

Flick Blender

Team 3

Increment 2 Report

Rakan Alanzi

Bill Capps

Doug Hoskisson

Introduction

<https://github.com/billjcapps/CS5551Team3/wiki>

Project Objectives and Features

Successful television shows often spawn additional shows and movies. While fans are delighted to see more from their favorite characters, it is easy for a person to become confused about the order each episode should be seen in. Many fans have created websites to help instruct other fans how to watch their show, but these sites often only cover a single franchise. For an individual attempting to create a list himself or herself, it can be a very tedious and time-consuming task. Our project solves this problem by using information from the internet to automatically create a list of episodes in the correct order for viewing.

As mentioned above, there are some websites that solve this problem for an extremely limited scope. Other websites, such as www.IMDB.com [1], provide information about shows without letting combine the information from other shows that are related. There are also several application programming interfaces (APIs) that give access to the same information as the websites. Our project would be unique by letting users create a viewing checklist using any combination of shows and movies that they want.

The objective is to allow people to spend their time enjoying their favorite television programs instead of frustratingly attempting to determine what episode to watch. The application should also include additional information about the shows and individual episodes to prevent the user from having to go to additional locations to find facts such as a cast list or episode synopsis. Finally, the application should be interactive, allowing the user to find information and make changes to the episode list.

The primary feature of the application is to merge television shows and movies together using the date they were released for viewing. Other features will allow the user to modify this sort order using the order the episodes appear on DVD or manually alteration. Additional information is available through APIs. This information could include cast, synopsis, reviews, and what streaming services offer the episode. The list will also let the user mark which episodes have been seen so that the user always knows what to watch next. It may also notify the user of new episodes that will be aired within the next few days.

We are considering also adding some social features to the application. One feature would be allowing the user to see what his or her friends have watched. Another option would be to automatically (at the user's discretion) post viewed episode on Facebook or Twitter. We may also include the ability for the user to provide his or her own ratings or reviews of watched videos.

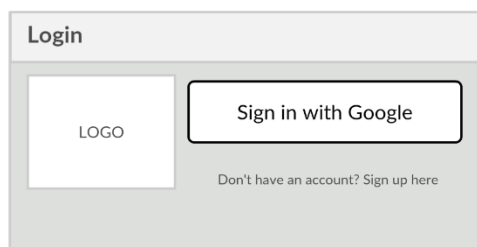
This application would fulfill a need and provide useful features to the user. Like most of applications of this type, it would be supported by advertisement revenue or donations if put into production.

Existing Services

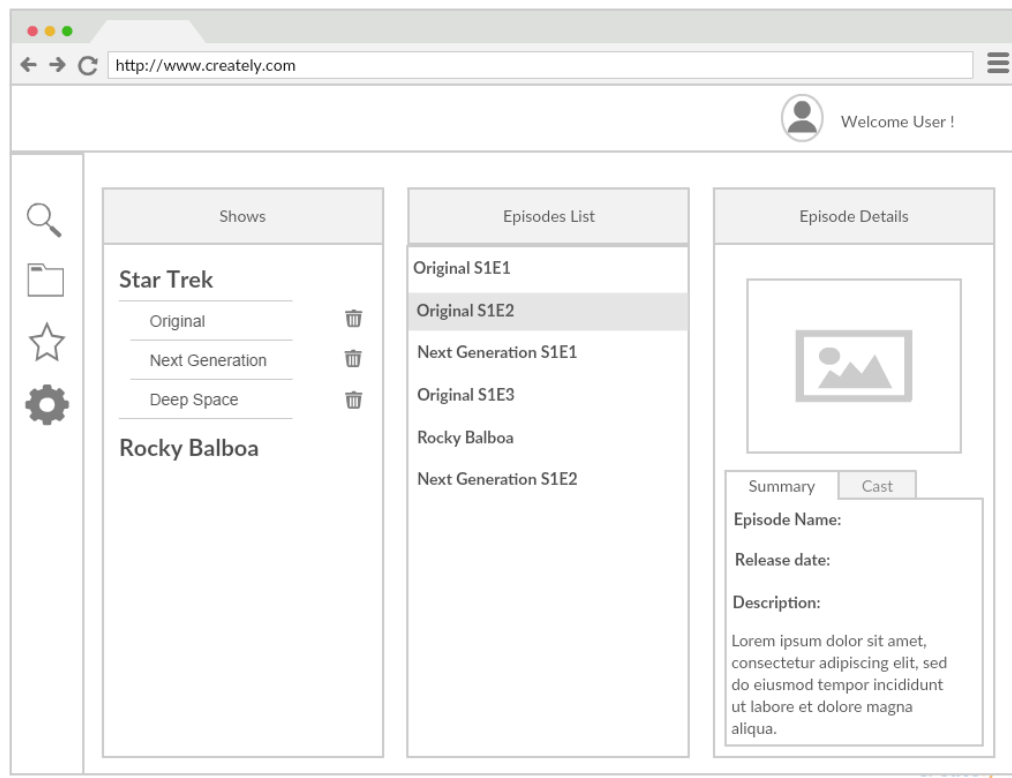
- Google oauth
 - Used to authenticate the user into the system
- TheMovieDB.org API
 - Used to access information about TV shows and movies
- Bootstrap
 - Used to format site and provide advanced CSS functionality

Detail Design of Features

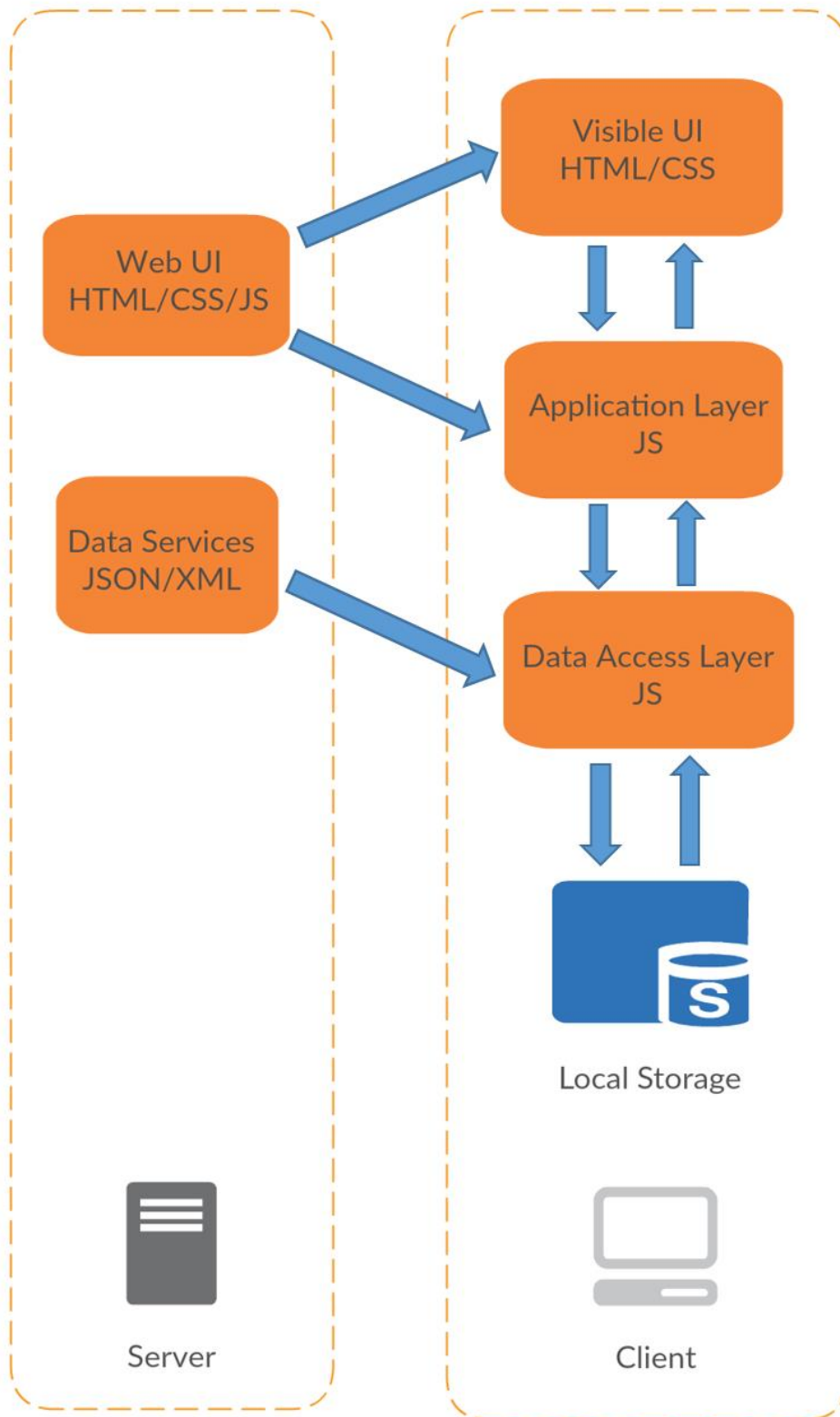
Login page



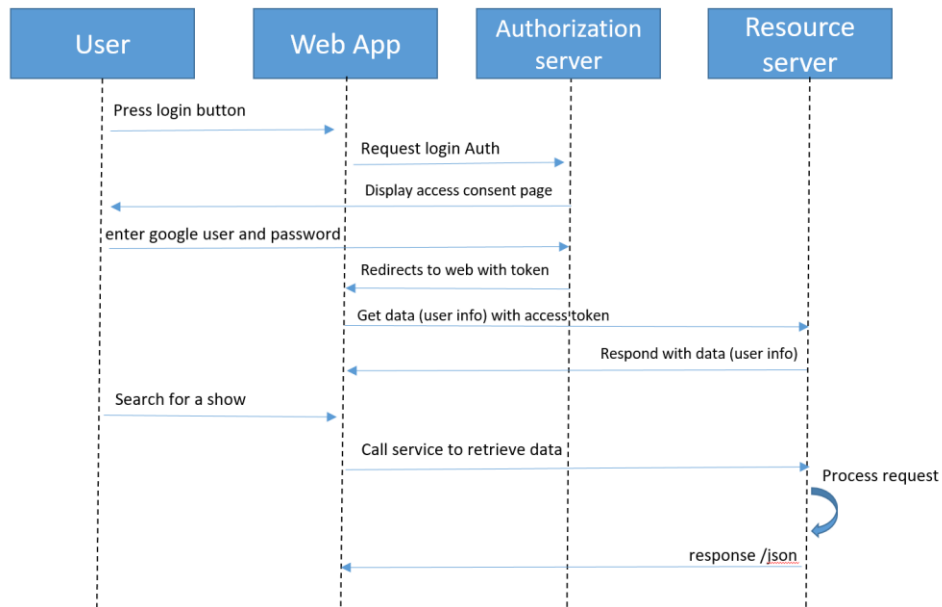
Home Page



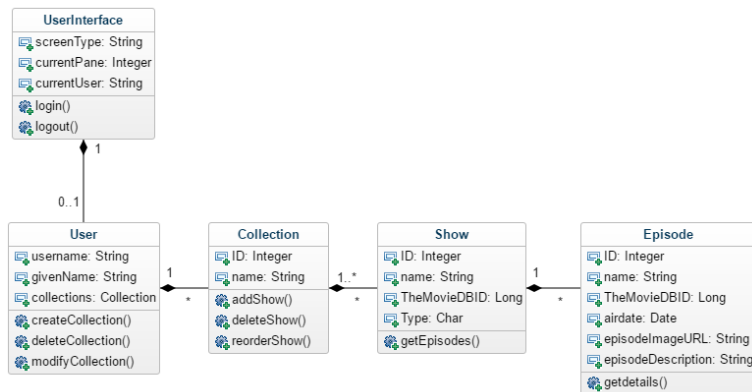
Architecture



Sequence Diagram



Class Diagram



User stories

- Critical
 - As a user, I can create a collection of movies and television shows
 - As a user, I can add and delete from my collections
 - As a user, I can view a list of episodes automatically sorted by airdate
 - As a user, I can manually change the order of episodes
 - As a user, I can see detailed information about an episode
 - As a user, I can mark episodes that I have viewed
- Optional
 - As a user, I can share my collections with friends
 - As a user, I can submit ratings and reviews of shows and episodes
 - As a user, I can get suggestions of items to add to my collection
 - As a user, I can get notifications about upcoming episodes

Testing

| # | Title | Description | Expected Outcome | Result |
|---|-----------------------------|---|--|--------|
| 1 | User with Google account | A user authenticates using Google credentials | The user should be taken to the home page | Pass |
| 2 | User without Google account | A user attempts authentication but does not have Google credentials | The user should be denied access to the system | Pass |
| 3 | Create collection | A user clicks the create collection button | The user is prompted to give a collection name and the new collection appears in the first pane | Fail |
| 4 | Add movie | A user clicks the add movie button | The user searches for a movie, the selected movie appears in the collection and in the second pane | Fail |
| 5 | Add show | A user clicks the add television show button | The user searches for a show, the selected show appears in the collection and all episodes appear in the second pane | Fail |
| 6 | Rename collection | The user clicks the edit collection button | The user is prompted to give a new name for the collection, the new name appears in the first pane replacing the old name | Fail |
| 7 | Delete movie | The user selects the delete button next to a movie | The user is prompted for confirmation; if the user cancels then the user is returned to the main page, if the user accepts then the movie is removed from the collection and the second pane, then the user returns to the main page | Fail |
| 8 | Delete show | The user selects the delete button next to a show | The user is prompted for confirmation; if the user cancels then the user is returned to the main page, if the user accepts then the movie is removed from the collection and the second | Fail |

| | | | | |
|----|--------------------------|---|--|------|
| | | | pane, then the user returns to the main page | |
| 9 | Change order of episodes | The user clicks the episode order button | The episode moves up or down the list depending on if the up or down button was pressed, if the episode is the first episode and the user pressed the up button then nothing happens, if the episode is the last episode and the user pressed the down button then nothing happens | Fail |
| 10 | View episode details | The user clicks on an episode title | The third pane is populated with the episode's image, title, number, description, and review | Fail |
| 11 | Sign out | The user clicks the sign out button | The user is logged out of the application and returned to the sign-in screen | Pass |
| 12 | Export collection | The user clicks the export button | The user is prompted for a location and name for the collection file, a file is generated with the collection information | Fail |
| 13 | Import collection | The user clicks the import button | The user is prompted to select a collection file, the collection in that file is populated in the first pane and episodes contained in that collection are populated in the second pane | Fail |
| 14 | Review show | The user clicks the review show button | The user is prompted to give a rating (1 to 5) for the show and provide a brief statement, the rating and statement are sent to TheMovieDB. | Fail |
| 15 | Review episode | The user clicks the review episode button | The user is prompted to give a rating (1 to 5) for the episode and provide a brief statement, the | Fail |

| | | | | |
|----|---------------------------|---|---|------|
| | | | rating and statement are sent to TheMovieDB. | |
| 16 | Mark episode as watched | The user selects the checkbox next to an episode that is marked as unwatched | The checkbox changes to checked and the episode title is struck-through | Fail |
| 17 | Mark episode as unwatched | The user selects the checkbox next to an episode that is marked as watched | The checkbox changes to unchecked and the episode title is not struck-through | Fail |
| 18 | Get suggestions | The user clicks the get suggestion button | The application displays 10 shows that are similar to the current collection, there is an add show option next to each show | Fail |
| 19 | Create notification | The user selects the new episode notification checkbox next to a show that is not flagged for notifications | The checkbox changes to checked, if the user logs in within 72 hours of a new episode airing, a notification message will display | Fail |
| 20 | Cancel notification | The user selects the new episode notification checkbox next to a show that is flagged for notifications | The checkbox changes to unchecked, no notification is given even if a new episode is going to air soon | Fail |

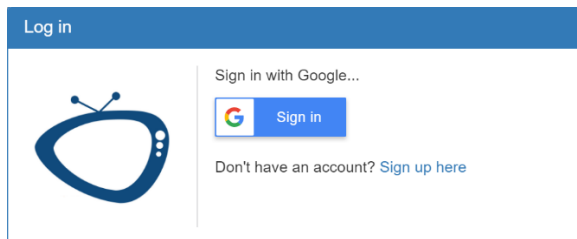
*Additional unit tests can be found in the [source code on GitHub](#)

Implementation/Deployment

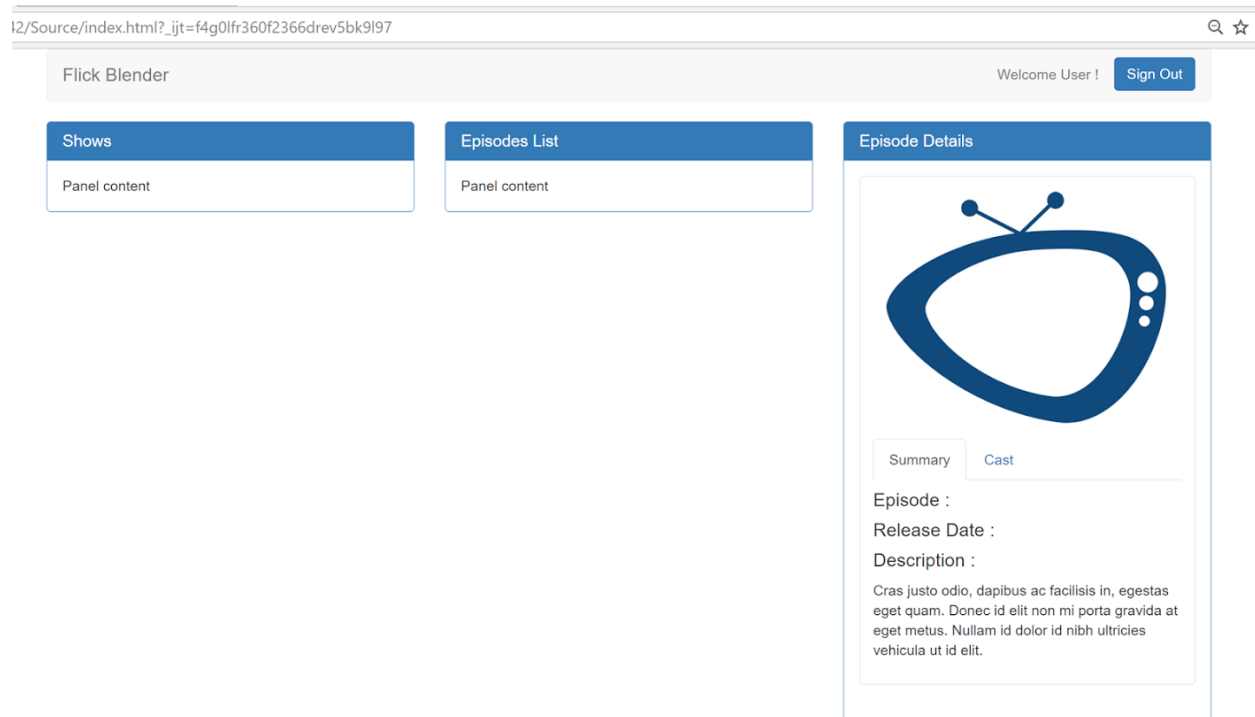
Technologies used:

- HTML5
- REST
- Bootstrap
- CSS
- JavaScript
- Angular.js

The user is first presented with a login page:



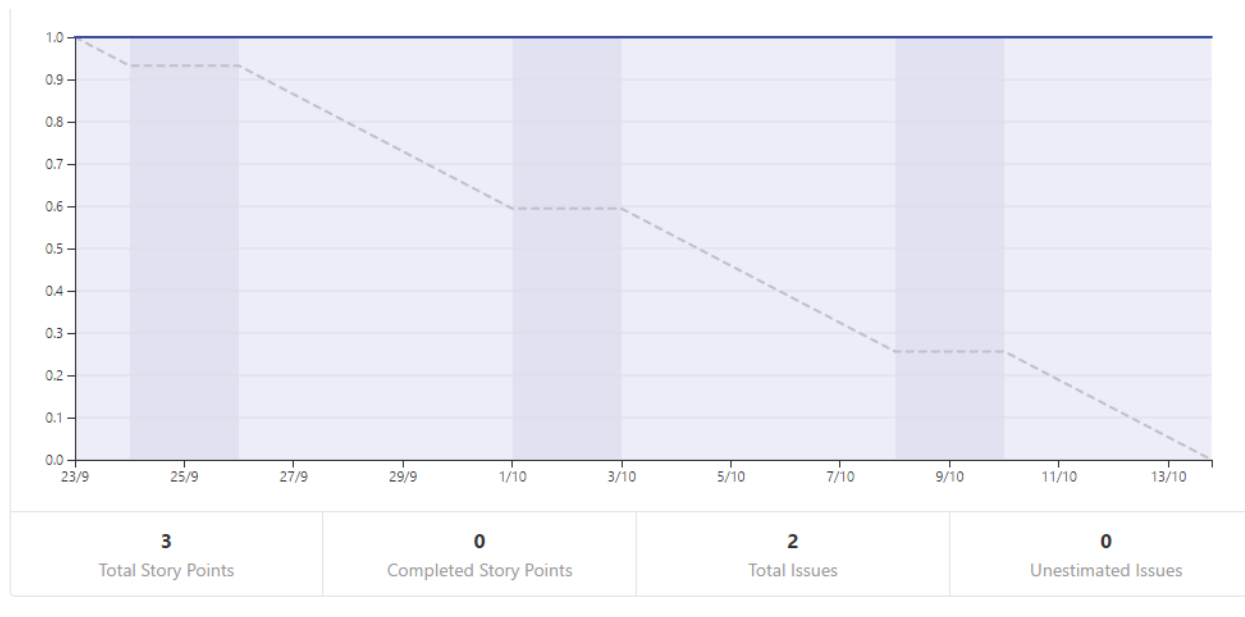
The user is then presented with the main page of the application. Most of the buttons have not been added yet, but all functionality for the application will occur on this page.



Project Management

- Rakan (33%)
 - Assisted with documentation
 - Created application framework
 - Researched possible technologies
 - Attended team meetings
 - Around 10 hours
 - Future tasks:
 - Add collection display and buttons
 - Add episode display and buttons
 - Add detailed display and buttons
- Bill (33%)
 - Got API key for Google oAuth
 - Got API key for TheMovieDB
 - Created reports
 - Created classes to access TheMovieDB
 - Created classes to store data
 - Attended team meetings and reserved space
 - Around 10 hours
 - Future tasks:
 - Make data classes accessible to the rest of the application
 - Make collection information permanent (local storage)
 - Create data structure to contain episodes
- Doug (33%)
 - Created authentication modal
 - Researched possible technologies
 - Created unit tests
 - Attended team meetings
 - Around 10 hours
 - Future tasks:
 - Create add movie/show modal
 - Create collection naming modal
 - Create suggestions modal

Burndown Chart



Bibliography

1. IMDB - <http://www.imdb.com/>
2. Episode Guides Android Application - <https://play.google.com/store/apps/details?id=com.lordmarty.epguides>
3. TVmaze.com - <http://www.tvmaze.com/>
4. Yahoo! Answers - <https://answers.yahoo.com/question/index?qid=20090713153353AAw4mTd>
5. GateWorld - <http://www.gateworld.net/news/2009/05/stargate-recommended-viewing-order/>
6. Arvin Johansson Arbab - <http://arvtard.com/stargate-sg-1-and-atlantis-viewing-order-python-code>
7. StackExchange - <http://scifi.stackexchange.com/questions/4877/what-order-should-i-watch-the-various-stargate-television-series-in>
8. Tumblr - <http://nowwearealltom.tumblr.com/whovieworder>
9. Rothwell - <http://rothwell.im/etc/doctor-who/>
10. Blogspot - <http://wibblywobbly-timeywimeystuff.blogspot.com/2012/12/doctor-who-torchwood-viewing-order.html>
11. Daniel Goleman - <http://www.danielgoleman.info/>