



CY Tech – PréING1 – Année 2024-2025 - MEF1

Rapport du Projet

Application de Gestion du Chenil – ChenYI-Tech

11/05/2025 – Module: Informatique 2

Réalisé par :

- Qays Chaabaoui
- Firmin Godinou
- Ayman Madrassi



SOMMAIRE

1. Introduction

1.1 Description de l'équipe

1.2 Description du Sujet

2. Organisation du Projet et Flux de Travail

2.1 Organisation de l'équipe

2.2 Flux de travail

3. Problèmes Rencontrés et Solutions Apportées

4. Résultats Obtenus

4.1 Gestion des données des résidents

4.2 Fonctionnalités de l'application via une interface textuelle

4.3 Variantes implémentées

4.3.1. Inventaire du refuge (INV_AGE_ASC)

4.3.2. Gestion du quotidien - Nettoyage (DAY_CLEAN)

5. Conclusion

Introduction :

1.1 Description de l'équipe

Ce projet a été réalisé par notre équipe constitué de trois personnes: Ayman Madrassi, Firmin Godinou et Qays Chaabaoui. Du fait de notre complémentarité et nos compétences nous nous sommes investis afin concevoir un programme répondant aux exigences du cahier des charges. La répartition des tâches s'est faite de manière intuitive nous permettant de répondre au mieux aux différents problèmes rencontrés.

1.2 Description du sujet

L'application conçue pour la gestion de pensionnaires du chenil ChenYI-Tech et développée en langage C dans le cadre de ce projet intègre certaines fonctionnalités qui visent à réduire le travail administratif à exécuter sur les animaux hébergés par ChenYI-Tech. À cet égard, le programme à concevoir doit permettre l'adoption automatisée ou semi-automatisée de processus de gestion de l'information. Cette adoption se base sur une rationalisation assez profonde permettant de garder les informations des animaux bien rangées dans des fichiers sur l'ordinateur, afin savoir qui est là et qui a été adopté. Tout au long ce programme, il inclura une interface utilisateur dite « en console ». Cette interface est pensable par la logique d'un système d'exploitation Windows, qui permet de réaliser toutes les requêtes citées. Il s'agit d'ajouter requêtes tels que les mises à jour et les recherches multicritères d'animaux, leurs fiches descriptives, mais aussi leur départ à la suite d'adoption. De plus, le projet s'intéresse plus particulièrement à sponsoriser d'autres volets comme l'inventaire du refuge ainsi que la gestion et la programmation des ressources quotidiennes.

L'objectif principal est donc de fournir une application fonctionnelle et stable qui simplifie les tâches administratives, améliore le suivi des animaux et soutient la mission du chenil ChenYI-Tech

2. Organisation du Projet et Flux de Travail

2.1 Organisation de l'équipe

Afin de réaliser ce projet, nous avons mis en place une organisation basée sur de nombreux piliers :

L'ensemble du projet est constitué d'un dépôt Git centraliser répertorier et hébergés sur GitHub. Notre utilisation de Git n'était pas optionnelle, c'était obligatoire.

- Chaque ajout, modification ou suppression de code faisait l'objet d'un commit avec un message décrivant l'action effectuée. Cela nous permettait de garder une trace sur notre évolution ainsi que de nous permettre d'éviter de reproduire les mêmes erreurs.
- Git nous a facilité le travail. Chaque membre pouvait travailler sur des fonctionnalités distinctes dans des branches séparées, nous permettant d'éviter des confusions ou la suppression de code.
- Le dépôt centralisé a joué un rôle de sauvegarde automatique de notre projet. En cas de problème matériel sur l'ordinateur d'un membre, le code restait accessible et protégé. De plus, cela nous garantissait que tous les membres du groupe travaillaient toujours sur la version la plus à jour du projet.

De plus, afin de faciliter la communication, nous avons eu recours à des outils tels que Discord afin que l'on puisse échanger de manière très simple.

- L'objectif était de créer un environnement où chaque membre se sentait à l'aise pour poser des questions, partager ses idées ou signaler des difficultés.
- Les décisions importantes ainsi que les clarifications sur le cahier des charges y étaient partagées afin que chacun d'entre nous puissent suivre l'avancer globale du projet.

2.2 Flux de travail

Afin atteindre les objectifs définis dans le cahier des charges de manière structurée et progressive, nous avons adopté un flux de travail répétitif. Nous avons développé le programme progressivement en instaurant constamment des retours et en nous adaptant aux défis rencontrés. Notre travail a été divisé en plusieurs phases que nous avons répété pour chaque fonctionnalité du projet. Nous avons tout d'abord planifié, analysé et conceptualisé les exigences du programme ainsi que

les attendus spécifiques de chaque fonctionnalité. Puis, nous débutions le code mais pas de manière globale. Que ce soit Qays, Firmin ou Ayman nous devions chacun développé de spécifique et pas à pas en incrémentant petit à petit chaque partie du code. Nous réalisions différents tests au fur et à mesure du projet afin de savoir quelle partie du code pourrait posséder d'éventuel problème que ce soit au niveau des donnée ou encore entre chaque incrémentation. Enfin, l'étape qui a été pour nous la plus importante a été la relecture et l'optimisation constante du code car cela nous permettait de constamment faire le lien avec le cahier des charges et ainsi limiter au maximum le risques d'oublier n'importe quelle consigne.

3. Problèmes Rencontrés et Solutions Apportées

Au début du projet, on a rencontré quelques difficultés, surtout au niveau de l'organisation du code et de la compilation. Comme on a structuré le projet avec plusieurs fichiers .c, il fallait faire attention à bien inclure les bonnes fonctions dans le fichier includes.h. Si on oubliait une déclaration, ça ne compilait pas. On a donc appris à mieux gérer les dépendances entre fichiers.

On a aussi eu quelques soucis au niveau de la compilation à cause de petites erreurs de syntaxe ou des oublis dans le Makefile. À force de tester et corriger, on a pu stabiliser le projet. Même si on pensait utiliser malloc au début, finalement on s'en est passé en structurant les données autrement, ce qui a simplifié notre code et évité des problèmes de mémoire.

Enfin, au niveau du travail en équipe, on a mis un peu de temps à bien utiliser Git au début, notamment pour gérer les branches et les conflits. Mais on a vite trouvé une organisation qui fonctionnait bien, et ça nous a permis de collaborer plus efficacement. Discord a aussi beaucoup aidé pour la communication entre nous.

4. Résultats Obtenus

4.1 Gestion des données des résidents

On a réussi à faire en sorte que chaque animal du chenil ait sa propre fiche avec des infos comme son nom, son âge, son espèce ou s'il a été adopté. Toutes ces données sont bien enregistrées et on peut les consulter facilement depuis l'application. On peut aussi les rechercher selon différents critères, ce qui rend la gestion plus efficace.

4.2 Fonctionnalités de l'application via une interface textuelle

On a fait une interface en ligne de commande assez simple à utiliser. Le menu propose les options principales comme : ajouter un animal, mettre à jour ses infos, le faire adopter ou le rechercher, afficher l'inventaire par quartile d'âge ainsi que d'estimer la charge de nettoyage hebdomadaire selon l'espèce. L'interface est claire, et on peut faire tout ce dont on a besoin pour gérer le chenil sans avoir à passer par une interface graphique.

4.3 Variantes implémentées

4.3.1 Inventaire du refuge (INV_AGE_ASC)

On a ajouté une fonction pour voir la liste des animaux triés par âge croissant. Ça permet de repérer rapidement les plus jeunes ou les plus vieux animaux encore présents.

4.3.2 Gestion du quotidien - Nettoyage (DAY_CLEAN)

On a aussi ajouté une fonction qui simule le nettoyage du refuge. C'est une sorte de rappel des tâches quotidiennes. C'est simple mais ça donne un côté plus réaliste à l'application.

5. Conclusion

Ce projet nous a appris beaucoup de choses, pas seulement sur le langage C, mais aussi sur comment travailler à plusieurs sur un même projet. Au début c'était compliqué, surtout avec la mémoire et les erreurs de compilation, mais on s'est accrochés et on a réussi à faire un programme qui marche assez bien.

On est contents du résultat : l'appli fonctionne, elle répond aux besoins du chenil, et elle est facile à utiliser. On a aussi appris à utiliser Git à mieux organiser notre code, et à se répartir les tâches. C'est un bon point de départ pour d'autres projets plus complexes plus tard, et on sait qu'on pourrait encore améliorer notre application, par exemple en ajoutant une interface graphique ou en connectant le programme à une base de données.