

# EXAMEN 5IDV2

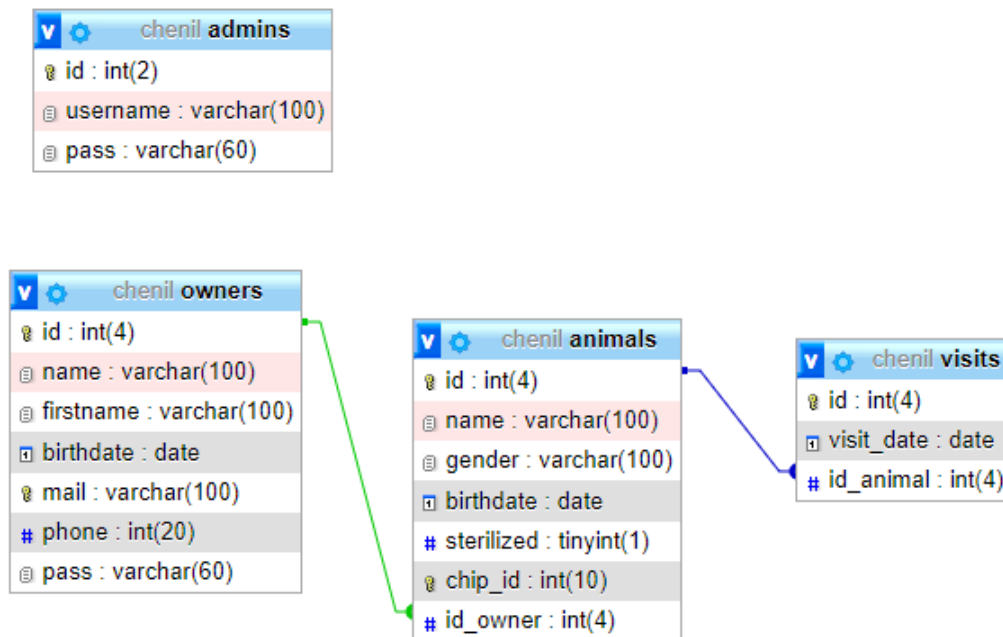
Juin 2022



Ifosup Wavre  
Lambert Thomas

Lambert  
Thomas

## Schéma relationnel de la base de données



## Liste des routes

POST /user/adminLogin

Paramètre	Type	Status	Description
username	string	Obligatoire	Username de l'admin Pour se connecter
pass	string	Obligatoire	Mot de passe de l'admin

POST /user/ownerLogin

Paramètre	Type	Status	Description
email	string	Obligatoire	Email du propriétaire pour se connecter
Pass	String	Obligatoire	Mot de passe du propriétaire

POST /user/addNewOwner

Paramètre	Type	Status	Description
name	string	Obligatoire	Nom du propriétaire à ajouter
firstname	String	Obligatoire	Prénom du propriétaire à ajouter
date	String	Obligatoire	Date de naissance du propriétaire à ajouter

Lambert  
Thomas

mail	String	Obligatoire	Mail du propriétaire à ajouter
phone	Int	Obligatoire	Numéro de téléphone du propriétaire à ajouter
pass	String	Obligatoire	Mot du passe du propriétaire à ajouter
cpass	String	Obligatoire	Confirmation du mot de passe

POST /animal/destroy

Paramètre	Type	Status	Description
idAnimal	Int	Obligatoire	Id de l'animal à supprimer

POST /visit/destroy

Paramètre	Type	Status	Description
idVisit	Int	Obligatoire	Id du séjour à supprimer

POST /owner/destroy

Paramètre	Type	Status	Description
idOwner	Int	Obligatoire	Id du propriétaire à supprimer

GET /owner/showUpdate

Réponse	Type	Status	Description
Formulaire pour modifier un propriétaire	Html	200	Renvoie un formulaire pour modifier un propriétaire

GET /animal/create

Réponse	Type	Status	Description
Formulaire pour ajouter un animal	Html	200	Renvoie un formulaire pour ajouter un animal

GET /animal/one

Réponse	Type	Status	Description
Les infos d'un animal	Html	200	Renvoie les infos d'un animal

Lambert  
Thomas

GET /animal

Réponse	Type	Status	Description
La liste des animaux	Html	200	Renvoie la liste des animaux

POST /animal/addNewAnimal

Paramètre	Type	Status	Description
name	String	Obligatoire	Nom de l'animal à ajouter
gender	string	Obligatoire	Genre de l'animal à ajouter « F » ou « M »
birthdate	String	Obligatoire	Date de naissance d'un animal
sterilized	Int (bool)	Obligatoire	Boolean stérilisé ou pas « 1 » ou « 0 »
chip_id	Int	Obligatoire	Id de la puce d'identification de l'animal. Entre 0 et 10 caractères.
owner	Int	Obligatoire	Id du propriétaire de l'animal (FK)

GET /animal/edit

Réponse	Type	Status	Description
Formulaire de la modification de l'animal	Html	200	Renvoie le formulaire pour modifier un animal

GET /owner

Réponse	Type	Status	Description
La liste des propriétaires	Html	200	Renvoie la liste des propriétaires

GET /owner/one

Réponse	Type	Status	Description
Les infos d'un propriétaire	Html	200	Renvoie les infos d'un propriétaire

GET /owner/register

Réponse	Type	Status	Description
Le formulaire pour créer un compte (propriétaire)	Html	200	Renvoie le formulaire pour créer un compte

Lambert  
Thomas

			en tant que propriétaire
--	--	--	--------------------------

GET /owner/dashboard

Réponse	Type	Status	Description
Le tableau de bord d'un propriétaire	Html	200	Renvoie vers le tableau de bord du propriétaire connecté

GET /main

Réponse	Type	Status	Description
Le tableau de bord de l'admin	Html	200	Renvoie vers le tableau de bord de l'admin connecté

GET /main/login

Réponse	Type	Status	Description
La page login	Html	200	Renvoie vers la page de login

GET /visit

Réponse	Type	Status	Description
La liste des séjours	Html	200	Renvoie la liste des séjours

GET /visit/one

Réponse	Type	Status	Description
Les infos d'un séjour	Html	200	Renvoie les infos d'un séjour

GET /visit/create

Réponse	Type	Status	Description
Le formulaire pour ajouter un séjour	Html	200	Renvoie le formulaire pour ajouter un séjour

POST /visit/addNewVisit

Paramètre	Type	Status	Description
date	String	Obligatoire	La date du séjour à ajouter
animal	Int	Obligatoire	L'animal du séjour à ajouter (FK)

## Problèmes rencontrés

- Problème pour lier les scripts et le css dans certains fichiers. La solution était de rajouter un / avant le href. Je n'ai pas compris pourquoi ça marchait dans certains fichiers et pas dans d'autres.
- Pour l'implémentations du design pattern strategy j'ai eu un problème pour ma fonction login(). Je ne savais pas comment faire pour différencier un \$this->username quand c'est un objet Admin et un \$this->mail pour un objet Owner.

```
• public function login(){  
•  
•         if ($this instanceof Admin) {  
•             $login = $this->username;  
•         }elseif($this instanceof Owner){  
•             $login = $this->mail;  
•         }  
•  
•         $this->loginBehavior->login($login, $this->pass);  
•     }
```

Donatien Pauwel m'a aidé pour ça.

- Pas mal de problème dans le js. J'avais beaucoup de mal à sélectionner des class, surtout vers la fin quand il y en avait partout. C'est la première fois que je fais un aussi gros projet avec du js du coup j'ai vraiment eu du mal à gérer toutes les class de balise et les id. J'ai testé des trucs en faisant des « toggle ».
- Des fois les scripts ne se rechargeaient pas car il fallait que je fasse ctrl+f5, ça m'a fait perdre beaucoup de temps.
- J'ai sûrement eu d'autres problèmes mais je n'ai jamais été vraiment bloqué très longtemps.

## Fonctionnalités implémentées

- Les fonctionnalités obligatoires.
- Les fonctionnalités complémentaires.
- Un système de login pour les propriétaires et les admin.
- Un design pattern strategy pour les login owner/admin.
- Des requêtes xhr (alors que j'étais dispensé !)
- Un tableau de bord admin où on peut « tourner » à l'infinie.
- Du lazy load.
- ...

Lambert  
Thomas

## Autocritique

J'ai eu les yeux plus gros que le ventre. J'ai voulu faire trop de trucs mais comme je n'ai commencé qu'une semaine avant la remise à cause du cours de poo. En conséquence mes fichiers js ont beaucoup de répétition de code mais j'ai dû rusher sur la fin pour finir. Je n'ai aussi pas nommé mes controllers de la même manière partout comme dans le parent abstract controller.php. Je n'ai pas vraiment eu le temps de faire du css. Sinon en vrai je suis plutôt fier de moi.