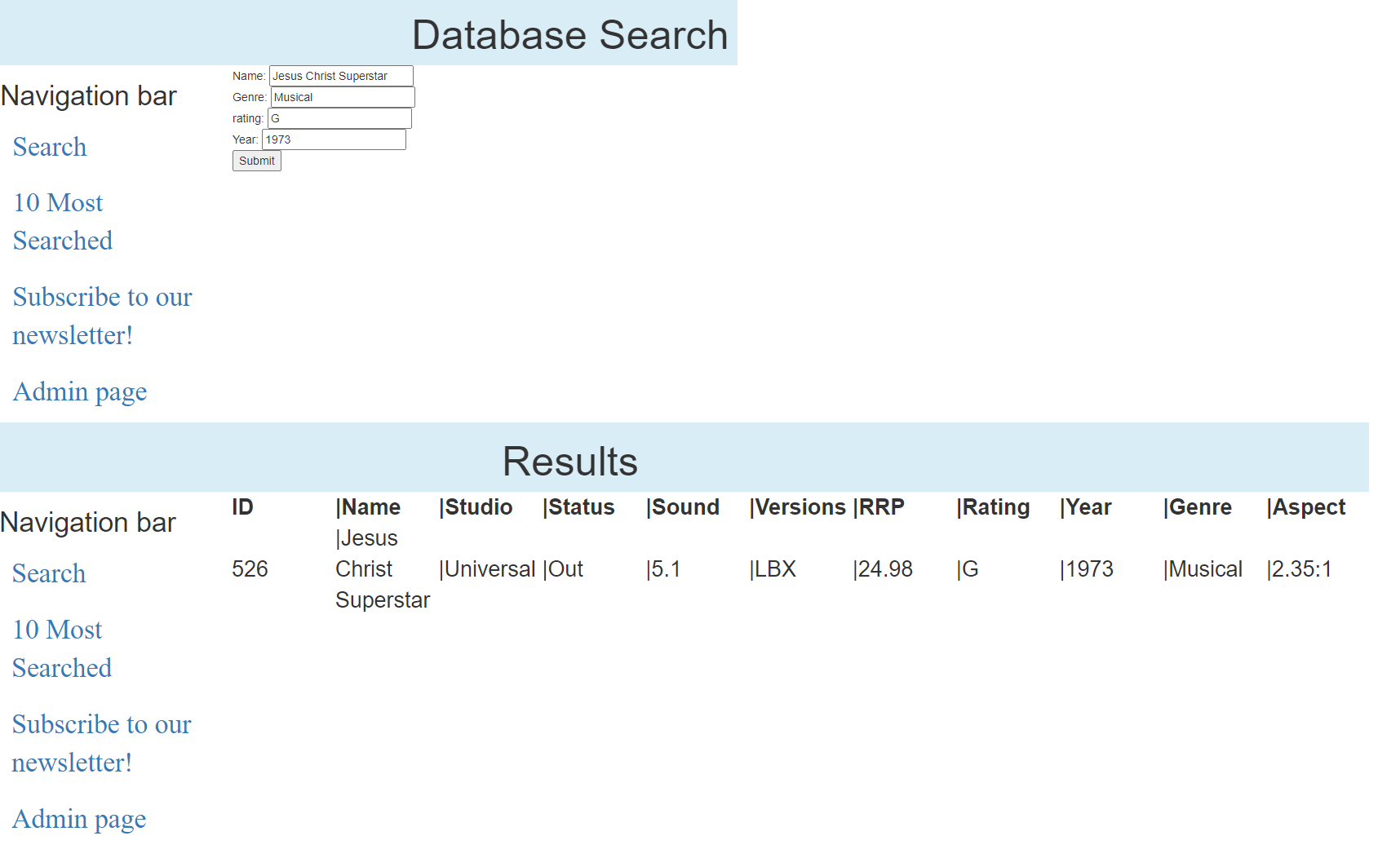


Figure 31- Name, Genre, Rating, Year

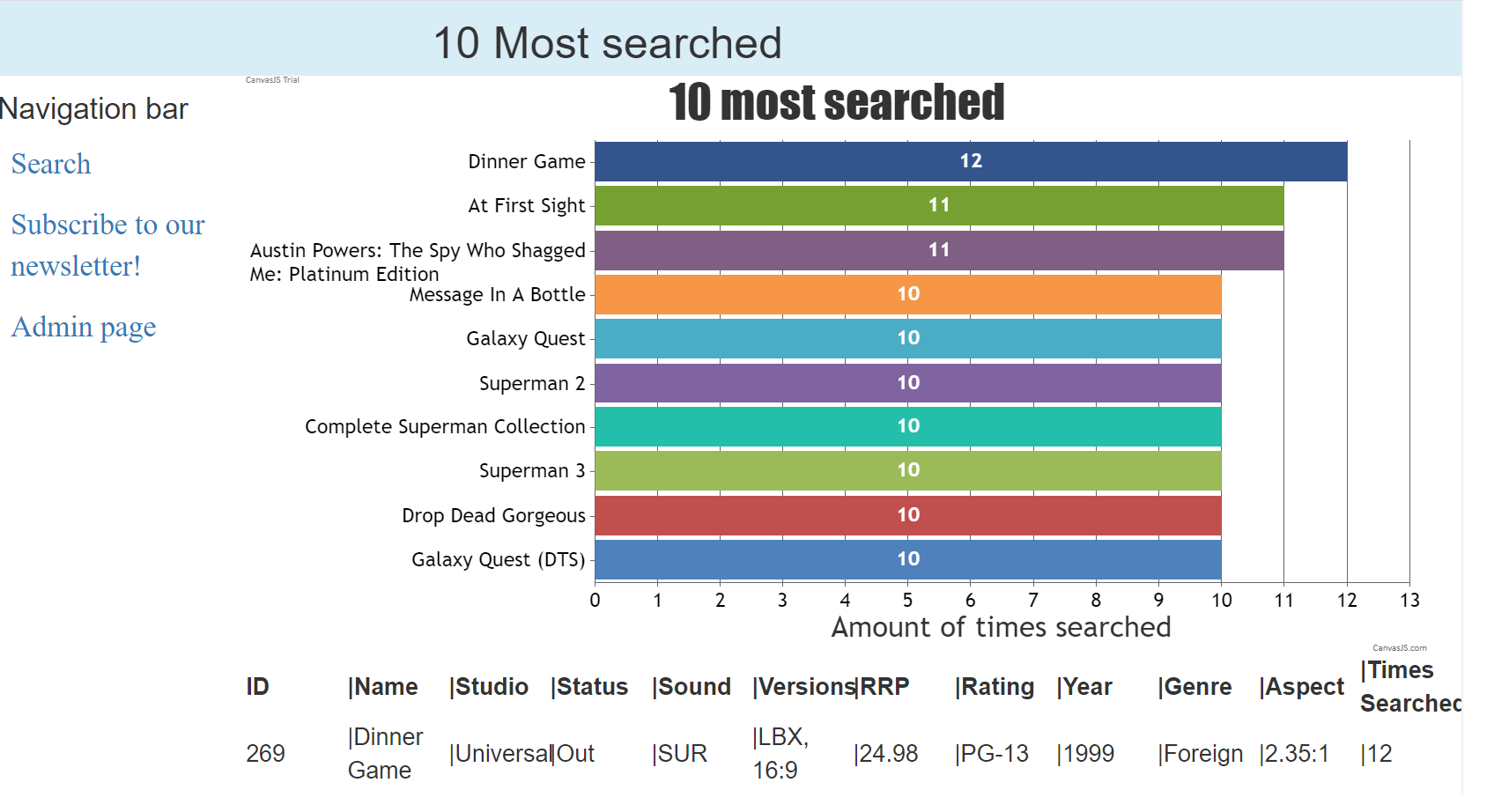


Figure 32- Top ten

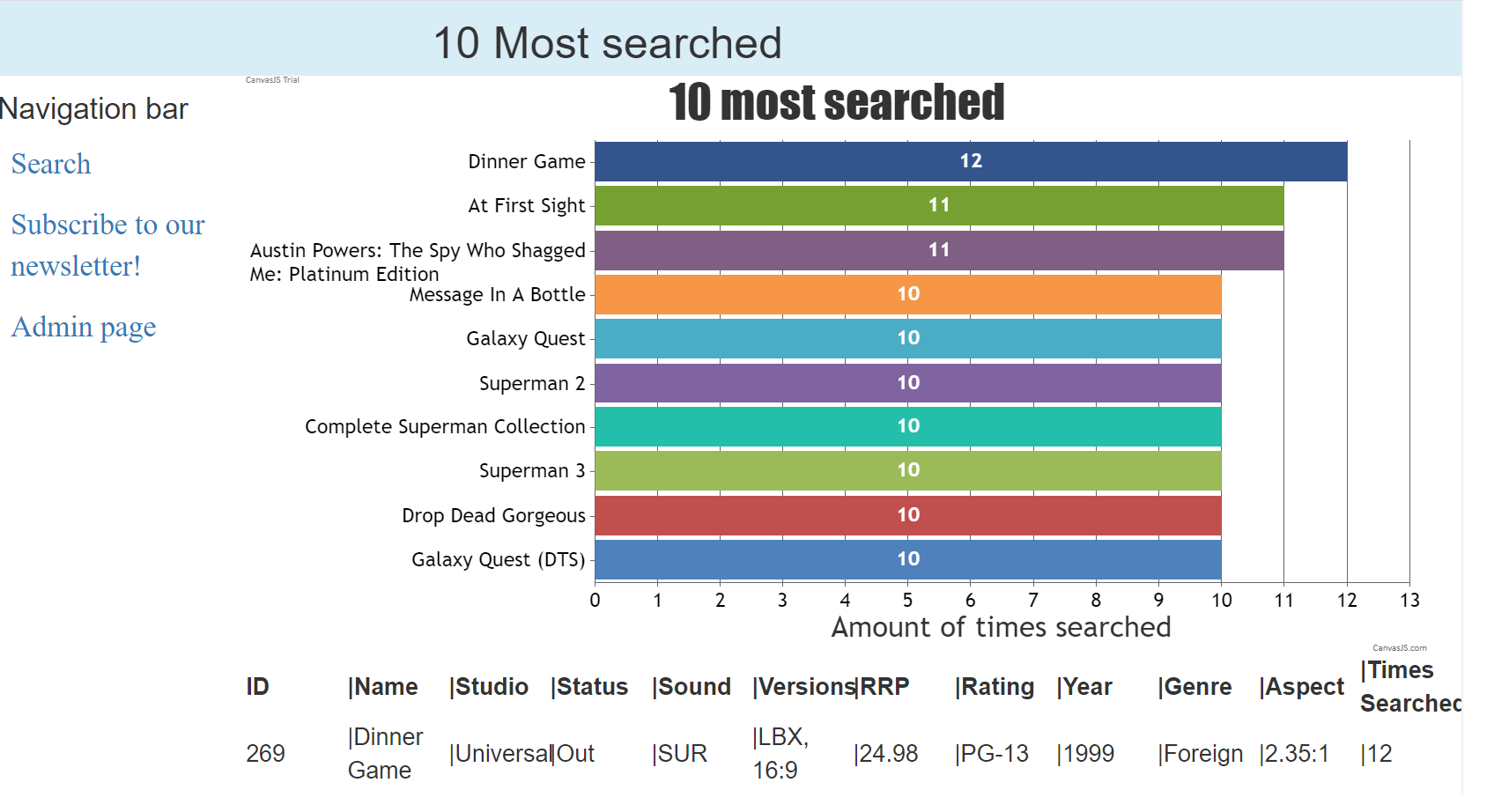


Figure 33- Runs on PC

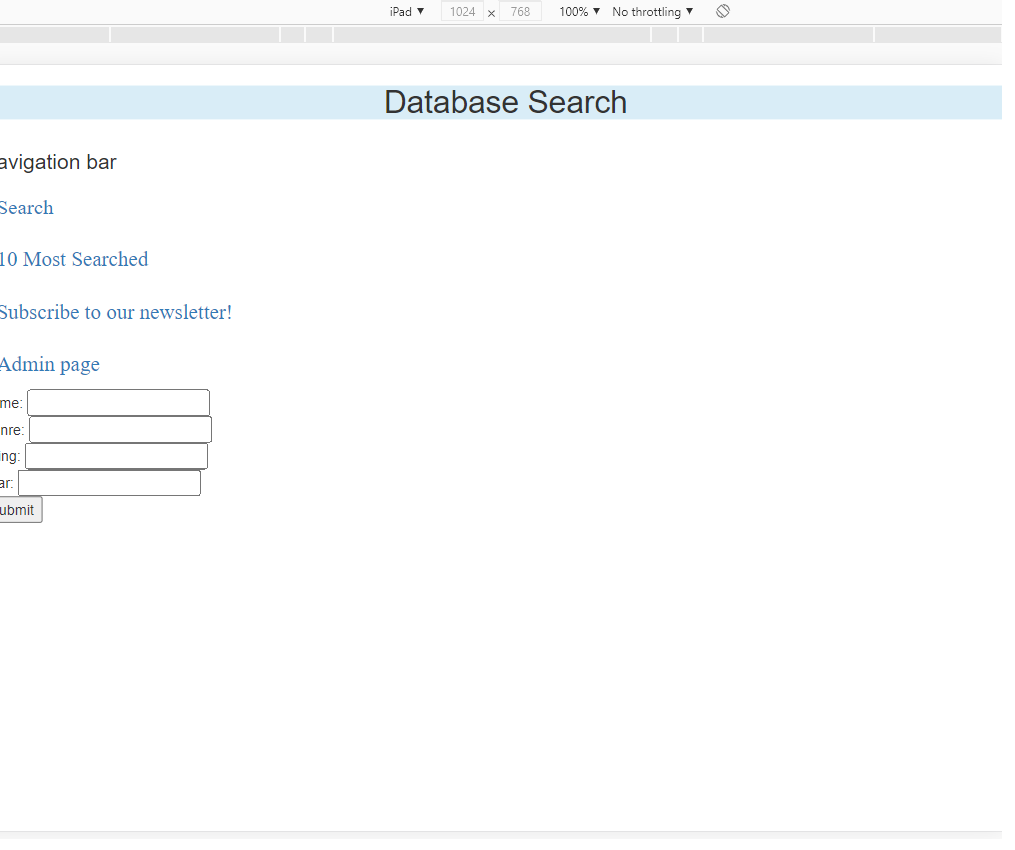


Figure 34- Runs on iPad

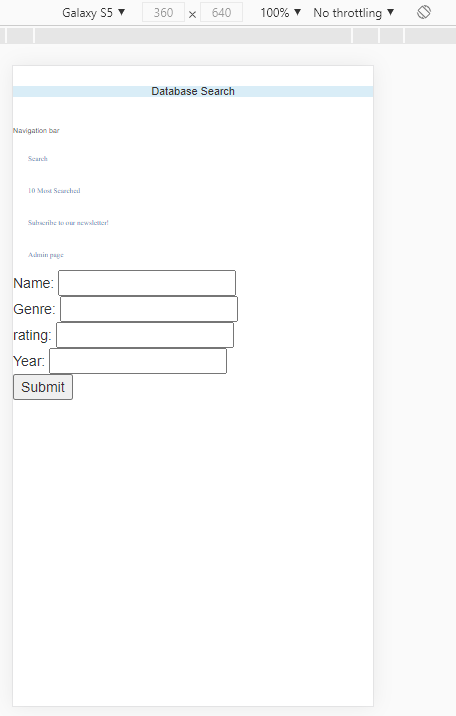


Figure 35- Runs on S5

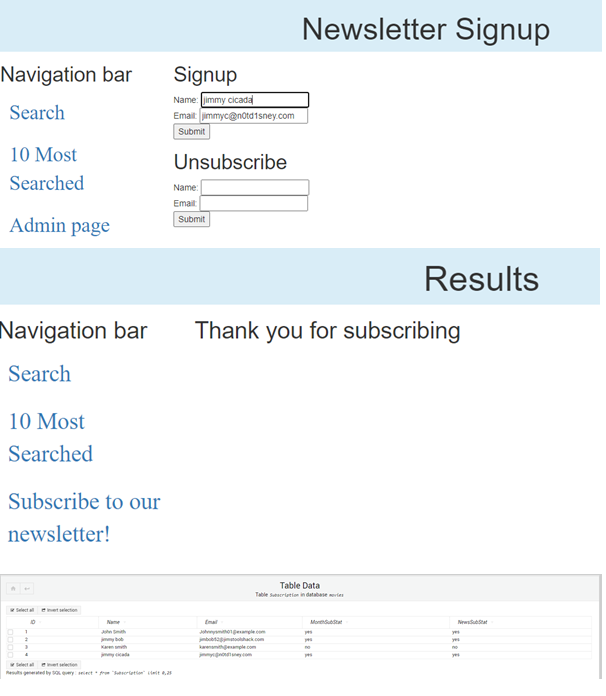


Figure 36- User unsubscribes, database not yet reflected

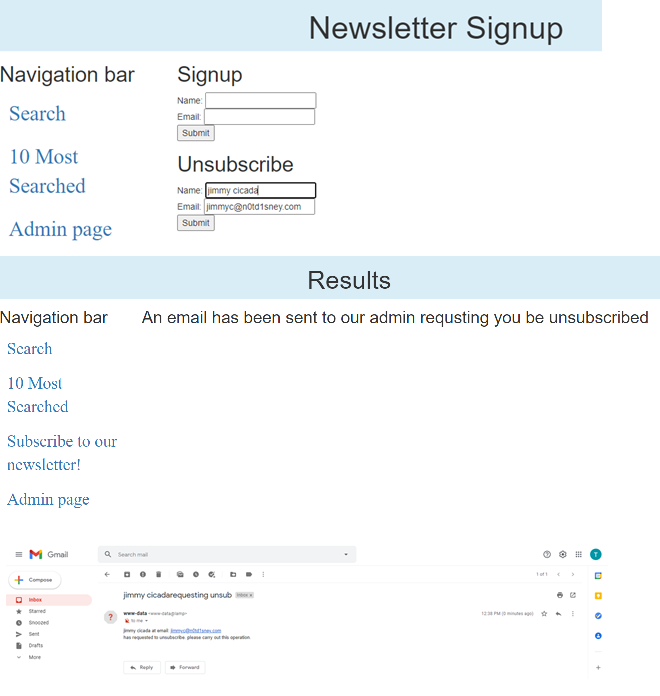


Figure 37- User unsubscribes, admin email

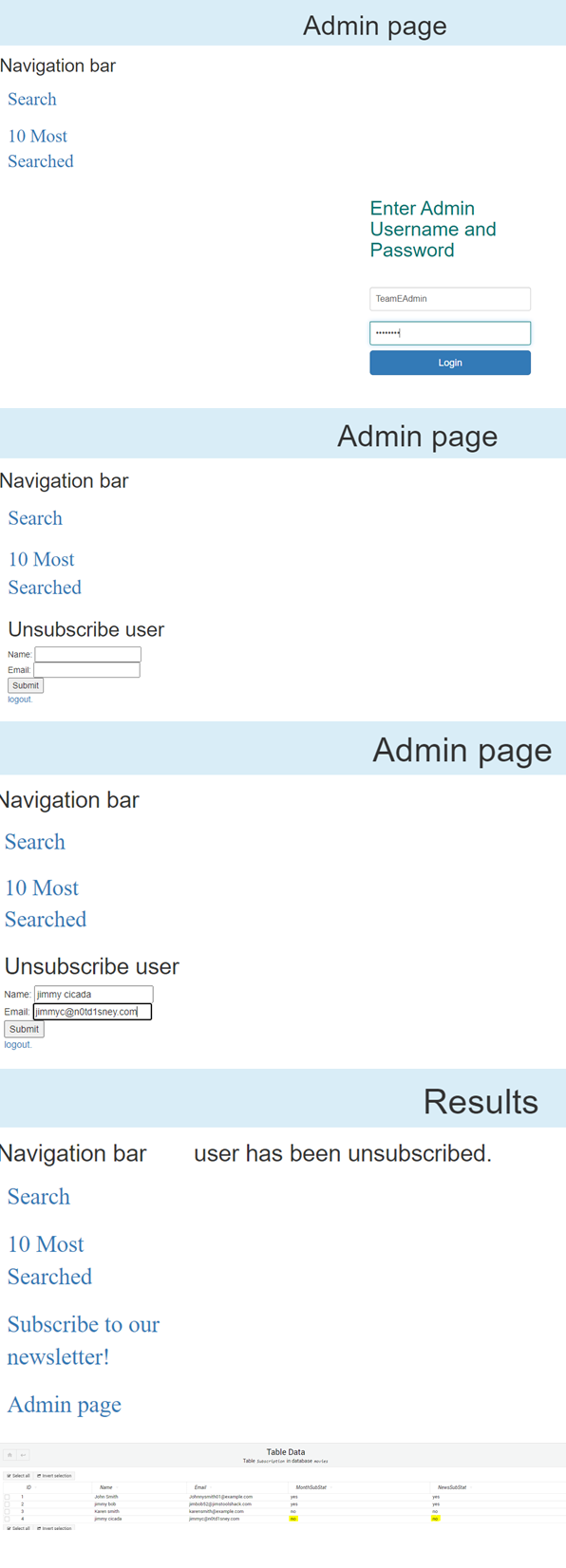


Figure 38- Admin unsubscribes user



Figure 39- Admin logs in

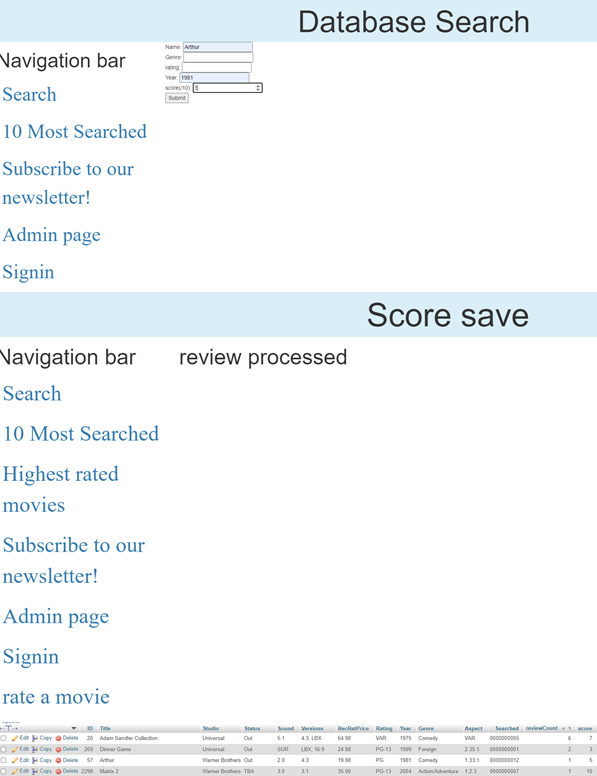


Figure 40- User leaves a score

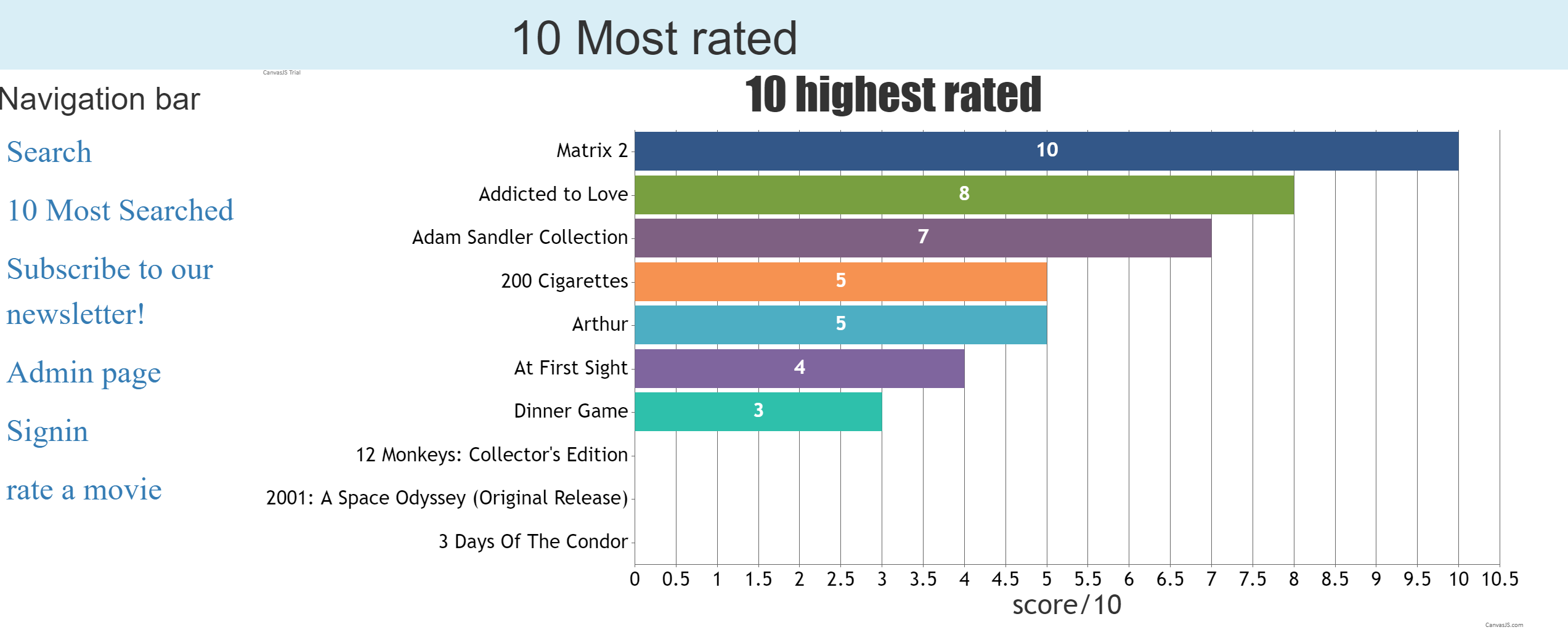


Figure 41- Ten highest-rated

### Team Meeting 3

#### Meeting Minutes

11/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 17/06/2021, 9:00AM, Murdoch Campus

**Erol: Source Control Snapshot**

* Update GitHub
* Create screenshots, add to master document

**General Version Control**

* Manage repositories (GitHub)

**GANTT Chart for SPRINT THREE (week eight)**

* Timeline
* Include who does what

**Optimization report**

* List details of optimization tools to be used
* Create plan for use of tools

**BLAKE Scrum Master, Development**

* Discuss roles
* Record team meeting minutes
* Allocate tasks
* Update requirements based on client meeting
* Record minutes of client meeting

**Development tasks:**

* Admin page
* Database tables for:
  + Ratings on movies
  + User accounts
* Authentication system for accounts:
* Admin features locked behind admin accounts
* Top 10 ratings:
* Add system for rating movies out of 10
* Add a page for showing the top 10 rated movies that updates live.

**Demonstrate the Movie Database Application**

* Read and understand all documentation and code
* Demonstrate application

**NATHAN: Project Management Plan and Testing Plan**

**Software Review Plan**

* Review application against new requirements
* Update testing plan to include new features

**Master document control**

* Collate files into master document
* Proofreading
* Formatting

### Client Meeting 3

#### Meeting Minutes

10/06/2021

Present: Client, Blake Topham, Erol Anil, Nathan Tai

Next meeting: 17/06/2021, 9:30AM, Murdoch Campus

**1. Review Requirements**

* Project requirements discussed and documented.
* Project planning commenced.
* Additions requested.
  + Admin page
  + Database tables for:
    - Ratings on movies
    - User accounts
  + Authentication system for accounts:
  + Admin features locked behind admin accounts
  + Top 10 ratings:
    - Add system for rating movies out of 10
    - Add a page for showing the top 10 rated movies that updates live.
* Requirements noted and agreed.

**2. Optimization**

* Optimization report requested

### Bug Triage 3

#### Meeting Minutes

15/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 17/06/2021, 9:00AM, Murdoch Campus

**1. Bug Report**

No bugs detected.

**2. Triage (Bug Priority)**

N/A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **DESCRIPTION** | **SEVERITY** | **FREQUENCY** | **RISK** | **PRIORITY** |
| N/A | N/A | N/A | N/A | N/A | N/A |