





# Table des matières

Introduction	4
Rappel du cahier des charges	4
But de l'application	4
Fonctionnalités à réaliser	4
Matériels et logiciels nécessaires	4
Technologie à utiliser	4
Livrables	4
Méthodologie	4
Analyse fonctionnelle	4
Fonctionnalités implémentées	4
Interfaces	4
Page d'accueil (connexion)	4
Page Principale	4
Page modela animal	4
Page adoption	4
Page propriétaire	4
Analyse organique	4
Architecture	4
Modélisation de la base de données	4
Modèle conceptuel de données (MCD)	4
Modèle logique de données (MLD)	4
Dictionnaire de données	4
Modèle physique de données (MPD)	5
Arborescence	5
Description des fichiers	5
Script SQL	5

Modèles	5
Vues	5
Contrôleurs	5
Point d'entrée de l'application	5
Description des classes	5
Classe Database	5
Classes « Tables »	5
Classes « Records »	5
Librairies / APIs	5
Geoapify Leaflet	5
Bootstrap	5
Améliorations possibles	5
Tests	5
Plan de tests	5
Rapport de tests	5
Durant l'implémentation (20.04 – 08.05)	5
A la fin de l'implémentation (13.05)	5
Conclusion	5
Bilan personnel	5
Table des illustrations	5
Liens et références	5
Documentations	5
Règles de nommages	6
Aides spécifiques	6
Annexes	6
Planning prévisionnel	6
Planning effectif	6
Code source	6

# Introduction

Ce document représente la documentation du projet réalisé dans le cadre de mon Travail Pratique Individuel pour l'obtention de mon CFC d'informaticien, et est destiné aux experts en charge d'évaluer mon travail ainsi qu'à ma formatrice. Il porte sur une application web d'adoption d'animaux intitulée «AdoptMe». Le document comprend différents chapitres visant à fournir une compréhension approfondie du processus de développement, des résultats obtenus et du fonctionnement du projet, comme le cahier des charges, une analyse fonctionnelle et organique ainsi qu'un plan et un rapport de tests.

## Rappel du cahier des charges

### But de l'application

J'ai été mandaté pour créer l'application web d'un refuge pour animaux qui permet de recenser tous les animaux de ce refuge ainsi que les adoptions d'animaux effectués au sein de celui-ci.

### Fonctionnalités à réaliser

- Une fenêtre d'accueil de l'application avec un formulaire de connexion (afin que l'employé bénévole du refuge puisse s'y connecter) ;
- Une fenêtre affichant la totalité des animaux du refuge
  - Une fenêtre modale permettra d'ajouter un animal, de le modifier ou de le supprimer (cette fenêtre s'affiche par le biais d'un bouton sur lequel l'utilisateur peut cliquer dans la fenêtre n°2) ;

- Une fenêtre permettant d'enregistrer l'adoption d'un animal par un propriétaire (cette fenêtre s'affiche par le biais d'un bouton sur lequel l'utilisateur peut cliquer dans la fenêtre n°2) ;
  - Une fenêtre modale permettra d'ajouter un propriétaire, de le modifier ou de le supprimer (cette fenêtre s'affiche par le biais d'un bouton sur lequel l'utilisateur peut cliquer dans la fenêtre n°2.b).
- La fenêtre n°1 peut mener à la fenêtre n°2. La fenêtre n°2 peut mener à la fenêtre n°2.a modale de gestion d'un animal (insertion, modification ou suppression) ou à la fenêtre n°2.b. La fenêtre n°2.b peut mener à la fenêtre n°2.b.i. Il est possible de se déconnecter de l'application ou de retourner à la fenêtre antérieure (ou fenêtre appelante) à tout moment.

## Matériels et logiciels nécessaires

- Un pc standard école, 1 écran;
- Windows 10;
- Suite bureautique Microsoft Office
- Visual studio code
- Workbench
- Server de base de données MySQL

## Technologie à utiliser

Créer une application web avec les outils et technologies web utilisés à l'école (WSL2, MariaDB, HTML, CSS, Bootstrap, JavaScript et PHP) tout en respectant le design pattern MVC ;

## Livrables

- Planning
- Rapport de projet
- Le code source de l'application
- Manuel utilisateur (si applicable)
- Journal de travail

# Méthodologie

Pour le bon fonctionnement de mon projet il m'a fallu mettre en place une méthodologie de travail pour être organisé et efficace. J'ai donc pris la décision de suivre la méthode en 6 étapes. J'ai choisi celle-ci car parmi toutes les méthodes enseignées durant la formation la méthode en 6 étapes est la plus élaborée pour un travail individuel à court terme. Mais ce n'est pas tout, cette méthode en 6 étapes regroupe d'autres méthodologies de planification. Notamment, le backlog et les user story qui sont tirés de la méthodologie agile ou encore un diagramme Excel pour la planification temporelle des différentes tâches. De plus, j'ai aussi réalisé des schémas de base de données avec le MLD et MCD.

## 6 étapes

Cette méthode fonctionne sous différents points :

### 1. S'informer

La première étape du processus est axée sur la collecte de toutes les informations essentielles relatives au projet. Pour cette raison, j'ai effectué une lecture approfondie du cahier des charges ainsi que des fonctionnalités nécessaires à réaliser. Toutefois, pendant cette lecture, certains éléments étaient imprécis, ce qui m'a conduit à poser plusieurs questions à ma responsable, Mme Moro.

### 2. Planifier

Une fois que toutes les informations ont été collectées, il m'a fallu procéder à la planification. Pour ce faire j'ai dû découper mon cahier des charges et en réaliser des tâches sous forme d'user stories.

Chaque user stories a des priorités tirées de la Méthode MoSCoW, ce procédé me permet d'être clair sur les tâches à faire rapidement pour éviter d'être bloqué.

Les priorités sont :

□□ p1 : Indispensables, ✖ p2 : Importants, Ɔ-Ɔ p3 : Secondaires, □□ p4 : Luxe.

J'ai créé plusieurs user stories qui ont été rassemblées dans un product backlog.

Ce dernier est utilisé dans le cadre de méthodes de gestion de projets agiles, telles que Scrum, afin de planifier et d'organiser les travaux à effectuer par l'équipe de développement.

Pour mieux organiser mon projet, j'ai réalisé un diagramme Excel qui me permet de visualiser de manière claire et graphique l'ensemble des tâches à réaliser sur les onze jours que dure le TPI. Dans ce diagramme, chaque tâche est représentée par une barre horizontale dont la longueur correspond à la durée de la tâche et dont la position sur l'axe vertical indique sa place dans la hiérarchie des tâches. Le diagramme Excel est un outil très pratique de gestion de projet, car il permet de mieux comprendre les différentes étapes du projet, de suivre l'avancement des tâches et de prévoir les échéances à venir.

Au début du projet, j'ai établi un planning prévisionnel pour planifier les différentes étapes et définir un calendrier approximatif pour chacune d'elles et que j'ai transmis aux experts et à ma responsable le premier jour. Cela m'a permis d'avoir une feuille de route pour réaliser chaque étape du projet.

Cependant, une fois que le planning prévisionnel a été établi, je ne l'ai plus



modifié sauf sur demande de Mme Moro.

Au lieu de cela, j'ai régulièrement mis à jour le planning effectif tout au long du projet pour suivre et contrôler l'avancement réel des différentes étapes du projet.

### 3. Décider

À cette étape, les décisions sont prises en fonction des informations collectées et du plan d'action élaboré. Cela peut impliquer la sélection d'une option ou d'une solution à mettre en œuvre, ou la prise d'une décision concernant la mise en place du plan d'action.

### 4. Réaliser

Cette étape cruciale représente l'un des moments clés de la mise en œuvre de mon projet, car elle implique la concrétisation des décisions prises lors de l'étape précédente. Elle nécessite ainsi une grande rigueur et une planification minutieuse pour assurer la réussite de la mise en œuvre du plan d'action élaboré.

Grâce aux tâches que j'ai planifiées j'ai pu réaliser plusieurs actions pour faciliter le développement. Pour commencer, j'ai réalisé le modèle conceptuel de données et le modèle Logique de données pour la base de données de l'application.

Ensuite, j'ai pu effectuer un mood board et les maquettes de l'application qui sont en annexe du rapport.

Et pour finir, grâce à ce que je viens de faire et les différentes planifications, j'ai pu réaliser mon application dans de bonnes conditions.

## 5. Contrôler

L'étape de suivi est primordiale pour s'assurer que le plan d'action est correctement mis en œuvre et que les objectifs fixés sont atteints. Ainsi, à chaque fois que je terminais une tâche, j'ai effectué des tests fonctionnels pour évaluer la nouvelle fonctionnalité implémentée ainsi que son intégration au sein de l'application. Cette étape est essentielle pour détecter les éventuelles erreurs ou dysfonctionnements, et pour y remédier rapidement afin de garantir la qualité de l'application. Le suivi régulier du progrès de la mise en œuvre du plan d'action permet également de prendre conscience de son degré d'avancement dans le projet.

## 6. Évaluer

La dernière étape de la méthodologie en 6 étapes consiste à évaluer les résultats obtenus et à réfléchir aux moyens d'améliorer les futures décisions et actions.

Pour ce faire, j'ai tenu un journal de bord où j'ai rédigé des bilans journaliers pour avoir une vision critique de mon travail et détecter les erreurs commises.

# Analyse fonctionnelle

## Fonctionnalités implémentées

### Interfaces

Page d'accueil (connexion)

Page Principale

Page modale animal

Page adoption

Page propriétaire

# Analyse organique

## Architecture

## Modélisation de la base de données

Modèle conceptuel de données (MCD)

Modèle logique de données (MLD)

Dictionnaire de données

Modèle physique de données (MPD)

## Arborescence

## Description des fichiers

Script SQL

Modèles

Vues

Contrôleurs

Point d'entrée de l'application

## Description des classes

Classe Database

Classes « Tables »

Classes « Records »

## Librairies / APIs

Geoapify

Leaflet

Bootstrap

## Améliorations possibles

# Tests

## Plan de tests

Le plan de test couvre la vérification des fonctionnalités principales de la gestion des animaux dans le refuge, notamment l'affichage, l'ajout, la modification, la suppression d'animaux et l'enregistrement d'une adoption, mais aussi, la connexion au site.

## Environnement de tests

- Système d'exploitation : Windows 10 / Ubuntu 20.04
- Navigateur : Brave v 1.74.50
- Serveur : Apache 2.4 / PHP 8.1
- Base de données : MySQL 8.0
- Dépendances : Bootstrap 5, PHP MVC Framework

## Scénarios (les tests, les données et leurs évolution)

Nom	1.1 Connexion à l'application
User story	Connexion à l'application
Situation	L'admin doit pouvoir se connecter à l'application via un formulaire de

	connexion en saisissant son <b>**Nom d'utilisateur**</b> et son <b>**mot de passe**</b> . Une validation doit être effectuée pour s'assurer que les identifiants sont corrects et existent dans la base de données.
Résultats obtenus	<ul style="list-style-type: none"> <li>- Si les identifiants sont corrects, l'admin est redirigé vers la page d'accueil.</li> <li>- Si les identifiants sont incorrects, un message d'erreur est affiché.</li> </ul>
Statut	OK

Nom	<b>2.1 Affichage de la liste des animaux</b>
User story	Voir la liste des animaux du refuge
Situation	L'admin connecté doit voir la liste des animaux disponibles avec leurs informations (id, nom, date de naissance, sexe, espèce).
Résultats obtenus	- Tous les animaux présents dans la base de données sont affichés.
Statut	OK

Nom	<b>2.2 Ajouter un animal</b>
User story	Ajouter un nouvel animal au refuge
Situation	L'admin clique sur "Ajouter un animal", remplit le formulaire et valide.
Résultats obtenus	- L'animal est ajouté dans la base et apparaît dans la liste immédiatement.
Statut	OK

Nom	2.3 Modifier un animal
User story	Modifier les informations d'un animal
Situation	L'admin sélectionne un animal et modifie ses informations via la modale de modification.
Résultats obtenus	<ul style="list-style-type: none"> <li>- Les nouvelles informations sont mises à jour dans la base de données.</li> <li>- La liste des animaux affiche les changements en temps réel.</li> </ul>
Statut	OK

Nom	2.4 Supprimer un animal
User story	Supprimer un animal du refuge
Situation	L'admin clique sur "Supprimer".
Résultats obtenus	<ul style="list-style-type: none"> <li>- L'animal est supprimé de la base de données.</li> <li>- Il n'apparaît plus dans la liste.</li> </ul>
Statut	OK

Nom	3.1 Enregistrement d'une adoption
User story	Enregistrer l'adoption d'un animal
Situation	L'admin sélectionne un animal, clique sur "Adopter" et assigne un propriétaire existant, l'employé responsable et la date d'adoption.
Résultats obtenus	<ul style="list-style-type: none"> <li>- L'animal adopté se montre dans la base.</li> <li>- Il n'apparaît plus comme disponible dans la liste.</li> </ul>

Statut	OK
--------	----

Nom	<b>4.1 Ajouter un propriétaire</b>
User story	Ajouter un propriétaire
Situation	L'admin ouvre la modale des propriétaires et enregistre un nouveau propriétaire.
Résultats obtenus	<ul style="list-style-type: none"> <li>- Le propriétaire est ajouté à la base de données.</li> <li>- Il est visible dans la liste des propriétaires.</li> </ul>
Statut	OK

Nom	<b>5.1 Déconnexion de l'application</b>
User story	Déconnexion de l'admin
Situation	L'admin clique sur "Se déconnecter".
Résultats obtenus	<ul style="list-style-type: none"> <li>- La session est détruite.</li> <li>- L'utilisateur est redirigé vers la page de connexion.</li> </ul>
Statut	OK

## Evolution des tests

N° test	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11
1.1	✗	✗	✗	✓	✓	✓	✓				
2.1	✗	✗	✗	✗	✓	✓	✓				

2.2	✗	✗	✗	✗	✓	✓	✓				
2.3	✗	✗	✗	✗	✗	✓	✓				
2.4	✗	✗	✗	✗	✗	✓	✓				
3.1	✗	✗	✗	✗	✗	✗	✓				
4.1	✗	✗	✗	✗	✗	✓	✓				
5.1	✗	✗	✗	✗	✓	✓	✓				

## Conclusion

Bilan personnel

## Table des illustrations

## Liens et références

Documentations

Règles de nommages

Aides spécifiques

## Annexes

Planning prévisionnel

Planning effectif

Code source