

# DOCUMENTATION TECHNIQUE

## 1-REFLEXION INITIALE TECHNOLOGIQUE

### Analyse de l'outil technologique et choix du Framework

Une réflexion approfondie sur l'outil technologique a été menée lors du développement de la **partie back-end**. L'utilisation du Framework **Symfony** pour la gestion de la base de données (BDD) a soulevé la question de l'étendre à l'ensemble du projet.

Après des recherches approfondies, il a été décidé d'adopter **Symfony et Twig** également pour le développement de la **partie front-end**. Ce choix s'inscrit dans le respect du modèle MVC, favorisant une architecture logicielle claire et maintenable.

## 2-CONFIGURATION ESPACE DE TRAVAIL

### Environnement de développement et outils

Le développement du projet a été réalisé au sein de Visual Studio Code, en s'appuyant sur les extensions suivantes :

- **Symfony for Vs code**
- **Php intelephense**
- **Twig Language**
- **Github Copilot**

### Configuration technique

- **Version de PHP** : 8.3
- **SGBD relationnel** : MySQL Workbench 8.0
- **Framework** : Symfony 7
- **Base de données NoSQL** : MongoDB et MongoDB Compass
- **Plateforme de déploiement** : Heroku (facilitant la connexion avec MongoDB)

**Remarques :**

- Le choix de Visual Studio Code et des extensions mentionnées s'explique par leur ergonomie et leur utilité pour le développement web avec les technologies employées.
- L'utilisation de PHP 8.3 garantit l'accès aux dernières fonctionnalités du langage.
- Le choix de MySQL Workbench 8.0 assure une gestion efficace de la base de données relationnelle.
- L'adoption de Symfony 7 permet de bénéficier d'un framework performant et évolutif pour le développement de l'application web.
- L'utilisation de MongoDB et MongoDB Compass offre une solution flexible et performante pour la gestion de données non structurées.
- Le déploiement sur Heroku facilite la mise en production et la maintenance de l'application.

### 3-DIAGRAMME D'UTILISATION



# 4-MODELE CONCEPTUEL DE DONNEES

www.quickdatabasediagrams.com

## Users

id	int
Email	string
Password	string

## Services

Id	int
title	string
description	string
image	string
serviceName	MANY_TO_MANY

## Avis

Id	int
Pseudo	string
Message	string

## Horaire

Id	int
Horaire	int
HoraireId	ONE_TO_ONE

## Habitats

id	int
Nom	string
Description	string
Image	string
ListeAnimaux	string
Commentairevet	string

## Animaux

prenom	string
race	string
image	string
habitatId	int

## EspaceEmploye

DateTime	datetime
animaux	string
quantite	string
datetime	datetime
animauxid	string

## EspaceVet

id	int
AnilauxId	string
Etat	string
Nourriture	string
DatevisiteVet	date
Race	string
image	string

+

+

+

+

## 5-DEPLOIEMENT HEROKU

### Choix de la plateforme de déploiement : Heroku

Heroku a été sélectionné comme plateforme de déploiement pour les raisons suivantes :

- **Facilité de déploiement des applications Symfony** : Heroku offre un processus de déploiement simple et intuitif, adapté aux applications Symfony.
- **Support des bases de données NoSQL** : Heroku permet l'intégration de bases de données NoSQL, telles que MongoDB, ce qui correspondait aux besoins du projet.

### Déroulement du déploiement

1. **Inscription via un compte GitHub** : L'utilisation d'un compte GitHub a facilité la connexion et l'authentification sur la plateforme Heroku.
2. **Fusion des branches master et dev** : La branche dev a été fusionnée dans la branche master avant le déploiement, garantissant une intégration cohérente du code.
3. **Déploiement sur Heroku** : Le déploiement de l'application sur Heroku a été effectué en suivant les instructions fournies par la plateforme.
4. **Mise en place du fichier Procfile** : Le fichier Procfile, manquant initialement, a été créé pour définir les instructions de démarrage de l'application.
5. **Exécution des migrations Doctrine** : Les migrations Doctrine ont été exécutées via l'application Heroku pour mettre à jour la structure de la base de données.
6. **Alimentation de la base de données** : Les données concernant les services, les habitats, les animaux et les utilisateurs (admin, employé et vétérinaire) ont été importées dans la base de données.
7. **Ajout des rôles d'accès** : Des rôles d'accès ont été définis pour sécuriser les différentes interfaces d'administration et limiter l'accès aux fonctionnalités en fonction du profil utilisateur.
8. **Tests finaux** : Des tests approfondis ont été menés pour vérifier le bon fonctionnement de l'application et garantir une expérience utilisateur optimale.