Lyon’s graffs

Table des matières

Remerciements1

Résumé du projet en français4

Résumé du projet en anglais4

Liste des compétences déployées4

Concept et Fonctionnalités attendues4

A. Concept

B. Fonctionnalités

Organisation et déroulement du projet4

A. Organisation

B. Outils mis en place

* 1. Méthode Agile
  2. Trello
  3. Github/Git

Conception Fonctionnelle et Technique4

A. Spécifications Fonctionnelles

* 1. Diagramme de Use Case
  2. Structure Fonctionnelle
  3. Diagramme d’interaction
  4. Maquettage de l’application

B. Spécifications Techniques

* 1. Choix techniques
  2. Front-End
  3. Back-End

Structure du projet4

* + - 1. Architecture
      2. Réalisation Technique

Remerciements

Résumé du projet en Français

Ayant essayé une multitude d’applications servant à découvrir de nouvelles œuvres de street art , j’ai été surprise de découvrir qu’aucune ne proposé aux utilisateurs de localiser eux même les œuvres qu’ils avaient découvert.

En effet les applications que j’ai jusqu’à présent trouvé proposées une carte générée par leur soin qui malheureusement n’était que très rarement à jour … il m’est arrivé de nombreuses fois de me rendre à un lieu indiqué et que l’œuvre ai disparue ou cette fonctionnalité payante. Le street art étant par essence un dont que fait un artiste à la collectivité en offrant aux yeux de tous une de ses œuvre, je me suis toujours refusée à payer pour ce genre de service …

C’est pourquoi pour ce Titre Professionnel, j’ai choisi de lancer un de mes objectifs personnels: créer une application dédiée aux œuvres de street art avec une géolocalisation renseignée par les utilisateurs.

Cette application prend la forme d’une application web responsive qui peut s’adapter à toutes les tailles d’écran. Les utilisateurs peuvent y créer un profil et publier des photos d’œuvres dénichées ainsi que leur localisation.

Cette application remplira les critères de compétences requises pour l’obtention de ce diplôme grâce aux fonctionnalités suivantes:

- une interface utilisateur permettant la création d’un compte et de publications,

-la possibilité pour un simple utilisateur de modifier/supprimer ses publications,

-la possibilité pour un administrateur de gérer les différentes catégories d’œuvres existantes (modifier ajouter supprimer)

-une authentification.

-l’utilisation de données enregistrées dans une base de données.

J’ai choisi de développer ce projet en Php dans un souci de cohérence entre mes expériences et l’uniformité des technologies utilisées. J’ai employé le framework PHP Laravel pour la partie Back de l’application, ainsi que d’autres outils de compilation et gestion de tâches. VueJS m’a permis de générer l’application côté client alors que Node/Express/MongoDB ont servi pour la création de l’API REST et délivrer les données. Il s’agit là d’outils et frameworks que je ne connaissais pas ou que je voulais approfondir avant de démarrer cette application. J’ai donc profité de ce projet pour découvrir ces nouveaux outils et élargir mes connaissances de manière générale.

Résumé du projet en Anglais

Liste des compétences déployées

1. Maquetter une application.

2. Développer une interface.

3. Développer une application simple de mobilité numérique

(responsive).

4. Concevoir une base de données.

5. Mettre en place une base de données

6. Développer des composants d'accès aux données

7. Développer des pages web en lien avec une base de

données.

Concept et Fonctionnalités attendues

A. Concept

B. Fonctionnalités

a. Affichage des images publiées

b. Formulaire de creation de compte/inscription

c.Formulaire de connexion

d. Page profile de l’utilisateur inscrit

e. Formulaire de création d’image et localisation

f. Page gallerie

g. Gestion differente des droits suivant rôle de l’utilisateur

Organisation et déroulement du projet

A. Organisation

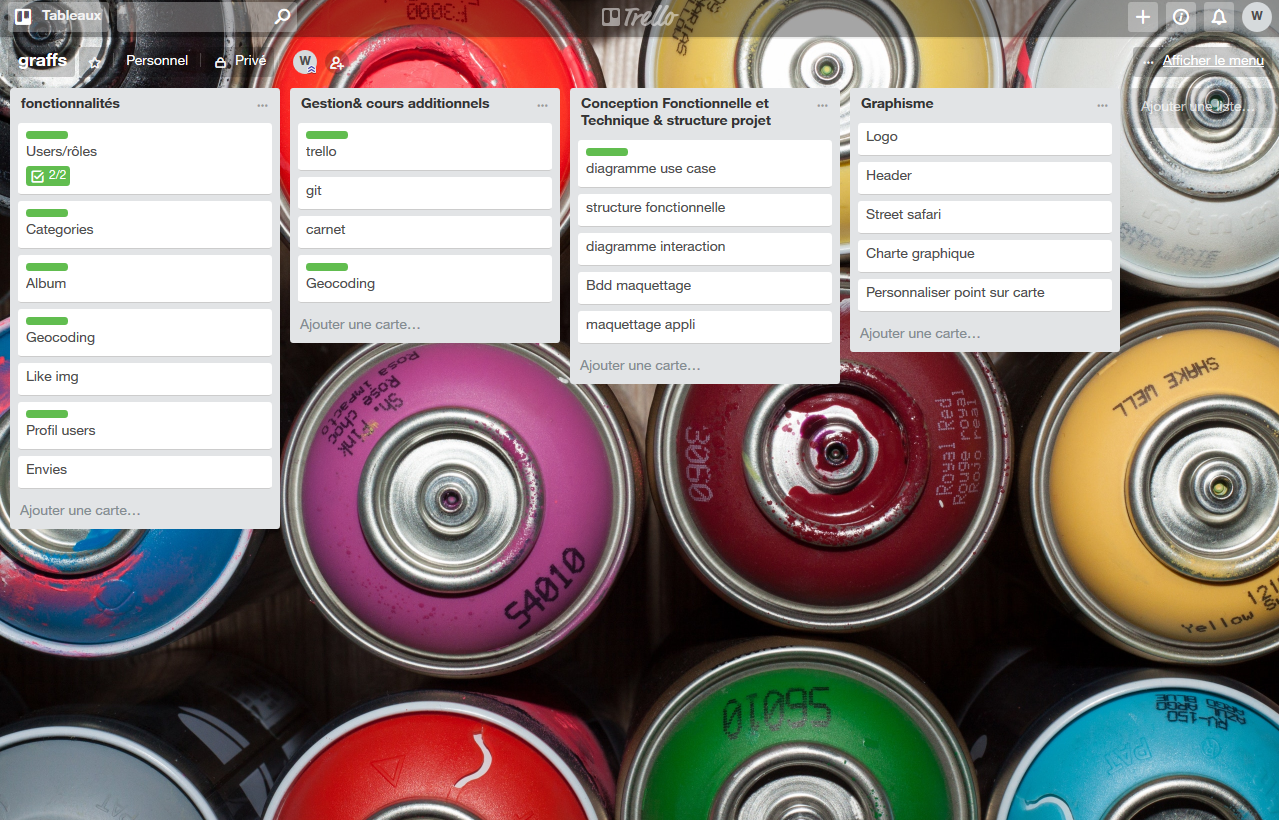
B. Outils mis en place

a. Méthode Agile

b. Trello

c. Github/Git

d. Carnet de route



b. Trello

**Trello** est un outil de gestion de projet en ligne, inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

Afin de voir plus clair dans l’avancée du projet j’ai utilisé l’outil Trello car il me permet de gérer de manière rapide et facile les listes de tâches que je souhaite effectuer. J’ai ainsi pu grouper les différentes tâches par type et ai rajouté une barre verte à chaque fois que l’une d’elle était accomplie.

Voici les listes créées :

* Fonctionnalités : listant les fonctionnalités que je souhaitais créer dans mon projet. Pour chacune d’entre elles j’ai effectué des recherches et ou suivit des cours en parallèle afin d’entre en mesure de les réaliser.
* Gestion & cours additionnels : listant les méthodes que je souhaitais mettre en place pour m’aider à organiser la création de ce projet ainsi que les notions que je devais acquérir afin de pouvoiur créer les fonctionnalités que je souhaitais réaliser.
* Conception Fonctionnelle et technique & structure du projet : Ce sont toutes les tâches ayant attrait à la conceptualisation de l’application avant de commencer à la coder.
* Graphisme : Liste des éléments visuels que je devais réaliser afin de rendre mon application plus esthétique.



c. Github

**GitHub** (exploité sous le nom de *GitHub, Inc.*) est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

Afin de conserver le code de l’application de manière sûre et maintenable, j’ai utilisé

l’outil de versionning Git et le projet a été envoyé vers la plateforme GitHub.

L’outil Git permet de sauvegarder différentes versions du code, sur l’ensemble du

projet ou pour chaque fichier individuel. Cette fonctionnalité permet de marquer les

différentes étapes du développement et de les communiquer aux autres

contributeurs (dans un projet de groupe) ou encore de revenir à un état précédent

lorsqu’un problème bloquant arrive par exemple.

Cet outil permet également de créer des branches, c’est à dire une version parallèle

du projet sur laquelle on pourrait développer une fonctionnalité particulière avant

de l’intégrer au corps du projet.

Il y a de nombreuses autres fonctionnalités inhérentes à l’outil Git qui permettent de

gérer l’agrégation du code, vérifier des changements et bien plus encore.

La plateforme Github offre, quant à elle, de nombreux outils supplémentaires

notamment sauvegarder son code en ligne en cas de perte, publier le code (sauf en

cas de dossier privé), visualiser des statistiques et graphiques ainsi que les

informations sur les différents ajouts et contributions au projets.

Ces outils m’ont permis de gérer mes ‘sprints’ au niveau de la gestion du code au

travers des ‘commits’ (point de sauvegarde d’une version du projet/fichiers

modifiés). Par exemple, un commit était créé lorsqu’une fonctionnalité était

achevée ou lorsque une base de travail était posée. J’ai également pu conserver

une sauvegarde de mon code en cas de perte.

* 1. Carnet de route

Le Bullet Journal (appelé aussi le « bujo ») est un **système d'organisation** inventé il y a quelques années par Ryder Carroll, un designer américain. Contrairement aux applications disponibles sur smartphone, cette méthode révolutionnaire nécessite seulement un carnet et un crayon.

N’étant pas à la base quelqu’un d’organisé et en plus très tête en l’air j’ai découvert cette méthode il y a quelques mois et cela m’a grandement aidé dans la vie de tous les jours à accomplir mes objectifs et ne plus oublier les choses à faire. C’est donc tout naturellement que je me suis tournée vers elle pour m’aider dans l’organisation de la création de mon projet tant au niveau de la conceptualisation que la création à proprement parler.

En effet j’ai pris pour habitude de toujours emmener mon carnet avec moi ce qui m’a permis d’y noter toutes les idées qui me venaient à n’importe quel moment de la journée, cela m’a aussi permis d’emmener les choses qui me posées problèmes ce qui me permettait d’y réfléchir à tête reposée.

De plus le fait de tenir et surtout cocher des to do liste a eu un effet hyper motivant sur moi dans le sens où je voyais que mon projet avancé même dans les moments où cela s’est fait à petit pas.

Conception Fonctionnelle et Technique

A. Spécifications Fonctionnelles

a. Diagramme de Use Case

Afin de mieux comprendre le fonctionnement de l’application, voici un diagramme

de cas d’utilisation (use-case) qui exprime les différents aspects auxquels doivent

avoir accès les utilisateurs.

Sur Lyon’s graffs, plusieurs model d’utilisateur n’ayant pas accès aux mêmes fonctionnalités. La seule plus grosse différenciation sera entre un visiteur non enregistré et un titulaire d’un compte.

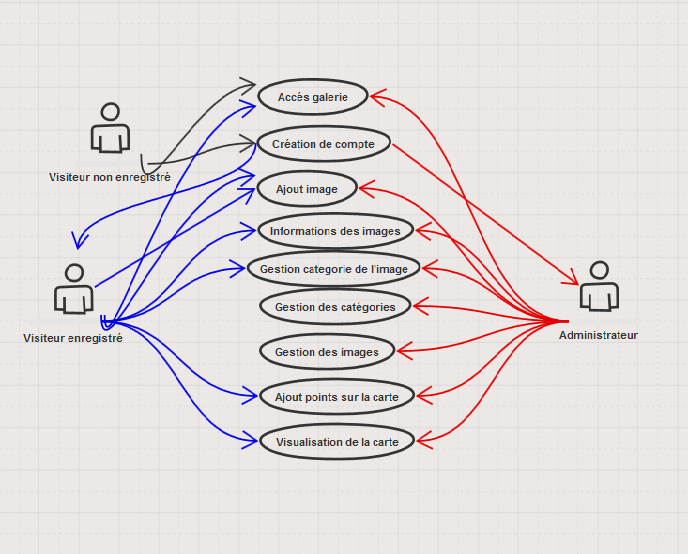
L’utilisateur non connecté(ou visiteur) ne pourra lui que visionner les images, et avoir accès à la zone de création de compte/ zone d’identification mais ne pourra pas interagir avec le site.

Il ne pourra pas ajouter ni d’image ni de localisation d’œuvres de street art avant de devenir un utilisateur enregistré.

L’utilisateur connecté pourra lui avoir accès aux galeries, à l’ajout de photos : localisation d’œuvres de street art, les informations de chaque œuvres, toutes les images d’un utilisateur de son choix, la gestion des photos qu’il a publiées et la visualisation de la carte recensant les œuvres de street art sur Lyon. Ainsi que le traditionnel espace de login in.

L’administrateur lui aura accès à toutes les fonctionnalités dont joui l’utilisateur connecté et pourra aussi modifier, supprimer ou ajouter les catégories de la galerie, il sera aussi le seul à pouvoir décider de supprimer la photo d’une œuvre définitivement.

1. Diagramme de cas d’utilisation (use-case)



Lyon’s graffs

1. Structure Fonctionnelle

En fixant pour objectifs que l’application fournissent les fonctionnalités attendues et présentes dans le Diagramme de Use-case (page précédente), les pages suivantes ont été conçues :

* Page 1 :
* Page 2 :
* Page 3 :
* Page 4 :
* Page 5 :
* Page 6 :
* Page 7 :

1. Diagramme d’interaction

Ensuite, ce diagramme illustre le parcours type de l’utilisateur sur le site et les

interactions qui lui seront présenté lors de sa visite sur Lyon’s graffs.

1. Maquettage de l’application

B. Spécifications Techniques

a. Choix techniques

b. Front-End

c. Back-End

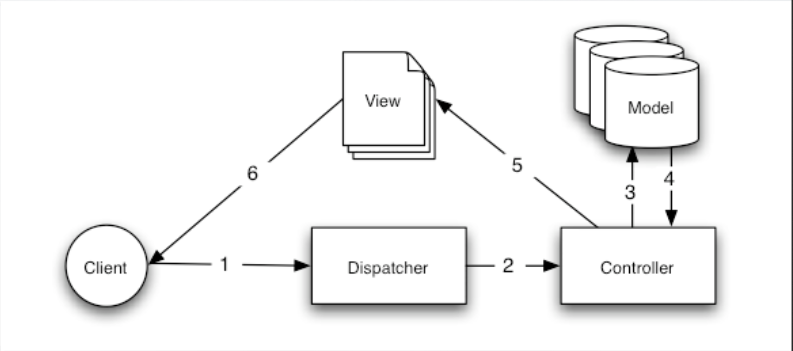
Structure du projet

1. Architecture

**Modèle-vue-contrôleur** ou **MVC** est un motif d'[architecture logicielle](https://fr.wikipedia.org/wiki/Architecture_logicielle) destiné aux [interfaces graphiques](https://fr.wikipedia.org/wiki/Interface_graphique) lancé en 1978 et très populaire pour les [applications web](https://fr.wikipedia.org/wiki/Application_web). Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

* Un modèle (Model) contient les données à afficher.
* Une vue (View) contient la présentation de l'interface graphique.
* Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

Ce motif est utilisé par le framework Laravel avec lequel j’ai choisi de travailller.



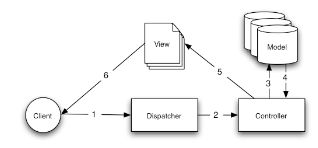
Organisation des fichiers :

• 1 fichier global pour le Dispatcher (la route « route.php »)

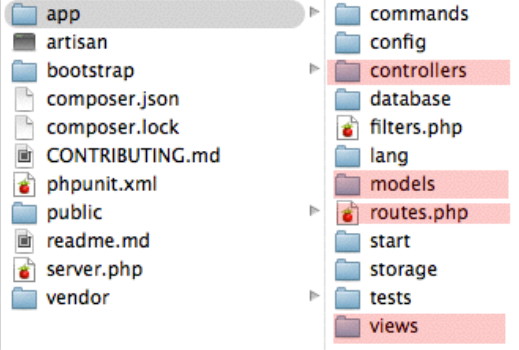
• 1 fichier par vue « mavue.php »

• 1fichier par contrôleur « moncontroleController.php »

• 1 fichier par model « Model.php »



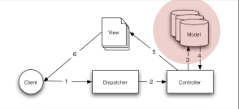
Emplacement

de ces fichiers

dans Laravel :

MVC :

• **le fichier Modele,**



• il permet de décrire la méthode d’accès aux

données de la base. tous cela à travers un

objet.

• attention aux règles de nommage! (Laravel se

sert de ces regles pour trouver les tables et les

clés. exemple: une clé étrangère dans une table

s’écrit « nom de la table »\_id , la clé primaire

d’une table doit impérativement s’appeler

« id »)

**• ORM Eloquent**

• Object-Relational Mapping, permet de voir

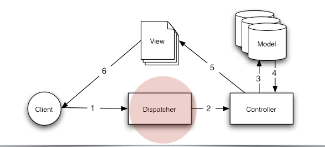
notre base de données relationnelle comme

un objet

* pour pouvoir accéder à votre

base de données il faut

parametrer le fichier de config

**• la route:**

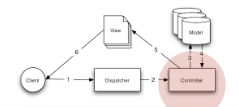
• elle permet de décrire la relation entre l’url

tapée par le client et le contolleur à lancer

• ici le client demande « catalogue ».

• dans la route « catalogue » lancera la méthode «index» du controleur «CarticlesController»

• **le contolleur**

• il permet de récupérer les

informations du modele et de

l’envoyer vers la vue pour la

mise en forme.

• « Article::all() » récupére

toutes les données du modele

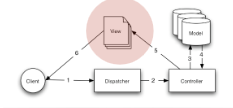
« Article »

• make(« Varticle ») appelle la

vue « Varticle »

• ->with(‘articles’,$articles) passe

toutes les données à la vue



• la vue (ici une structure minimaliste avec

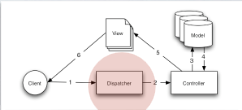
boucle en php) « Varticle.php ». ici du

php dans la vue…. à éviter, nous verrons

plus loin l’utilisation de « Blade » pour

restructurer cela.

• $articles est la variable envoyée depuis

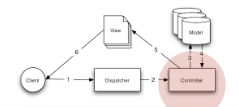
le controleur

• Récupération de paramètre

d’URL (ex résultat d’un

formulaire renvoyant ses infos via

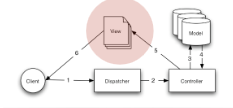
l’URL)



• declaration des paramètres

d’url dans la route

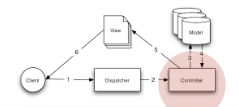
•Moteur de template «Blade»:

•création d’un modele de template du site

pour toutes les pages

(«template.blade.php»)

•création de la vue «Vcatalogue.blade.php» qui fera appel au template global



•appel de cette nouvelle vue dans le

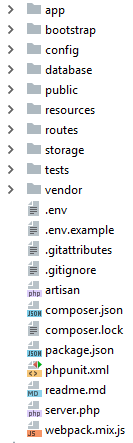
contrôleur

1. Réalisation technique
2. L’installation

Pour l’installation c’est tout simple, je me suis servie de la commande :

composer create-project laravel/laravel graffs --prefer-dist

Ce qui m’a permis de voir se créer ces dossiers :



Et cet aspect sur mon navigateur :



Pour compléter l’installation, j’ai aussi installer l’authentification grâce à ces commandes :

cd graffs

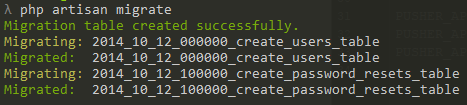
php artisan make:auth

Puis j’ai créer ma base de données grâce à PhpMyadmin que j’ai intitulé album et paramétré le fichier .env .

J’ai ensuite lancé mes migrations à l’aide de la commande :

php artisan migrate

Afin de créer mes trois premiéres tables :



Ce qui m’a permis de faire apparaitre sur mon navigateur une barre pour l’authentification :

https://laravel.sillo.org/wp-content/uploads/2017/11/Capture-36.png

# Pour ajouter la langue française, dans le dossier lang de mon projet j’ai ajouté le package Laravel-lang à l’aide de composer : https://laravel.sillo.org/wp-content/uploads/2017/11/Capture-40.png

composer require caouecs/laravel-lang

Et ai changer cette ligne dans le fichier config/app.php :

‘locale’=>’fr’

1. le style