



Diplôme Universitaire de Technologie (DUT)

Ingénierie Logicielle et Cybersécurité (ILCS)

VetCare360 – Application Web de Gestion d'une Clinique
Vétérinaire

Présenter Par :

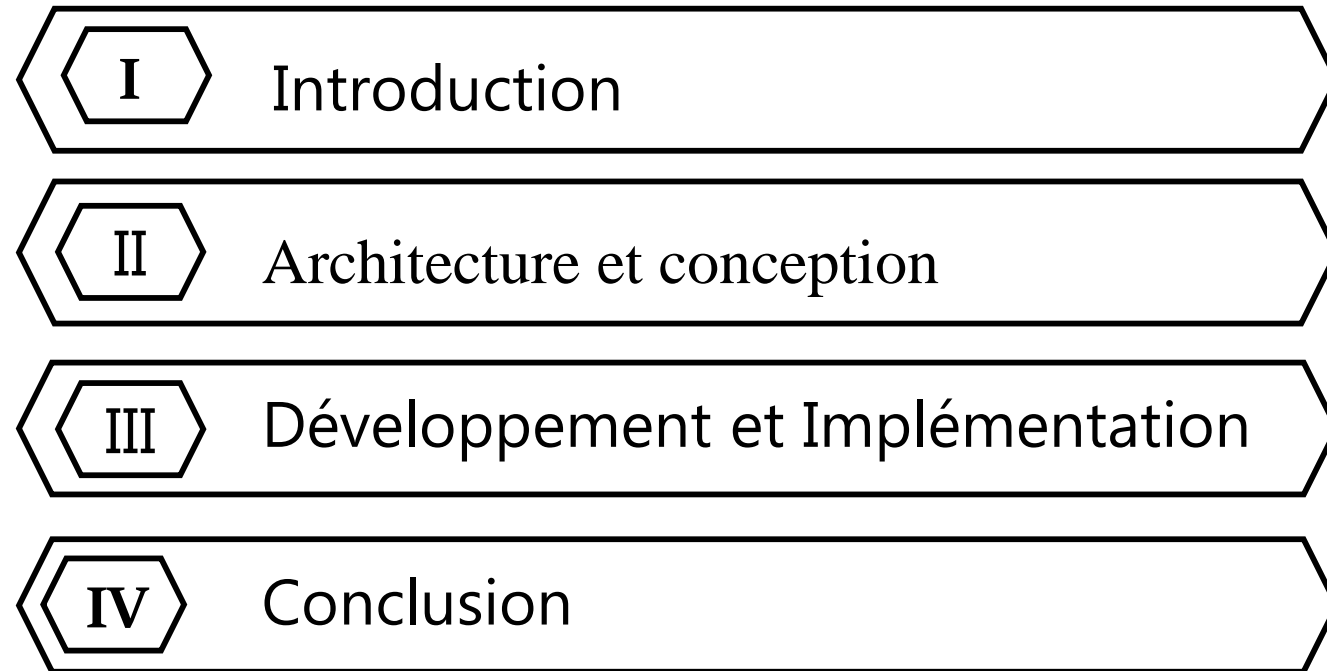
JAAI Malak & MHARI Ilham

Encadrer par:

Pr. ESBAI Redouane

Année universitaire : 2024-2025

Le Plan Du Présentation



Introduction

Avec l'évolution des outils informatiques, la gestion numérique des cliniques devient une nécessité incontournable. C'est dans ce cadre que nous avons développé **VetCare360**, une application desktop réalisée en **Java** avec **JavaFX**, destinée à la gestion complète d'une clinique vétérinaire. Elle intègre plusieurs fonctionnalités adaptées aux besoins du personnel médical et administratif, avec une interface graphique conviviale et intuitive. Le projet est né d'une volonté de moderniser les pratiques manuelles souvent longues, inefficaces et sujettes aux erreurs, en offrant une meilleure organisation, une traçabilité accrue des données et une prise de décision facilitée.

VetCare360 permet ainsi la gestion des propriétaires et de leurs animaux de compagnie, l'enregistrement et la consultation des visites médicales, la gestion du personnel vétérinaire avec leurs spécialités, tout en utilisant un système de **stockage temporaire en mémoire**, sans recourir à une base de données. Chaque entité du domaine (propriétaire, animal, visite, vétérinaire) est modélisée sous forme de classes **Java** manipulées via une interface ergonomique, ce qui fait de **VetCare360** une solution à la fois fonctionnelle et pédagogique, alignée sur les bonnes pratiques du développement logiciel.

Architecture et conception

Architecture MVC :

L'application utilise l'architecture MVC (Modèle-Vue-Contrôleur) afin de bien distinguer les rôles et faciliter la gestion du code :

☐ **Modèle (Model)**

Contient la logique métier et les données (Propriétaire, Animal, Visite, Vétérinaire).
Données gérées uniquement en mémoire avec des listes JavaFX (ArrayList, ObservableList).

☐ **Vue (View)**

Interface utilisateur conçue avec FXML pour une organisation claire.
Style graphique appliqué via un fichier CSS pour une interface moderne.

☐ **Contrôleur (Controller)**

Gère les interactions utilisateur (clics, formulaires).
Met à jour le modèle et la vue en fonction des actions.

Architecture et conception

Données en mémoire :

Dans cette version de VetCare 360, les données ne sont pas stockées dans une base de données externe, mais directement en mémoire. Ainsi, au démarrage de l'application, toutes les listes sont initialement vides. L'utilisateur peut ensuite ajouter, modifier ou supprimer des informations (propriétaires, animaux, visites) durant l'utilisation, mais ces données ne sont conservées que pendant la session en cours.

Architecture et conception

Conception orientée objet

L'application repose sur une conception orientée objet, où chaque entité métier est représentée par une classe Java adaptée. Les principales classes sont :

- **Propriétaire** (informations personnelles)
- **Animal** (détails et référence au propriétaire)
- **Visite** (date, motif et lien vers l'animal)
- **Vétérinaire** (données du praticien)

Chaque classe contient ses données et méthodes, et les relations entre elles sont gérées par des associations. Cette organisation en mémoire reflète efficacement le fonctionnement d'une clinique vétérinaire sans base de données.

Architecture et conception

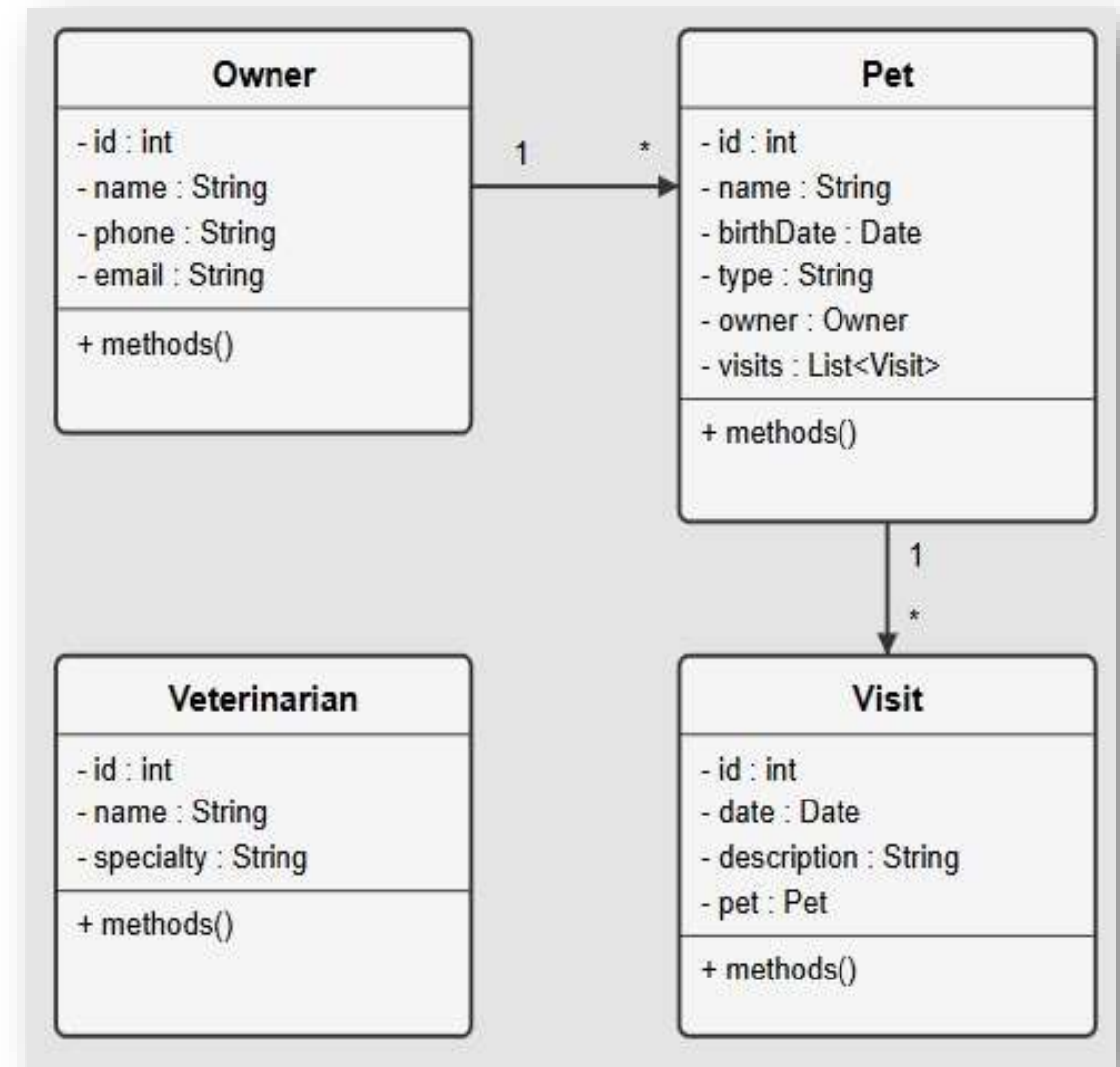
Diagramme de classes :

Les principales classes du modèle sont :

- **Owner** : id, name, phone, email
- **Pet** : id, name, birthDate, type, owner, visits
- **Visit** : id, date, description, pet
- **Veterinarian** : id, name, specialty

Relations entre les classes :

- Un **Owner** peut avoir plusieurs **Pets** (1---*)
- Un **Pet** peut avoir plusieurs **Visits** (1---*)



Environnement de développement (IDE) :

L'IDE **IntelliJ IDEA Community Edition** a été choisi pour sa richesse fonctionnelle, sa compatibilité native avec JavaFX, ainsi que sa capacité à organiser des projets multi-fichiers dans une architecture modulaire claire et maintenable.

Gestion de projet :

Le projet ne repose ni sur Maven ni sur Gradle, car il utilise exclusivement des bibliothèques standards fournies avec le JDK (JavaFX, collections Java, ...). La structure du projet est gérée de manière manuelle, sans système de build externe.

Développement et Implémentation

Technologies et outils :

Langages utilisés :

- **Java** est utilisé pour implémenter la logique de l'application, définir les entités métier et gérer les actions via les contrôleurs.
- **FXML** permet de concevoir les interfaces graphiques de manière déclarative, facilitant la séparation entre la logique et la présentation.
- **CSS** est employé pour styliser l'interface et personnaliser l'apparence visuelle des composants JavaFX.

Frameworks et bibliothèques :

- **JavaFX** : framework utilisé pour concevoir et gérer l'interface graphique de l'application.
- **Collections Java** (telles que ArrayList) : bibliothèques employées pour stocker temporairement les données en mémoire durant l'exécution du programme.

Présentation de l'application :

L'application propose plusieurs interfaces accessibles via un menu principal, chacune gérée par un contrôleur spécifique :

- ☐ Une **interface de gestion des propriétaires** (ajout, modification, suppression).
- ☐ Une **vue dédiée aux animaux**, associés à leurs propriétaires respectifs.
- ☐ Un **formulaire d'enregistrement des visites vétérinaires** pour chaque animal.
- ☐ Une **page d'affichage des vétérinaires**, avec leurs informations et spécialités.



Chaque interface est conçue en FXML, stylisée avec CSS, et reliée à un contrôleur Java qui assure la gestion des événements et la manipulation des données en mémoire.

Conclusion

La réalisation du projet VetCare360 a été une expérience enrichissante tant sur le plan technique qu'humain. En répondant à un besoin concret de gestion numérique d'une clinique vétérinaire, nous avons conçu une application moderne, complète et conforme aux standards actuels.

Ce projet nous a permis d'appliquer concrètement nos connaissances en programmation orientée objet, architecture MVC, conception d'interfaces graphiques, ainsi que des compétences en travail d'équipe, gestion de versions avec Git, et résolution de problèmes techniques.

Le choix de JavaFX nous a offert un cadre puissant pour développer une application desktop fonctionnelle et modulaire.

Au-delà d'un simple exercice, VetCare360 a renforcé notre compréhension du métier de développeur et posé les bases pour des améliorations futures.

Références bibliographiques

- [1] Oracle Corporation. (2024). *JavaFX Documentation*. Disponible sur : <https://openjfx.io/>
- [2] Spring Team. (2020). *Spring PetClinic Community Edition*. GitHub Repository. Disponible sur : <https://github.com/spring-projects/spring-petclinic>
- [3] Oracle Corporation. (2024). *Introduction to FXML – JavaFX Documentation*. Disponible sur : https://openjfx.io/javadoc/17/javafx.fxml/javafx/fxml/doc-files/introduction_to_fxml.html
- [4] Oracle Corporation. (2024). *Cascading Style Sheets Reference Guide – JavaFX CSS*. Disponible sur : <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html>

Merci pour votre attention !