COMS W4170: User Interface Design—Fall 2018

Prof. Steven Feiner
Date out: Friday, November 16, 2018
Teams formed: Tuesday, November 20, 2018
Proposed design concept due: Thursday, November 29, 2018
Project due: Tuesday, December 11, 2018
Project presentation: Tuesday, December 11, 2018 6:30pm, 413 Kent

---

Final Project: In a GIF

Introduction

For your final project, you will be taking advantage of the Giphy API through its Giphy Javascript SDK. Giphy supports the ability to search for GIFs; for example, through strings or a request to find trending or random GIFs. (Giphy will also allow you to upload GIFs. However, since Giphy imposes a limit of 10 uploads a day, your final project should *not* use the upload functionality.)

The goal of your four-person team final project is to create a web application using Giphy to address a domain of your choice. The domain can be as general or as specific as you would like. Bt before you begin, please verify that there will be the right kind of data available to meet your needs. Since the Giphy API is rather limited, you are encouraged to use, in addition (*but not instead*), other APIs, beyond the Giphy API.

Your application should first (a) display a main page of information relevant to your chosen domain when the application is invoked. The user should then be allowed to (b) interact with information that has already been displayed to obtain additional related information, and (c) formulate and execute queries in your domain that use the Giphy API. In general, your application should (d) allow the user to interact with the GIFs and any other information presented in a way that supports the goal of your application. Your application should also (e) support saving user-selected information across sessions (e.g., a query and its response); to do this, you can use store.js, or any other approach.

Please be aware of the rate limits imposed by the Giphy API. Your app should be able to work fine for development and testing purposes with a regular API key, not a production key. While you should not use the Giphy Upload endpoint in your application, you are welcome to use the Giphy Upload webpage to upload GIFs outside of your application.

This is all that you are being given as a "spec," so a lot of what you do is going to be up to you. Needless to say, your team has a limited amount of time in which to do this, making it important to restrict your scope. (But please don't think too simple. Applications similar to Random Goat or Giphy TV, proudly featured at Giphy Labs, are not going to be acceptable when we review your design concept.) Remember that this assignment is weighted more heavily than the previous assignments, and that it is a group project. Be creative!

Personas and Use Scenarios

Once you have a basic understanding of the functionality provided by the APIs you intend to use, you should identify a single class of users with whom you are familiar (e.g., college students), who could benefit from your application. Create at least two detailed personas that represent this class of users, and at least one use scenario for each persona. Then, identify the capabilities that the user(s) in each use scenario will need.

Process

As announced in class, to complete this assignment you will work in a *team of four students of your own choosing*. If you would like, your team may be the same as the one in which you worked for Assignment 2, but need not be.

By Tuesday, November 20, 11:59pm, one team member should submit your team name and the names of its members to the IAs via email cc'ed to the other team members. Please let us know in advance if you would like help in assembling a team!
By Thursday, November 29, 11:59pm, your team should submit your proposed design concept—that is, the description of your personas and use scenario(s), and your design sketches—in pdf format, to CourseWorks. This document should include:

- A brief (one paragraph) description of your class of users.
- A detailed description of your two or more personas (one paragraph per persona).
- Your two or more use scenarios, at least one for each persona. Each use scenario should describe in one or more paragraphs how that persona would complete some task and be accompanied by one or more design sketches

Throughout the project, in performing your work, you should choose from among the full range of design and development approaches that we explored in class and in earlier assignments, including (but not limited to): design sketches, storyboards, lo-fi and hi-fi prototypes, and testing by users other than your team members, which you may use in any combination that you believe is appropriate. You will need to document your choice and use of these techniques as part of your submission. To make this easier, we will be using the myBalsamiq web-based prototyping tool. There is a Columbia University myBalsamiq site at which we will create a project for your team after you have provided us with your team's name and members. Since the site was created using an academic license, your team will use it for free.

Final Submission

In addition to your working program, implemented using HTML5, CSS, JavaScript, and the Giphy API (possibly along with additional APIs from other sources), your team's final submission should include in a folder the following items, each of which should contain your team name/member names/UNIs:

- A development document.
- A brief user manual.
- A video (or file containing the URL of an unlisted YouTube video that is dated no later than the deadline) showing your application in action.
- A screenshot, overview paragraph, and (optional) use permission

Each of these is described in more detail below.

Working program

Your program will be graded based on what it does (how it meets the very high-level "functional spec" presented in this assignment, and the description provided by your user manual) and how it does it (the user experience/program usability, measured in part against the Nielsen heuristics and the needs of the users whom you are targeting). We want your code to be well-structured, readable, and commented. (Note: While the "Consistency and standards" heuristic might appear to imply that you should match the Giphy website, we do not want you to do that for this assignment. Please use a different look and feel.)

Development document

This document should list your team name and member names, and UNIs. Its target audience is the course staff (in an industrial environment, it would be your manager and peers). The development document should include information about the following:

- *Overall process*. At a high level, describe your process from start to finish. Include roles and responsibilities of each team member.
- *Target users.* Describe your target users and the tasks in which they are interested, including descriptions of your personas and use scenarios, which you have refined based on our feedback.
- *Design decisions.* Explain the design decisions you made, including ones that were intended to address the specific needs of your target users and tasks. Your goal is to justify what you did. Please reference the Nielsen heuristics, and describe how your decisions satisfy them. In addition, you could provide a rationale for any of your decisions based on a user need or task. If parts of your user interface are inspired by or borrowed from any existing user interface designs or techniques, please cite them (whether or not they were covered in the course).
- *Design, prototyping, and testing process.* Provide a description of your design, prototyping and testing process. Diagrams and screenshots are strongly encouraged. Explain the design and prototyping techniques that you used and why you chose them, including your use of the myBalsamiq project that we created for your team. While testing with participants outside of your own team is encouraged, it is not required. However, you will need to explain how you tested your design and program (i.e., at least with members of your own team). Explain how the results of your design, prototyping and testing processes informed the final design of your program.
- *Software engineering*. Briefly describe the tools that you used, including any third-party code, libraries, or other resources (and please let us know about these in advance, so we can confirm that it will be OK to use them).

Brief user manual

This document should list your team name, names and UNIs. It should explain to an end user in your target user group how to use your program. Assume that the user *already* has basic familiarity with the web. Therefore, your user manual should address only the functionality

provided by your program. Please use screenshots, both by themselves (perhaps with annotations labeling parts of the user interface), and within storyboards that describe how to perform specific tasks. Be sure to explain the meaning of any metaphors, symbols, or color coding that you use and include any special instructions needed to run your program. (If your program has any problems that you know about or if you have not completed all of the requirements, please tell us the details here.)

The development document and brief user manual your team turns in should each be a .pdf file with embedded images. Please choose pdf-creation options that will help ensure that the images are legible. Please name your development document **yourteam_develop** and your user manual **yourteam_user** (where **yourteam** is the name you have given your team).

Video

You should create a brief (ideally at most five-minute long) video showing your project in action. Since you are creating a running application, please show one or more of your team members interacting with the application, acting as one or more of the personas in one or more of your use scenarios. You can capture a video of your system by using a phone or another computer's camera. (This puts no additional load on your computer, allowing it to run your application as fast as possible, and allows you to show both the screen and your user.) For parts of your video in which you would like to show the best quality screen image, you are also welcome, but not required, to use a screen capture system such as SnagIt (full functionality, 15-day free trial).

Please keep it simple: exotic visual and sound effects are neither needed nor desirable! Your video can be submitted as an unlisted YouTube video dated no later than the submission deadline

Screenshot, overview paragraph, and (optional) permission to use your names

Please provide a representative, full-resolution, screenshot of your project (png or jpg) and a brief, one-paragraph overview to include on our course project overview page. If you would like to have your names listed next to your screenshot on the overview page, please also give me your permission to do that in the form of the sentence: "[YOUR NAMES] are willing to have their names appear next to their presentation of their work on the project web page for COMS W4170."

When everything is done, please create a .zip file named **yourteam_proj.zip**, where **yourteam** is your team name. This file should contain

1. A folder that includes everything related to your web application, including any of your images that are local files.
2. A folder containing your development document, user manual, video, screenshot, overview paragraph, and (optional) use permission.

Next, just to make sure you've done this right, move your .zip file to a temp directory, unzip the file, and check whether everything is there. (Your IAs are not going to be sympathetic to explanations that something accidentally got left out of your .zip file.)

Assuming that things are okay, now you're ready to submit, using CourseWorks under the group assignment that we will create for the project.

Presentation

Your entire team will give a presentation providing a brief overview of your design concepts and process, followed by or integrated with a demo of your working program. Presentations will be held <To be announced>. Refreshments will be provided, and we'd like each of you to be there for *all* the presentations. Your team will have a total of **9 minutes for your entire presentation**, so you'll need to practice in advance! We recommend that you begin by spending approximately 2 minutes providing an overview of your design concepts and process, 5 minutes on your working demo, and 2 minutes on questions. Note that it may be possible to interleave a discussion of the design concepts with the demo if you wish to structure your presentation that way.

You will be responsible for ensuring *in advance* that

- you bring a laptop to demonstrate your project,
- your presentation can be viewed using the projector, noting that you shouldn't count on the native resolution of the projector being more than 1024×768, and
- any audio input or output needed has been set up correctly.

Please do not wait until the day of the presentation to determine how to set up your laptop to output the right resolution video to the projector. When your team walks up to the podium to start your presentation, your application should already be running and your laptop should already be set to output at a resolution that the projector supports. (Note: Many of you have laptops whose native resolution is higher than 1920x1080, and which the projector might not be able to handle properly. If you do not make sure that you set your external video resolution to one that the projector supports, you *may* still be able to output displayable video, *but* there is a good chance that it will be *hard to read, incorrectly scaled,* and *partially cropped*.)

Hints

A major component of the grading process will be our heuristic analysis of your application, focusing on its usability. Please take into account the metrics and approaches that we have covered in class.

Please let us know in advance if you would like to use any third-party library that provides additional functionality, so we can confirm that it will be OK to use it. You should also clearly acknowledge the use of any third-party library in your documentation.

Look through the APIs carefully to make sure that you understand their structure, and appreciate what you will and will not need to use.

One way to cut down on the number of API calls to address rate limiting is to cache the API calls you make and the data that you receive in response to them. Then, when you are ready to make a new call, if it is the same as one you've cached, then just reuse the cached response;

otherwise make the call and add it and its response to the cache, deleting an older cached call/response if necessary. Thus, while debugging your code, you could repeatedly test your program while making few if any API calls, especially if you make the cache persist between sessions. (Here, it would be important to have a way to clear the cache!)

You may not use any late days for this project.