

Phase 1 Report

EE461L

Github repo: https://github.com/chrisjoswin/EE461L_Project

Team Information:

Jacob Grimm	jacobgrimm@utexas.edu	jacobgrimm
Jerad Robles	sebastian.robles@utexas.edu	JSRobles
William Gu	williamgu@utexas.edu	Minalinnski
Christopher Erattuparambil	chris.joswin@utexas.edu	chrisjoswin
Josh Kall	joshuakall@utexas.edu	j-ka11
Haosong Li	hl27346@utexas.edu	hdlee9885

Team Canvas Group: Morning-9

Project Name: Internet Comic Database (ICDb)

Motivation and Users

Although there are many comic databases online, the official websites only have comics from the same universe while unofficial ones are usually messy and ad-heavy. Our team intended to integrate online sources and build a clean, user-friendly database for all famous comics. The intended users are all comic readers, no matter if they are fans of a character or a creator, or they are interested in reading the origins of the newly released movie.

User Stories

- As a fan of Dan Slott, I want to search by his name so that I can find other works he wrote.
 - Time Estimate: 1 hour
 - Actual Time Required: 2 hours
- As a lover of Captain America, I want to find issues that he is featured in so I can read more comics with Captain America in them.

- Time Estimate: 1 hour
 - Actual Time Required: 1.5 hours
- As an amateur comic writer, I want to learn more about who wrote my favorite comic issue so I can be more familiar with their work.
 - Time Estimate: 1 hour
 - Actual Time Required: 1.5 hours
- As a hardcore Iron Man fan, I want to read his bio page so I can learn everything about him.
 - Time Estimate: 2 hours
 - Actual Time Required: 2 hours
- As a new comic reader, I want to learn more about all the characters in my first comic book so I can broaden my comic knowledge.
 - Time Estimate: 1 hour
 - Actual Time Required: 1 hour
- As a comic collector, I want to find issues from the Silver Age (1970-1984) so I can see if my collection is complete.
 - Time Estimate: 2 hours
 - Actual Time Required: 1.5 hours
- As a comic enthusiast, I want to see issues where two of my favorite character crossover so I can read them.
 - Time Estimate: 2 hours
 - Actual Time Required: 1 hour
- As an International Reader, I want to find writers from my country so I can read their comics.
 - Time Estimate: 2 hours
 - Actual Time Required: 1.5 hours
- As a casual comic reader, I want to find issues from a new series so I can start reading that series.
 - Time Estimate: 1.5 hours
 - Actual Time Required: 2 hour
- As a fan of hero groups, I want to see what groups a character belongs to so I can see how that character fits in the comic universe.
 - Time Estimate: 1 hour
 - Actual Time Required: 1.5 hours

Use Case Diagram

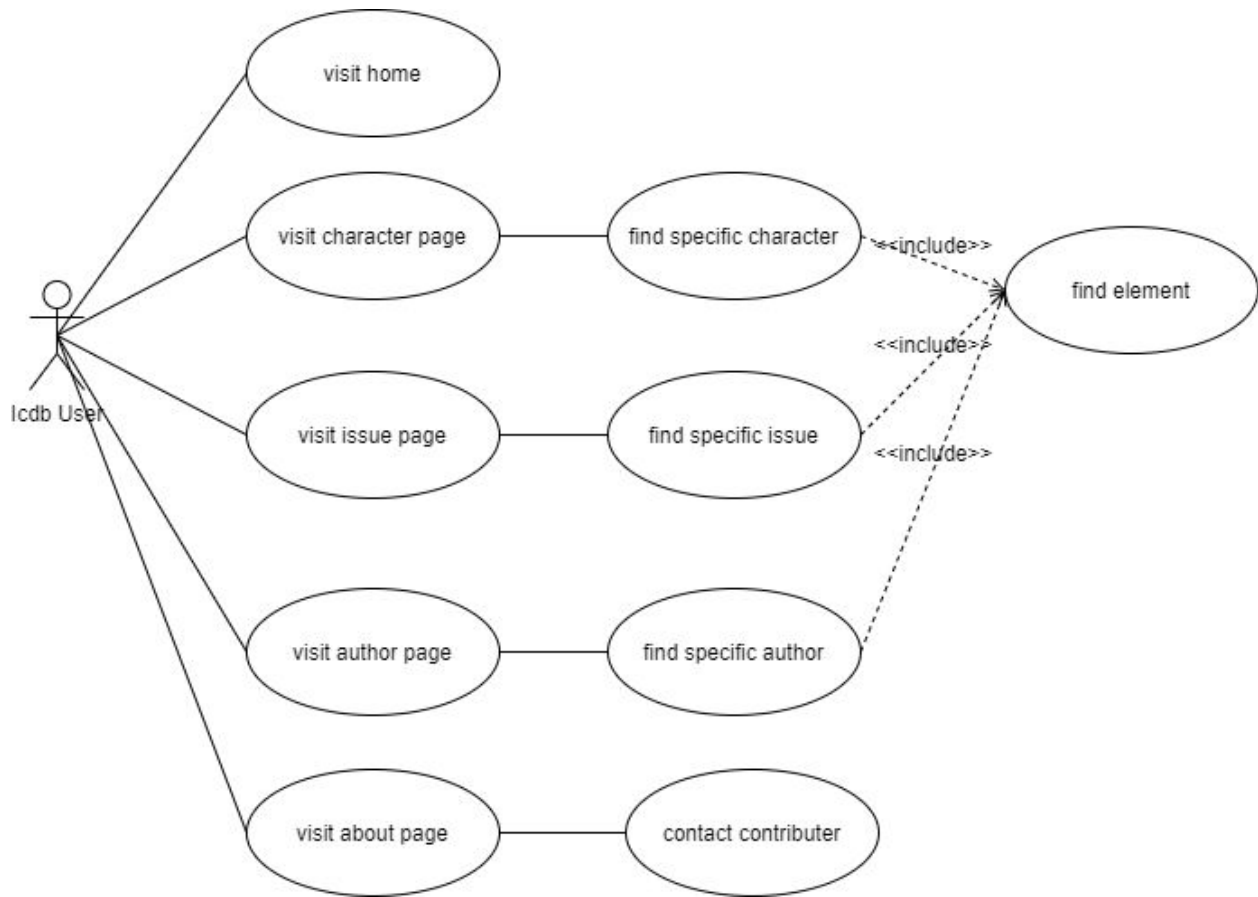


Figure 1: Use Case Diagram for Icdb user

Design

Front end:

The front end currently consists of 4 different types of pages constructed using Angular. The 5 pages are the home page, this provides a clean entry in our website and allows the user to have easy access to website features. The about page allows the user to learn more about our team and progress on the websites development. The model page provides access to all instances of that particular model, there are 3 of these pages, one for each model. Lastly is the instance page, this gives the user all the

information attached to that particular instance, there is 1 such page for every instance of the 3 models.

Back end:

The back end consists of several python scripts that pull relevant information to a specific model from a specified API and outputs the information into a JSON file. For the Character model the Superhero API is used, for the Issues model the Marvel Comics API is used, and for the Authors model the Comic Vine API is used.

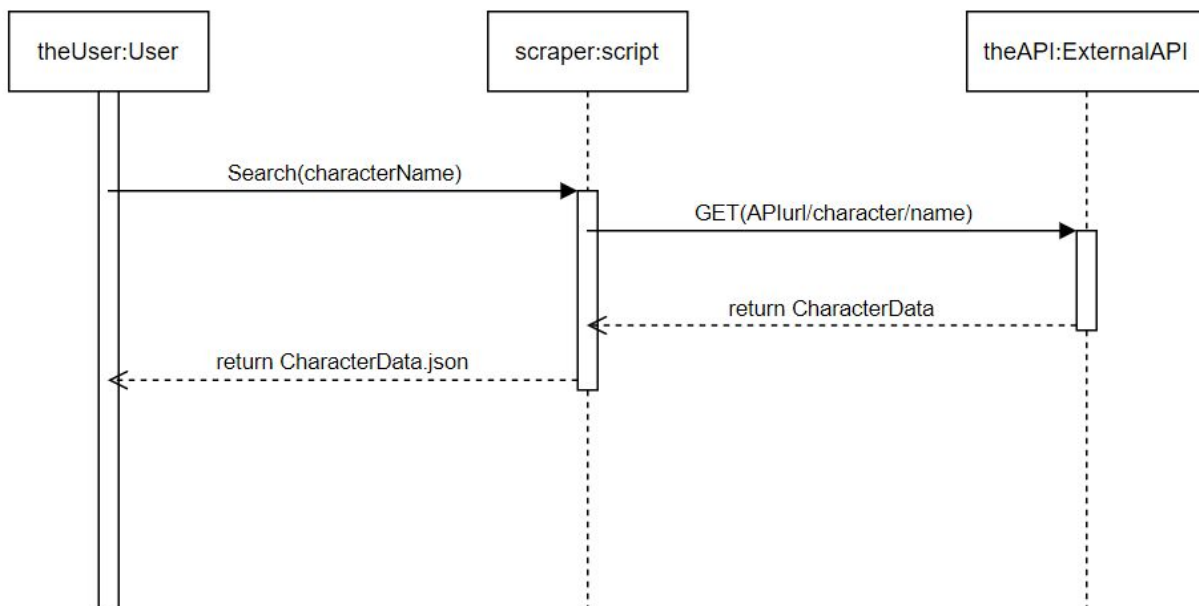


Figure 2: Sequence Diagram for data retrieval script

Testing

After our testing methods we are quite confident in the robustness of our web application and error-free programming that we have done. Of the many testing techniques we have done, the most significant front-end testing was clicking and examining the success of the links between each of our static pages. As well, we tested our navigation bar on both mobile and pc browsers, and the rate of reply was quick and it returned the proper responses.

In regards to our back-end testing, we tested our scrapers on a number of different API resources and each run returned exactly what we wanted. So we are fairly confident in the success rate of it being quite high.

Models

Our database website consists of three models: Comic Book Characters, Issues, and Authors. Every instance of each model will contain data and media obtained from a source. The data and media for each model as well as the source will be as follows.

The Comic book characters model we are using will consist of a few distinct elements, with a few holding an important relationship to the others. The elements held in each hero model will consist of some of the superhero's fictional biographical data, it is fictional because as far as we know, none of these superheroes exist outside of their respective comic books or movies. As well, we will be including information about the powers they have, specific/notable issues that the respective character is featured in, as well as the author and the hero's first appearance. We will also include a nice thumbnail image of each character, so the user can better identify who the hero is. The sources for this information are from each the SuperHero API, the Marvel Comics API, and the ComicVine API.

- Comic Book Characters
 - Data: Biographical information, powers, issues appeared in, name of the creator, and creation date.
 - Multimedia: Image of the character, alternate images, and video if available.
 - Source: SuperHero API

Another model we are opting to pursue, is individual comic book issues, and the information associated with them. The certain elements of each model include, the name of the issue, the name of the series that the issue is a part of, a meaningful summary/description of the issue, the characters included in the issue, and the writers of the issue. As well, we plan to include a picture of the front cover of each particular issue so that the user will be able to recognize if they have viewed it in the past, and also understand the artists' style a little better. This information will be gathered from the Marvel Comics API and the ComicVine API.

- Comic Book Issues
 - Data: Name of Issue, name of Series, description of issue, characters featured, and story authors.
 - Multimedia: Image of the Issue cover, frames from comic if available, and series image if available.
 - Source: Marvel Comics API

Our final and least significant model is the Comic Book authors model. The elements we will try our best to include are the author's biographical data, which is not fictional, because the authors are real people. This is a stark contrast to the existence of the characters, a powerful irony considering the authors write the characters, but the characters do not exist in our sense of existence. As well we will add elements of what specific issues the authors wrote, the characters they have helped write for, and the social media handles of each other, in case the user is inspired to tweet at the writer. We also plan to include images of the author, of the covers of the comics they made, and interview videos they are featured in (if available). The primary source for this information will be the ComicVine API, with maybe some slight help from the Marvel API.

- Comic Book Authors
 - Data: Biographical information, issues written, and social media handles if available.
 - Multimedia: Image of Author, covers of issues written, and interview videos if available.
 - Source: Comic Vine API

A very interesting aspect of our models is that they are all interconnected through one aspect of each other. The author models are connected to the issues and characters, through the issues and characters they've written. The character models are connected to the authors and issues, by the authors that have written them, and the issues they are featured in. And the issue models are connected to the characters by the characters in the specific issue, and the authors by the author of each specific issue. The significance of this is that we will be able to create convenient and nice hyperlinks to the author models quite easily, making our website more robust.

Tools, Software, and Frameworks

The tools we used for this section were quite interesting. For the frontend we elected to use Angular. In the implementation, it proved somewhat difficult to configure and setup, but it seems as though the efforts will be worth it for the ensuing phases. Angular is a front-end framework that uses javascript to help you make more robust websites. The frontend team really learned a lot about angular and how to develop a nice website with angular components.

Another tool we used was more in the backend, but we programmed a few python scripts to scrape the necessary data we needed from the APIs. This proved to be surprisingly interesting! The scripts essentially made the calls, grabbed the We

really enjoyed writing the scripts and cannot wait to implement them into a pipeline next phase to put the data we acquire from the API into our database!

One final tool we used was Postman, we used this to test out API calls that we were trying to make and examining the return content. Postman is a tool that can make API calls from a very friendly user interface, it is very useful for API testing. We found it quite useful just to ensure that we were calling the APIs properly.

Reflection

Our team performed very well in bringing our individual skill sets together. We did not run into any problems with getting particular technologies to work for us and because of this, we were able to complete a lot of the technical work fairly quickly. One thing our team needs to improve on is our communication, especially between sub-teams. There were a couple of instances where the front-end team did not know what the back-end team was doing or what technologies they were planning on implementing and vice-versa. It's not easy to move around and understand the new techniques we just learned for either frontend or backend in a short time, but we should try to eliminate the cutoff between sub-teams by understanding the others' works. We learned that skill-wise our team is very capable, but that we need to work on improving our communication and scheduling of project components and team deadlines.