

# CircuitForge

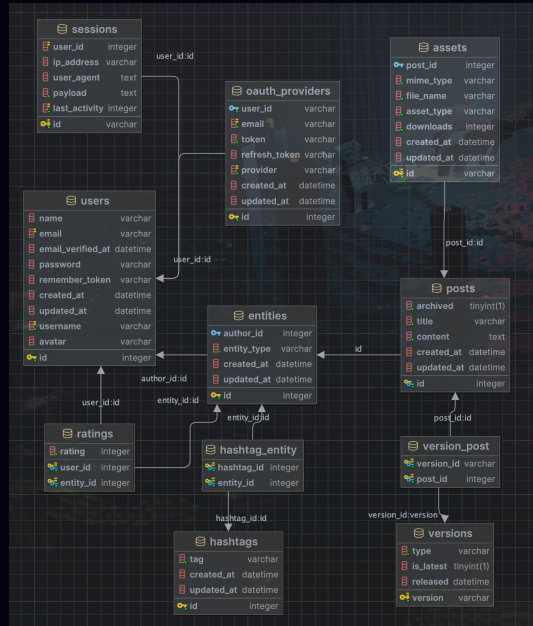
Jakob Huemer-Fistelberger 3AHTIM

# CircuitForge

Eine Plattform für das Teilen von Redstone-Technik und -Kreationen in Form von Posts.

Posts haben Bilder, eine Beschreibung und Downloads (Blaupausen).

# Datenbank



# Code Review: Asset Management

Speicherung von File in das System.

```
1 public static function makeAssetFromFile(UploadedFile $file, AssetType $type =  
  AssetType::ASSET): self {  
2     if (!$file->isFile()) throw new \Error("Uploaded file is not a file");  
3     $savedAsset = Asset::create([  
4         "file_name" => $file->getClientOriginalName(),  
5         "asset_type" => $type,  
6         "downloads" => 0,  
7         "mime_type" => $file->getMimeType(),  
8     ]);  
9     $savedAsset->save();  
10    Storage::put(ASSET_PATH . $savedAsset->id, $file->get());  
11    return $savedAsset;  
12 }
```

# Code Review: OAuth Authentication

## Simple OAuth Weiterleitung

```
1 public function addOAuthRedirectHandler(OAuthProviderType $providerType) {
2     return Socialite::driver($providerType->value)
3         ->redirectUrl(config("services." . $providerType->value . ".redirect_add
4     "))
5     ->redirect();
6 }
```

## OAuth Daten an Frontend senden

```
1 private function sendAuthLoginMessage(bool $success, string $error = ""):
2     RedirectResponse {
3     $response = [
4         "responseType" => ApiResponse::AUTH_LOGIN,
5         "data" => [
6             "success" => $success,
7             "error" => $error
8         ]
9     ];
10    $encodedResponse = base64_encode(json_encode($response));
11    return redirect()->away(env('FRONTEND_URL') . "api-handler#" .
12        $encodedResponse);
13 }
```

# Code Review: Version Seeder DB

Datenbank wird automatisch mit allen aktuellen Minecraft versionen befüllt

```
1 $data = $this->getJsonVersions();
2 $versionArr = $data->versions;
3 // release: {latest release}
4 // snapshot: {latest snapshot}
5 $latestVersions = array_values((array)$data->latest);
6 DB::table('versions')->truncate();
7 foreach ($versionArr as $value) {
8     DB::table('versions')->insert([
9         "version" => $value->id,
10        "released" => $value->releaseTime,
11        "type" => $value->type,
12        ...(in_array($value->id, $latestVersions) ? ["is_latest" => true] : []),
13    ]);
14 }
15 $this->command->info("Done seeding versions!");
```

