

App Idea: The app will store analogies for computing concepts. Computer science instructors will be able to search the analogies to find new ways to explain complex concepts to their students. Instructors will also be able to create new analogies that they use in their own classrooms so that other people will be able to use them in their own teaching.

Users: The app will primarily be used by computer science instructors at any level. They may be undergraduate TAs, graduate TAs, professors, high school teachers, etc, anyone who is teaching computer science topics. Because they are instructors, they will have some level of computing expertise. They may also be older (close to retirement), so the app must be accessible and able to use different text sizes, image alt text, high contrast colors, etc. Users can also view analogies in a “student” view, but the view will only be limited to searching and viewing analogies.

Major Workflows:

Required:

Browsing Analogies:

An instructor will open the app and see all of the analogies displayed on screen. They are able to look through those analogies and click them to get more information about that analogy, including its different parts.

Searching for Analogies:

A user will be able to use a search function to find all analogies that relate to a particular computer science topic. They will enter their search and then see all analogies matching the criteria. They are then able to browse the matching analogies and click them to get more information about that analogy.

Preferred:

Creating Analogies:

The user would be able to create a new analogy, and the app would then walk them through the process of creating an analogy. The user would not be forced to write the

analogy in a predefined order, instead they can write the components of the analogy in any order, for example starting with the source domain versus starting with the target domain.

Comparing Analogies:

While browsing analogies, the user will be able to select one analogy, but still be able to find other analogies on the search/browsing page. Once a second analogy is selected, the user can compare the two analogies and decide which one their class would most benefit from.

Optional:

Analogy Popularity:

The app will display how often each analogy has been viewed, suggest popular search terms, etc. This will let instructors know which analogies are popular and which ones people are looking for that don't currently exist.

Favoriting Analogies:

Users will be able to "favorite" analogies to save them for later. They will then be able to see a page that shows all of their favorite analogies so that they may look at them more later.

Views:

- Browsing page
- Analogy information page
- Analogy creation page
- Comparison page
- User page (also shows favorites)
- Login/Authentication page

Data: The app will store text data associated with the actual analogies. If a user system is implemented, it will also store user account information (username, password, created analogies, favorite analogies, etc). It may also store data on popular analogy searches.

Anticipated Challenges:

- Parallelization of our development
- Running out of time 😞
- User authentication