# Codegram

Software Architecture

February 26, 2024

Brae Webb

## 1 Brief

You are working on Codegram, a social media website where programmers can post screenshots of their favourite code to share with others. Naturally, developers want their code screenshots to look as good as possible, so many users are requesting image filter features to make their screenshots look more appealing.

You have been tasked to design the backend for this feature. The frontend team has developed the user interface. Images are already uploaded and accessible via a URL that is provided to your backend.

## 2 Requirements

1. The frontend should be able to make a HTTP request to the backend to apply a filter to an image.
2. The HTTP request should include the URL of the image to be filtered.
3. You can design the HTTP request however you like.
4. The response should be the filtered image, you do not need to re-upload the image.
5. The current filter options are:
    - **Brightness**: Increase or decrease the brightness of the image.
    - **Contrast**: Increase or decrease the contrast of the image.
    - **Saturation**: Increase or decrease the saturation of the image.
    - **Grayscale**: Converts the image to grayscale.
    - **Sepia**: Converts the image to sepia.
    - **Blur**: Blurs the image.

    But you should expect more filters to be added in the future.

## 3 Outline

### Introduction (5 minutes)

Introduction to the brief, what is a filter, what are the requirements, where to start.

### Design (10 minutes)

Individually or in pairs, sketch out a potential design for the backend. You can use any tools you like, but you should be able to explain your design to the class. Your design does not need to be complete or perfect, try to be creative so that we can discuss the pros and cons of various design options.

### Discussion (10 minutes)

With the class, present a few of the designs and discuss the pros and cons of each. Consider the following questions:

- Does the design meet the requirements?
- Which quality attributes are prioritised in this design?
- How would you extend this design to support more filters?
- Are there trade-offs in this design?

### Sketching (20 minutes)

Individually sketch out a basic implementation of your preferred design from the discussion. Your design sketch should include:

- A high-level overview of the system, including any architectural patterns used.
- A description of the format of the HTTP request and response.
- An example of a HTTP request and response.
- A free-form diagram that illustrates the communication between components of the system.

You may include any other details including pseudocode, class diagrams, etc. that you think are relevant.

### Optimisation (10 minutes)

Discuss how you would optimise your design to improve extensibility, performance, scalability, etc. in a real system. Consider the following questions:

- What are the time/computation consuming parts of your design?
- Are there any bottlenecks?
- Are there use cases of the system that you can optimise? e.g. what if users repost and filter the same image?
- How would you scale your design to support more users?
- Are there any security concerns?

## 4 Design Challenges

### Challenge 1: Malicious Images

Codegram routinely scans hashes of uploaded images to detect malicious images, such as uploads of PHP code. Users have used the filter feature to bypass this detection. How would you modify your design to prevent this?

### Challenge 2: Reordering Filters

Codegram users often apply the same filter in different orders. For example, they may apply the `grayscale` filter, then the `brightness` filter or the `brightness` filter, then the `grayscale` filter. How would you optimise your design to reduce the number of times the image is filtered? Could you conceive of filters that are not commutative (i.e. the order matters)?

### Challenge 3: Global Access

Codegram is a global service, and users are located all over the world. Some users have reported that the filter feature is slow. You have determined that the primary cause is the latency between the user and the server. How would you modify your design to reduce the latency?

## 5 Programming Challenge

In a programming language of your choice, attempt a basic implementation of your preferred design. Use any libraries you like to implement the image filters, e.g. Pillow, OpenCV, etc. You do not need to implement the HTTP request, you can simulate it by calling a function in your code, but creating Web APIs will be important in the course so this is good practice. Refer to the Week 1 practical [1] for an example of how to create a Web API.

## References

[1] B. Webb and E. Hughes, "Web application programming interface (api)," vol. 1 of *CSSE6400 Practicals*, The University of Queensland, February 2024. https://csse6400.uqcloud.net/practicals/week01.pdf.