UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS 301 Mini Project Phase 3

Sharing API

Team GAMMA

Participants	
Name and Surname	**E-mail Address**
Brett Fourie	u16024002@tuks.co.za
Jared Gratz	u16054972@tuks.co.za
Jesse Mwiti	u17306192@tuks.co.za
Jordan Nijs	u18305980@tuks.co.za
Stephen Du Plessis	u17049203@tuks.co.za
Thembinkosi Mtsweni	u16205457@tuks.co.za
William Tandwe	u15232672@tuks.co.za

1 Introduction

For our Sharing API we have decided to make use of PHP and JavaScript to implement our API. The user is able to make uploads and request download to our database which we have implemented using MySQL. The user will send a POST request to our API to download a mouth piece packet from our database and we will send the compressed images using JSON. When the user uploads a mouth piece packet to the database the images will be compressed using Lossy compression which will change the mouth piece dimensions to 200x100 and it is then stored on the server with the metadata of the mouth piece packet being stored in our database for later access to the image packet.

2 ERD Diagram

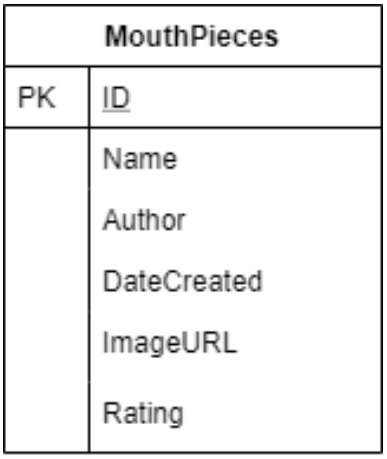

Figure 1: Mouthpiece ERD Diagram

2.1 ERD Description

The ERD we will be using will consist of 6 columns with ID acting as the primary key, which is used to uniquely identify each packet of mouth pieces. The mouth pieces table also include a name which is whatever the creator has named their packet. The Author column represents the creator of the uploaded mouth piece packet and DateCreated represents when the mouth piece packet was uploaded. The ImageURL is the most important column as it specifies the URL of where the mouth piece packet is stored which is how we will access the mouth piece packets and send it to the user when they request to download it. The Rating column displays the rating of the mouth piece packet given by other users who have used or viewed the specific mouth piece packet.

3 Optimization of Images

According to HTTP Archive, as of November 2018, images make up on average 21% of a total webpage's weight. This makes the compression of images an important component of any webpage to increase loading and request times. In our Mouthpiece application, images will be compressed using lossy compression before they are stored in the mouthpacks directory on the server. Lossy compression was chosen as greater decreases in file sizes are obtained while not affecting image quality greatly.

In order to retain uniformity and ease of use for all images stored in the database, they will first be resized to 200 by 100 and reformatted before being stored. This will be implemented with the use of a PHP library, Gregwar's Image class. An example of how it will be used:

```php
<?php use Gregwar\Image\Image;

Image::open('in.png')
     ->resize(200, 100)
     ->negate()
     ->save('out.jpg');
```

Figure 2: Gregwar's Image Class example

4 Apache (PHP)

Apache is an open-source and free web server software which allows website owners to serve content on the web. We will be saving our images on Apache which will be linking with our database. Only the user ID will be saved on our database. Thus our API will be pulling mouthpiece packet content from the Apache server on each user request and deliver it to the web and user profile depending on where is the user requesting.

5 Database Functionality

The user must name their creation and an ID will be generated automatically. The date of the upload will be stored in DateCreated and the author will be the user's username. If a user has not made an account yet the author will be "Anonymous". Moreover, the rating will be set to "Not rated" which will be updated once the mouth piece packet has been rated by other users.

References

[1] PHP.net blogger. Php: Apache functions-manual. Available at:https://www.php.net/manual/en/refs.remote.other.php. [Accessed March 12, 2020].

[2] Gregwar. Gregwar's image class[online]. Available at:https://github.com/Gregwar/Image. [Accessed March 11, 2020].

[3] Brian Jackson. How to optimize images for web and performance[online]. Available at: https://kinsta.com/blog/optimize-images-for-web/, August 2019. [Accessed March 11, 2020].