



YUCROQ

NANTENAINA Tina

GODFRIN Alex

SUTARIK Agathe

GARRIGA Léandre

VIEMON-DESPLANQUE Joffrey

Table des matières

I. PRESENTATION GENERALE	4
A. Contexte	4
B. Description et objectifs du projet	4
C. Contenu de l'application	4
D. Technologies utilisées	5
Back office.....	5
Front office.....	5
HTML (Hypertext Markup Language).....	5
CSS (Cascading Style Sheets).....	5
Javascript.....	5
JSON (JavaScript Object Notation).....	6
Outils utilisés.....	6
II. MODELE DE DONNEES.....	7
III. STRUCTURE DE L'APPLICATION.....	8
A. Architecture de l'application	8
B. Racine du projet.....	8
C. Sources	9
D. Les ressources	10
E. Base de données.....	11
F. Rendu de l'application	11
Page d'authentification.....	11
Page Mes compagnons.....	12
Page des Croquettes	17
IV. À FAIRE.....	18
Problèmes non résolus	18
Fonctionnalités restant à développer	18
Indications des prolongements possibles pour le projet.....	18
V. AUTRES ASPECTS TECHNIQUES.....	19
Google Charts.....	19

Au cours de notre première année d'étude en cycle d'ingénieur à ISIS, nous devions réaliser un projet tuteuré sur les semestres 5 et 6. À la fin du cinquième semestre, en janvier, une bibliographie, disponible sur le dépôt GitHub de notre projet, a été finalisée. Le projet finalisé et opérationnel a été soutenu fin avril. Ce rapport accompagne cette soutenance finale.

Le projet devrait être réalisé en groupe. Notre groupe était constitué de cinq personnes issues de la formation ingénieur de l'école ISIS (informatique et systèmes d'information pour la santé), afin que chacune d'entre elles puisse apporter son savoir-faire dans le domaine qui lui avait été confié.

Le thème du projet a été proposé par Joffrey (membre du groupe) et a été validé par M. Rémi Bastide, enseignant-chercheur à ISIS et chercheur à l'IRIT (Institut de Recherche en Informatique de Toulouse).

Ce projet ayant pour finalité la conception et le développement d'une application web nous a permis de mettre en pratique nos connaissances et d'améliorer nos compétences.

Dans ce rapport, nous vous parlerons en détail du projet, de son objectif, de ses enjeux, et des difficultés que nous avons rencontrées.

I. Présentation générale

A. Contexte

Nos animaux domestiques sont souvent mal nourris, et peuvent développer des maladies liées à cette mauvaise alimentation et à la sédentarité (obésité, diabète, arthrose...). De plus, de nombreuses marques de croquettes ont fait leur apparition ces dernières années, prônant la qualité, le naturel, voire le made in France... Mais leurs informations sont parfois incomplètes ou erronées... Qu'en est-il vraiment ?

B. Description et objectifs du projet

Ce projet a pour but de proposer une façon simple de nourrir nos compagnons à quatre pattes à l'aide d'une application.

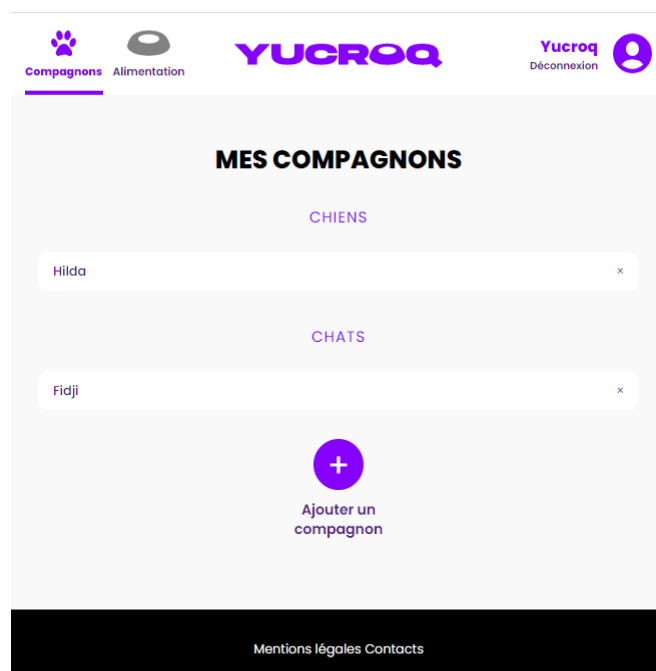
Celle-ci permettra de calculer la ration idéale de son chien ou son chat à partir de la composition de croquettes figurant dans une base de données complète, ainsi que des données fournies par le propriétaire sur son animal (âge, stade physiologique...).

Une autre fonction envisagée (non implémentée à l'heure actuelle) est de permettre de trouver, dans le cas de croquettes de mauvaise qualité, une alternative plus équilibrée et pas forcément plus coûteuse.

C. Contenu de l'application

L'application web se compose des parties suivantes :

- Compagnons
- Alimentation
- Moteur de recherche



D. Technologies utilisées

Back office

Pour cette partie, nous avons utilisé la technologie **Java Enterprise Edition J2EE**, avec le **framework Spring MVC** qui permet de s'affranchir de la gestion des servlets. Nous avons également effectué de la gestion directe des bases de données (couche DAO, requêtes SQL brutes) en utilisant un *framework* de persistance des objets : **Hibernate**. Nous avons aussi utilisé **Spring Boot** qui permet de s'affranchir des problèmes de dépendances et d'automatiser la création des fichiers .jar.

Pour gérer la sécurité de notre application web et les types d'utilisateurs, nous avons utilisé **Spring Security**, un autre *framework*.

Puis pour gérer la communication entre le *front* et le *back*, nous avons choisi le moteur de **template Thymeleaf** possédant une syntaxe plus souple. Celui-ci va permettre d'afficher à l'utilisateur des informations venant de la partie *back office*, comme des valeurs persistées en base de données par exemple.

AJAX est quant à lui utilisé pour communiquer des données de base de données vers l'application. Les différentes requêtes utilisées dans le projet sont principalement des méthodes « GET » et « POST » qui permettent, respectivement, de récupérer les informations et d'en envoyer dans la base de données

Front office

Pour réaliser les différentes interfaces, un ensemble de pages dédiées aux animaux et aux croquettes ont été réalisées en HTML5/CSS3 et contiennent des scripts JavaScript, intégrées directement dans les pages ou au sein de scripts externes.

HTML (Hypertext Markup Language)

C'est un langage conçu pour représenter les pages web. Ce langage de balisage permet de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images et des formulaires de saisie.

CSS (Cascading Style Sheets)

Le CSS est un langage informatique permettant d'ajouter des contraintes de mise en forme graphique dans des documents web, dont les normes sont établies par le *World Wide Web Consortium* (W3C).

Javascript

Abrégé JS, il s'agit d'un langage de programmation de scripts principalement utilisé dans les pages web interactives.

JSON (JavaScript Object Notation)

JSON est un format de données textuelles et génériques permettant de représenter de l'information structurée, comme le permet le langage XML par exemple.

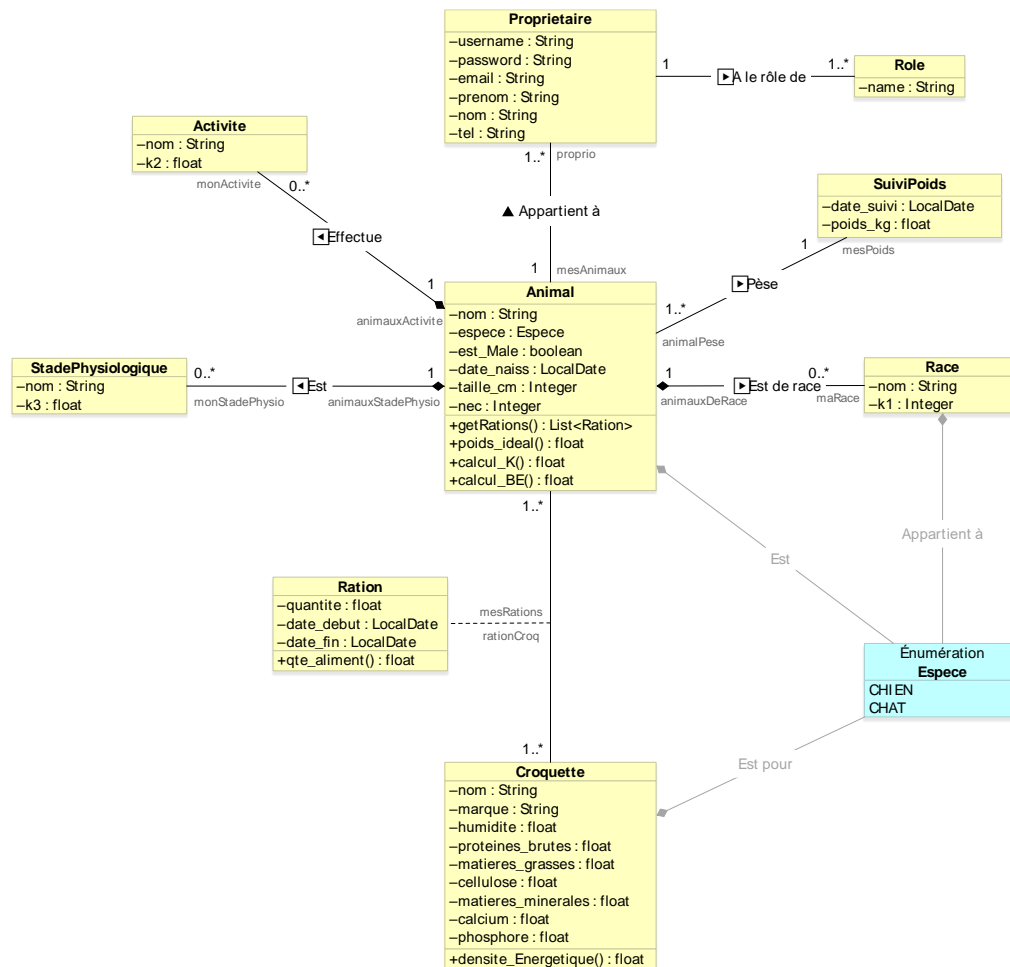
Outils utilisés

Les outils qui seront utilisés pour la réalisation du projet sont :

- **IDE Netbeans** : pour le développement avec J2EE, qui figure parmi les meilleurs environnements de développement intégré
- **Apache Maven** : outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Apache Maven est géré par l'organisation Apache Software Foundation
- Nous tenons à jour un projet **Git** sur le site GitHub, permettant le développement par de multiples personnes
- **Heroku** nous a permis de déployer notre application web sur une plateforme en ligne afin d'y permettre l'accès dans posséder le code source de l'application. De plus, le site nous a permis d'héberger nos données sur une base de données persistante et distante

II. Modèle de données

Le modèle de données est représenté sur la figure ci-dessous. La classe-entité de base est **Animal**.



Il est à noter que certaines entités disposent de méthodes permettant le calcul de valeurs caractéristiques déduites uniquement des données entrées dans l'entité. Par exemple, **densite_Energetique()** calcule comme son nom l'indique la densité énergétique d'un aliment à partir de sa composition.

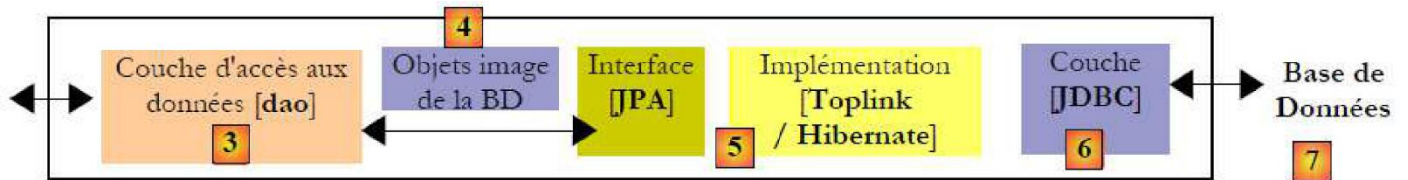
Ration constitue l'unique classe-association de notre modèle de données. En effet, il s'agit de l'unique association *Many to many*. Un animal reçoit successivement diverses rations, certaines consistant simplement à un changement de quantité des mêmes croquettes. Toutefois, il est important de permettre le suivi des rations successives pour voir leur effet sur la masse corporelle de l'animal.

Especie correspond à une énumération : CHIEN ou CHAT. Nous avons fait le choix de ne pas utiliser un booléen dans l'hypothèse de l'ajout d'autres espèces à l'avenir.

III. STRUCTURE DE L'APPLICATION

A. Architecture de l'application

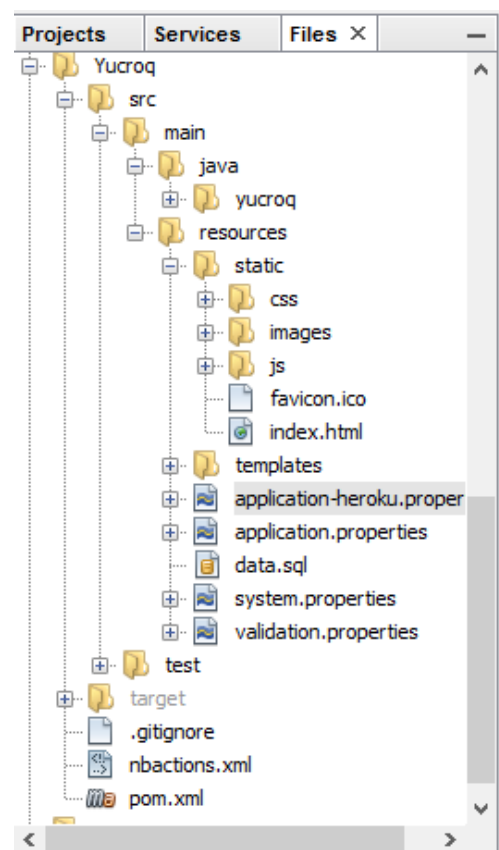
Notre application se base sur une architecture JAVA EE qui permet l'utilisation de JPA (Java Persistence API). Cette figure illustre les différentes couches de l'architecture de l'application.



En pratique, nous avons essayé de caractériser chaque couche par un package qui regroupe plusieurs éléments.

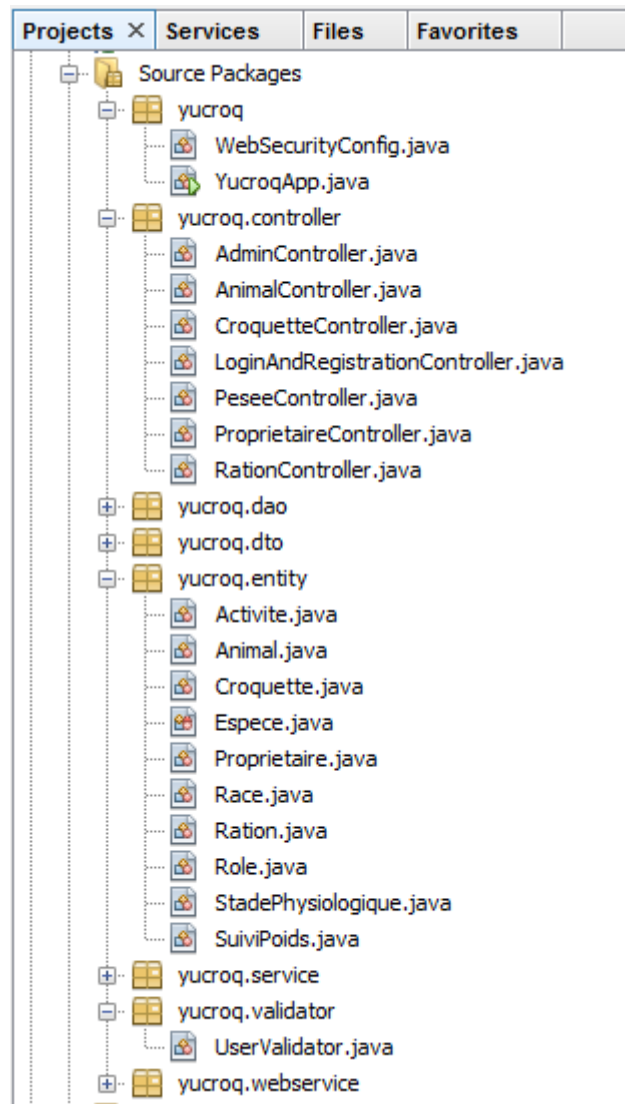
B. Racine du projet

- **Dossier Files** : c'est dans ce dossier que sont enregistrés les fichiers téléversés sur le serveur
- **application.properties** : fichier de configuration de l'application. On peut y trouver par exemple le nom de l'utilisateur et le mot de passe par défaut
- **pom.xml** : c'est le fichier utilisé par Maven dans lequel se trouvent les dépendances du projet



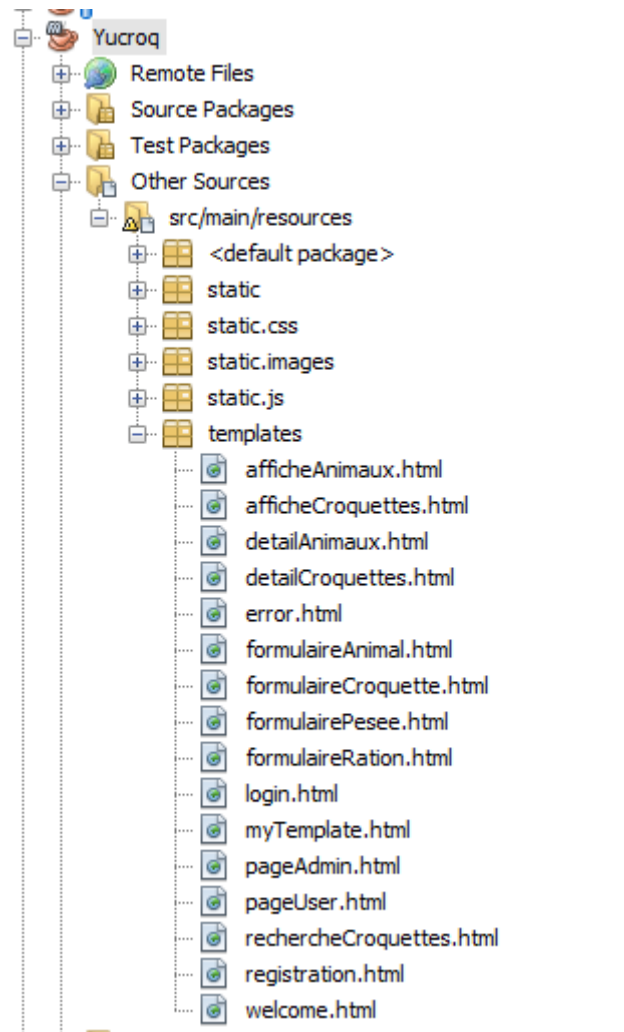
C. Sources

- **YucroqApp** : classe contenant une méthode d'initialisation de l'application. Contient aussi le programme principal (**main**) démarrant l'application Spring Boot
- **yucroq.controller** : dossier contenant les classes qui viennent faire correspondre toutes requêtes de l'utilisateur à des méthodes
- **yucroq.dao** : dossier contenant les **repository** de la classe correspondante
- **yucroq.entity** : dossier contenant toutes les entités
- **yucroq.service**, **yucroq.validator** et **yucroq.webservice** : contiennent la configuration utilisée par le framework Spring Security



D. Les ressources

- **static** : les fichiers qui ne sont pas concernés par du traitement opéré par un *framework* (Thymeleaf par exemple)
- **templates** : dossier utilisé par Thymeleaf pour générer les fichiers HTML à envoyer au client après le traitement du modèle (*template*) concerné
- **data.sql** : permet d'exécuter des instructions PSQL au démarrage de l'application



E. Base de données

The screenshot shows a database management interface. On the left, a tree view displays the database structure for 'jdbc:h2:mem:testdb', including tables like ACTIVITE, ANIMAL, CROQUETTE, PROPRIETAIRE, PROPRIETAIRE_ROLES, RACE, RATION, ROLE, STADE_PHYSIOLOGIQUE, SUIVI_POIDS, INFORMATION_SCHEMA, Sequences, and Users. The main area shows a SQL statement: 'select * from ration'. Below the statement, the results are displayed in a table with 6 columns: ID_RATION, DATE_DEBUT, DATE_FIN, QUANTITE, ID_ANIMAL, and ID_CROQ. The results show 4 rows of data.

ID_RATION	DATE_DEBUT	DATE_FIN	QUANTITE	ID_ANIMAL	ID_CROQ
1	2012-03-01	2012-03-03	200.0	1	1
2	2012-03-03	2012-03-05	180.0	3	2
3	2012-03-03	2020-04-04	400.0	1	1
4	2012-03-03	null	50.0	3	2

F. Rendu de l'application

Page d'authentification

The screenshot shows the authentication page of the application. At the top, there are icons for a paw print, a dog, the 'YUCROQ' logo, and a user profile icon. The main content area is titled 'Connexion'. It contains two input fields: 'NOM D'UTILISATEUR :' and 'MOT DE PASSE :'. Below these fields is a large blue button labeled 'Connexion'. At the bottom, there is a link labeled 'Créer un compte'.

Les utilisateurs de l'application sont représentés par l'entité **Propriétaire**, il est à noter que le mot **User** est réservé par PostgreSQL et ne peut être utilisé comme nom de table.

Les utilisateurs peuvent avoir un ou plusieurs rôles (gérer par l'entité **Rôle**). Nous avons prédéfini deux rôles : **ROLE_USER** et **ROLE_ADMIN**. Ces deux rôles sont créés automatiquement au démarrage de l'application.

Identifiant et mot de passe de l'administrateur sont définis dans le fichier **application.properties**. Un utilisateur ne pourra donc accéder à l'application que s'il s'identifie en indiquant son nom d'utilisateur et son mot de passe. Dans le cas de la première connexion, il doit créer un compte.

En cas d'erreur d'authentification, un message d'erreur va s'afficher comme suit :



The screenshot shows the YUCROQ login interface. At the top is the YUCROQ logo. Below it is the title "Connexion". There are two input fields: "NOM D'UTILISATEUR :" and "MOT DE PASSE :". The password field is masked with dots. Below the fields is an error message: "Nom d'utilisateur ou mot de passe incorrect." At the bottom, there is a blue button labeled "Connexion" and a link labeled "Créer un compte".

Page Mes compagnons

Si le nom d'utilisateur et le mot de passe sont corrects, ou un nouveau compte a été créé, l'affichage bascule sur la page des animaux. Elle est encore vide s'il s'agit de la première connexion du propriétaire. Il sera donc invité à ajouter un nouveau compagnon.



The screenshot shows the YUCROQ "Mes compagnons" page. At the top, there is a navigation bar with a paw print icon and the text "Compagnons", the YUCROQ logo, and a user profile icon with the text "tinou Déconnexion". Below the navigation bar is the title "MES COMPAGNONS". In the center, there is a blue circular button with a white plus sign and the text "Ajouter un compagnon".

Après l'ajout de ses compagnons, le propriétaire peut consulter les caractéristiques de son compagnon. Il peut aussi modifier certaines données, et peut supprimer un animal s'il le juge nécessaire.



MES COMPAGNONS

CHIENS

Milou



CHATS



Minou

+

Ajouter un
compagnon

Sur cette interface, sont introduites les coordonnées du nouveau compagnon.





Déconnexion

AJOUTER UN ANIMAL

NOM

Nom de l'animal

ESPECE

Chien

GENRE

Femelle

DATE DE NAISSANCE

mm/dd/yyyy

TAILLE

20 cm

NOTE D'ÉTAT CORPOREL

5/9 - Poids idéal

RACE

Berger Blanc Suisse

ACTIVITÉ

Peu actif

STADE PHYSIOLOGIQUE

Entier

Enregistrer

Réinitialiser

Retour au menu

L'utilisateur doit renseigner un nom à l'animal. Dans le cas contraire un message de rappel s'affichera :

The screenshot shows a web form titled "AJOUTER UN ANIMAL". It contains three input fields: "NOM", "ESPÈCE", and "GENRE". The "NOM" field is empty and has a validation error message "Please fill out this field." displayed above it. The "ESPÈCE" field is also empty. The "GENRE" field is a dropdown menu with "Femelle" selected. The "NOM" field has a placeholder text "Nom de l'animal".

AJOUTER UN ANIMAL

NOM

ESPÈCE

GENRE

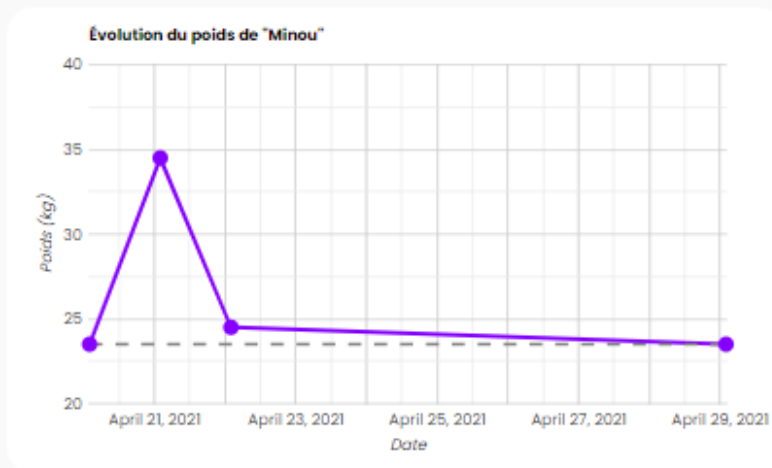
Please fill out this field.

Après l'enregistrement des données, la « carte d'identité » de l'animal est accessible. Cette interface nous renseigne sur le suivi de son poids, ainsi que les recommandations de ses rations

CARTE D'IDENTITÉ



NOM	Minou
ESPÈCE	Chien
GENRE	Femelle
DATE DE NAISSANCE	14/04/2021
TAILLE	23 cm
<hr/>	
NOTE D'ÉTAT CORPOREL	5/9
POIDS IDÉAL	24 kg
BESOIN ÉNERGÉTIQUE	1110 kcal/j

SUIVI DU POIDS




L'interface suivante montre les rations prises par notre compagnon. L'énergie totale fournie par la ration enregistrée est affichée en rouge si elle est trop basse ou trop haute (avec une marge de 5 %). Le bilan énergétique a été calculé dans la classe **Croquette.java**.

Une recommandation est également indiquée en fonction de ce bilan énergétique : le propriétaire est incité à adapter sa ration aux besoins énergétiques de l'animal.

 Compagnons
  Alimentation

YUCROQ

tinou 

Déconnexion

Ration actuelle

PURINA PROPLAN

MARQUE	Purina
NOM	ProPlan
COMMENCÉE LE	2021-04-20
DENSITÉ ÉNERGÉTIQUE	3551 kcal/kg
QUANTITÉ REÇUE	100 g/j
ÉNERGIE REÇUE	355 kcal/j

Modifier

BILAN ÉNERGÉTIQUE

ÉNERGIE TOTALE REÇUE 355 kcal/j



Recommandations

L'alimentation de Minou est trop pauvre en énergie, avec 355 kcal/j. Minou devrait manger 1110 kcal/j. Nous vous recommandons d'augmenter sa portion à 312 g/j de Purina ProPlan ou de basculer sur un aliment plus riche en énergie.


Anciennes rations

Aucune ancienne ration n'a été enregistrée.

S'il corrige effectivement la ration, le bilan énergétique est adapté en conséquence. Une ration équilibrée sera affichée en vert.

 Compagnons
  Alimentation

YUCROQ

tinou 

Déconnexion

ALIMENTATION

Ration actuelle

PURINA PROPLAN

MARQUE	Purina
NOM	ProPlan
COMMENCÉE LE	2021-04-27
DENSITÉ ÉNERGÉTIQUE	3551 kcal/kg
QUANTITÉ REÇUE	312 g/j
ÉNERGIE REÇUE	1108 kcal/j

Modifier

BILAN ÉNERGÉTIQUE

ÉNERGIE TOTALE REÇUE 1108 kcal/j

Recommandations

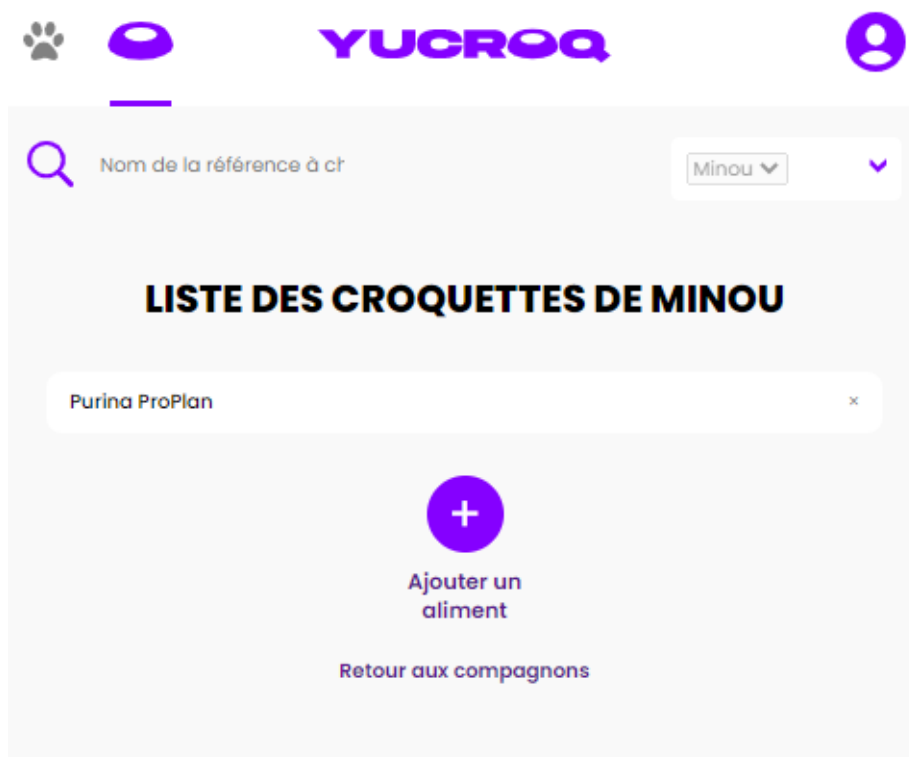
L'alimentation de Minou lui apporte la bonne quantité d'énergie. Minou doit en effet manger environ 1110 kcal/j. Nous vous recommandons de conserver cette ration quotidienne de Purina ProPlan en évitant les à-côtés caloriques.

Anciennes rations

PURINA PROPLAN 100 g/j du 20/04/2021 au 26/04/2021

Page des Croquettes

Sur cette page s'affichent les croquettes enregistrées pour un animal (rations actuelle et anciennes). Une barre de recherche est également disponible : elle permet de rechercher toutes les croquettes de la base de données correspondant à l'espèce de l'animal concerné.



The screenshot shows the 'Page des Croquettes' interface. At the top, there is a navigation bar with a paw print icon, a blue circular icon, the 'YUCROQ' logo, and a user profile icon. Below the navigation bar is a search section with a magnifying glass icon, a text input field containing 'Nom de la référence à cr', and a dropdown menu showing 'Minou' with a downward arrow. The main content area has a title 'LISTE DES CROQUETTES DE MINOU' in bold. Below the title is a search bar containing 'Purina ProPlan' with a close button (x). In the center, there is a blue circular button with a white plus sign, labeled 'Ajouter un aliment'. At the bottom, there is a link 'Retour aux compagnons'.

IV. À FAIRE

Problèmes non résolus

Hormis le problème de « compatibilité » entre les tests unitaire et la classe « YucroqApp » empêchant la réalisation des tests, nous n'avons jusqu'alors pas découvert de problème dans notre application.

Fonctionnalités restant à développer

D'un point de vue pratique, l'implémentation d'options permettant de **modifier les informations** de nos animaux et de nos croquettes via la partie web nous semble la priorité. **Diverses autres améliorations ergonomiques** sont également possibles comme des *pop-ups* modaux. L'introduction du scan de code-barres serait par ailleurs un progrès non négligeable.

Pour l'instant, nous avons deux rôles : USER et ADMIN, mais le rôle d'administrateur n'est pas utilisé, il faudrait pouvoir accéder à la base de données directement avec le rôle de l'administrateur. De plus on pourrait utiliser l'administrateur pour ajouter des croquettes.

Il nous reste bien sûr encore de nombreuses possibilités à déployer, notamment l'introduction du concept de **maladies** qui a pour but, en fonction de l'affection de l'animal, d'adapter les besoins énergétiques et *in fine* la ration de l'animal pour une améliorer la qualité de vie de l'animal.

La prise en compte des animaux en croissance est également envisagée, les chiots et chatons n'ayant pas les mêmes besoins que des animaux adultes.

L'ajout d'informations sur les **additifs** et **ingrédients** en rapport avec les croquettes est lui fortement envisagé, avec un indice de qualité des aliments et une comparaison de la composition avec les normes et les recommandations.

La possibilité a été laissée d'ajouter des espèces en prévision du futur, notamment les NAC (nouveaux animaux de compagnie) : rongeurs, furets...

Indications des prolongements possibles pour le projet

L'objectif final de cette application sera son utilisation par des particuliers, guidés par des professionnels qui pourraient affiner le profil d'un animal (note d'état corporel, coefficient K...) afin de donner des informations plus précises. Quelques applications sur le même thème existent déjà mais leur ergonomie dissuade fortement leur utilisation.

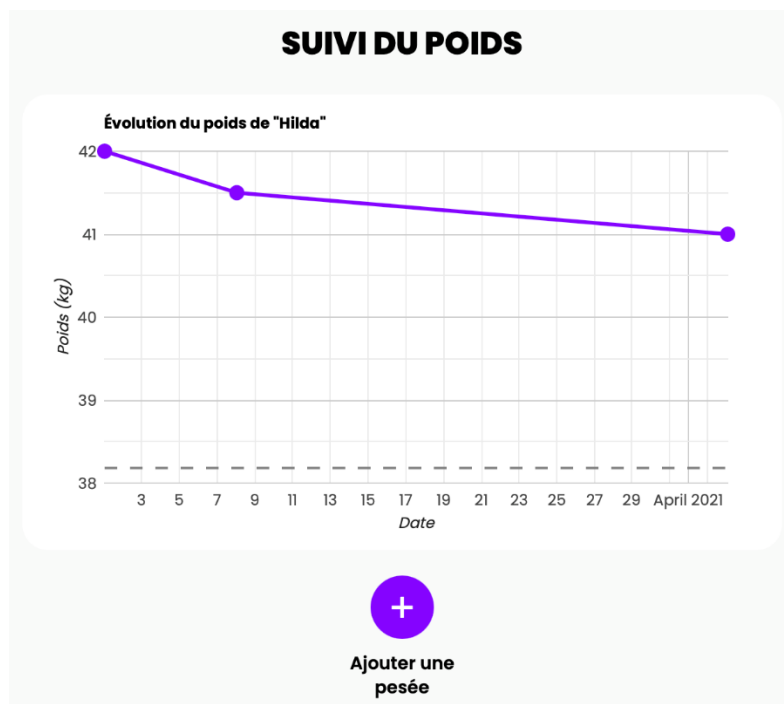
Le développement d'une application mobile est fortement envisagé dans le futur pour en améliorer l'accessibilité et l'utilisation.

V.AUTRES ASPECTS TECHNIQUES

Google Charts

Outil permettant l'affichage de graphique plus ou moins complexe basé sur un ensemble de données renseigné au préalable.

Utilisé ici pour l'affichage du suivi des poids d'un animal sous la forme de courbes, avec le poids de l'animal (en kg) en fonction de la date (au format « mm dd, yyyy »).



Le bouton « Ajouter une pesée » permet, après avoir rempli le formulaire de la nouvelle pesée de l'animal, de mettre à jour le graphique reliant le dernier point créé par le formulaire avec le nouveau.

Site de l'outil : <https://developers.google.com/chart>