Table des matières

[Introduction 3](#_Toc198385731)

[Préambule 3](#_Toc198385732)

[Exigences 3](#_Toc198385733)

[Remerciements 3](#_Toc198385734)

[Description de la problématique et du projet 4](#_Toc198385735)

[Méthodologie de travail 5](#_Toc198385736)

[Analyse du secteur 6](#_Toc198385737)

[Réglementation 6](#_Toc198385738)

[Etat du secteur dans l’Union Européenne. 6](#_Toc198385739)

[Etat du secteur en Belgique 6](#_Toc198385740)

[Etat du secteur en Wallonie 6](#_Toc198385741)

[Etat de l’art 7](#_Toc198385742)

[Introduction 7](#_Toc198385743)

[Présentation des solutions existantes 7](#_Toc198385744)

[Fonctionnalités proposées 9](#_Toc198385745)

[Tableaux comparatifs des solutions analysées 13](#_Toc198385746)

[Valeur ajoutée du projet 14](#_Toc198385747)

[Analyse Métier 15](#_Toc198385748)

[Description des intervenants 15](#_Toc198385749)

[Exigences Fonctionnelles 15](#_Toc198385750)

[*Fonctionnalités Communes* 15](#_Toc198385751)

[*Fonctionnalités Gérant* 15](#_Toc198385752)

[*Fonctionnalités Travailleur* 16](#_Toc198385753)

[*Fonctionnalités Application mobile* 16](#_Toc198385754)

[Exigences non-fonctionnelles 16](#_Toc198385755)

[Analyse fonctionnelle 17](#_Toc198385756)

[Sitemap de l’application 17](#_Toc198385757)

[Diagramme de cas d’utilisation 18](#_Toc198385758)

[Description détaillée des cas d’utilisation 19](#_Toc198385759)

[Wireframes de l’application mobile 51](#_Toc198385760)

[Description des entités 53](#_Toc198385761)

[Contraintes fonctionnelles 54](#_Toc198385762)

[Règles de structures 54](#_Toc198385763)

[Règles de validation 56](#_Toc198385764)

[Schéma relationnel de la solution 57](#_Toc198385765)

[Diagramme de classe 58](#_Toc198385766)

[Analyse Technique 59](#_Toc198385767)

[Choix technologiques 59](#_Toc198385768)

[Architecture applicative 59](#_Toc198385769)

[Base de données 59](#_Toc198385770)

[Back-end 59](#_Toc198385771)

[Front-end 61](#_Toc198385772)

[Bilan et perspectives 63](#_Toc198385773)

[Bilan du projet 63](#_Toc198385774)

[Perspectives 63](#_Toc198385775)

[Conclusion 64](#_Toc198385776)

[Bibliographie 65](#_Toc198385777)

# Introduction

## Préambule

Le sujet de ce projet m’a été inspiré par mon vécu familial. Durant mon adolescence, j’ai consacré une partie conséquente de mon temps libre dans l’exploitation agricole gérée par mes grands-parents. J’étais fasciné par de nombreux aspects du monde agricole : le contact avec les animaux, les impressionnantes machines agricoles ou encore le processus de production du lait. Cet intérêt pour le milieu agricole m’avait fait envisager une carrière professionnelle dans le secteur.

A l’époque, la digitalisation des processus de gestion n’était pas encore répandue, particulièrement pour les petites exploitations agricoles. J’ai ainsi pu directement constater la difficulté rencontrée par les agriculteurs dans la gestion économique et administrative de leur activité.

Bien que ma carrière professionnelle se soit finalement orientée vers d’autres horizons, le fait de pouvoir aujourd’hui développer une application informatique m’a rappelé mon intérêt pour le monde agricole et m’a fait prendre conscience que je pouvais y contribuer, si ce n’est indirectement.

Il s’est avéré naturel pour moi de mettre les compétences acquises au sein de l’EPHEC au service des petites exploitations familiales, qui représentent encore aujourd’hui une partie importante du monde agricole en Belgique.

Ce projet représente la conclusion de mon parcours académique. Il concrétise quatre années d’efforts, de découvertes et de progression des mes compétences. Ce travail final synthétise non seulement les connaissances pratiques acquises, mais également la méthodologie de travail, la rigueur et le professionnalisme développés progressivement au fil de mon cursus.

## Exigences

Le projet doit comporter une base de données avec des contraintes d’intégrité. Une partie back-end doit être en charge des requêtes sur la base de données, ainsi que de l’application de la logique métier sur ces données. Cette partie doit rendre ces données disponibles au travers d’une Web API. Une partie front-end aura la charge de la présentation des informations au travers d’une interface utilisateur, à l’aide des données fournies par la Web API.

Le Projet doit comporter un ensemble suffisant de fonctionnalités. Au minimum doivent être présentes :

* Un système d’authentification.
* Un système d’autorisation au moyen de rôles et de permissions.

Les technologies pour les différentes parties du projet sont laissées au libre choix de l’étudiant.

## Remerciements

Je tiens en premier lieu à remercier monsieur Mohamad Haji, rapporteur pour ce travail de fin d’étude. Ses conseils avisés et son aide précieuse, tant pour la partie développement logicielle que pour la rédaction du rapport, m’ont permis d’avancer avec confiance dans la réalisation de ce projet. Je souhaite également remercier l’ensemble du corps professoral et éducatif de l’EPHEC. Enfin, je veux aussi exprimer ma gratitude envers les élèves avec qui j’ai partagé ces 4 années et qui m’ont été d’un soutien indispensable tout au long de mon cursus.

# Description de la problématique et du projet

Le métier d’éleveur bovin en Belgique a fortement évolué au cours des dernières décennies. La concentration du secteur, les changements de la politique agricole commune et les nouvelles technologies ont bouleversé la manière de gérer une exploitation. Ces changements ont eu d’autant plus d’impact sur les petites exploitations familiales, qui constituent une partie conséquente du secteur en Belgique et en Europe. L’élevage d’un troupeau bovin demande aujourd’hui des outils digitaux de gestion permettant le suivi des animaux et de la production laitière, l’organisation du travail, et l’encodage des informations.

En outre, les exigences légales et administratives liées aux réglementations sanitaires sont toujours plus importantes et imposent aux exploitants de maintenir des informations complètes, claires et faciles d’accès sur leur troupeau.

Le projet concerne une application de gestion d’un troupeau bovin destinée à répondre efficacement, de manière accessible et à moindre cout aux besoins et nécessités décrites. L’application doit permettre la gestion des différents aspects relatifs au cœur de métier de la gestion du bétail. Elle s’adresse aux exploitations agricoles familiales de petite ou moyenne taille. Elle est prévue pour la gestion du bétail bovin élevé sur le territoire Belge, et pour les différentes finalités d’un élevage (production de lait, de viande, etc..). L’application offrira différentes fonctionnalités et se présentera sous la forme d’une application web d’une part, et d’une application mobile d’autre part.

Les fonctionnalités relatives à l’application seront les suivantes :

* Suivi individuel des têtes de bétail
  + Marquage,
  + Arbre généalogique
  + Maladies et soins
  + Vaccinations
  + Gestion des reproductions
  + Suivi des gestations
* Gestion d’un troupeau :
  + Planification et Suivi de l’alimentation

* Gestion des enclos, des stabulations, des prairies
  + Répartition du cheptel entre ces différents emplacements
  + Visualisation et marquage cartographique
* Production laitière
  + Encodage des traites
  + Suivi des rendements et des quotas laitiers
* Organisation du planning
  + Planification et assignation des tâches quotidiennes
  + Calendrier de planification

Les différentes fonctionnalités sont interconnectées. Une exploitation utilisant l’application peut enregistrer plusieurs utilisateurs avec différents rôles et permissions :

Un utilisateur « gérant » pourra ajouter, modifier et supprimer tout type d’informations. Il pourra également créer des tâches et les assigner aux travailleurs. Un utilisateur « travailleur » pourra être assigné à des tâches, et pourra encoder certaines informations relatives à la traite, des incidents etc...

# Méthodologie de travail

Pour la réalisation de ce projet, une approche agile a été mise en œuvre. La méthodologie Scrum a été utilisée afin de garantir une progression itérative et incrémentale du développement.

Dans un premier temps, le cadre et les objectifs du projet ont été définis. Les exigences fonctionnelles ont été établies sur base des axes suivants :

* Un état de l’art. Une analyse approfondie a été réalisée sur les solutions de gestion de bétail existantes actuellement sur le marché. Les fonctionnalités proposées ont été synthétisées, classifiées et filtrées afin de retenir celles se rapprochant au plus des objectifs de mon projet.
* Des entretiens avec un professionnel actif dans le domaine. Ces entretiens ont permis d’identifier la plus-value apportée par chaque type de fonctionnalités, d’affiner la sélection de celles-ci et de leur assigner un ordre de priorité.

Sur cette base, les cas d’utilisations ont été rédigés. Cette liste de cas d’utilisation a ensuite été organisée en fonction d’un niveau de priorité associé à chaque cas, puis divisés en plusieurs ‘sprints’ d’une durée de deux semaines.

A l’issue de chaque sprint, une évaluation de l’avancement du projet a été réalisée. En fonction des difficultés rencontrées et/ou de la nécessité de changements à apporter aux spécifications du projet, les priorités ont été réévaluées et le sprint suivant a été adapté.

L’analyse métier a été réalisée en suivant les principes de l’analyse UML (Unified Modeling Language). Les acteurs, les fonctionnalités attendues et les spécifications fonctionnelles du projet ont été modélisées au moyen des différents outils définis par l’UML. Une description haut-niveau des fonctionnalités a été illustrée au moyen d’un diagramme de cas d’utilisation et d’une description détaille de chaque cas. Les cas d’utilisation on en outre été accompagnés de mockups de l’interface utilisateur. Un diagramme de classe a été utilisé pour modéliser l’architecture logicielle et les interactions logiques des différents éléments composants l’application.

# Analyse du secteur

## Réglementation

La réglementation Européenne (Règlement (CE) n° 1242/2008 de la Commission) catégorise les types d’exploitations en fonction de la valeur monétaire de la production totale d’une exploitation. Sont définies cinq types pour la catégorisation d’une exploitation concernée par l’élevage bovin :

* Les exploitations laitières spécialisées
* Les exploitations bovines spécialisées à orientation élevage et viande
* Les exploitations bovines « combinées » => Elevage et viande
* Les exploitations de poly élevage (plusieurs types de cheptel, dont l’élevage bovin)
* Les exploitations mixtes grandes cultures-herbivores => Exploitations qui mixent culture et élevage.

## Etat du secteur dans l’Union Européenne.

Le secteur agricole est un secteur clé de l’économie Européenne. Il représente 1,3 pourcent du PIB de l’Union européenne, pour un total de 224 milliards d’euros en 2023. Il s’agit également d’un secteur stratégique pour la politique de l’UE, qui lui garantit un niveau conséquent d’indépendance et de sécurité alimentaire. Il est pour cette raison encadré par la PAC (La politique agricole commune).

Les 27 pays de l’Union européennes comptaient en 2020 plus de 9 millions d’exploitations agricoles. Parmi celles-ci, 1,8 millions sont catégorisées comme relevant de l’élevage bovin. Une partie importante du cheptel bovin en Europe est située au sein des pays du Benelux.

Sur les 13 millions de personnes travaillant dans une exploitation agricole au sein de l’UE, plus de 80 pourcents travaillent pour une exploitation familiale.

## Etat du secteur en Belgique

En Belgique, on compte environ 36000 exploitations agricoles, dont 80 pourcents sont dites « familiales ». En ce qui concerne l’élevage Bovin, le cheptel sur le territoire Belge est estimé en 2024 à environ 2,2 millions de têtes, dont 540 000 vaches laitières. Cela représente 3 pourcents de la population de l’élevage bovin dans l’union européenne.

## Etat du secteur en Wallonie

En Wallonie, la taille moyenne des exploitations agricoles a considérablement augmenté depuis 1980. Ce constat se caractérise par un phénomène de concentration. Depuis 1980, le nombre d’exploitations wallonnes a diminué considérablement, passant de 37000 à moins de 13000 actuellement.

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Evolution de la taille des exploitations agricoles, RTBF

# Etat de l’art

## Introduction

Il existe sur le marché une offre diversifiée de solutions logicielles de gestion pour les exploitations d’élevage de cheptel. On distingue deux grandes catégories parmi les solutions existantes.

1. D’une part, les solutions généralistes qui permettent la gestion d’une exploitation agricole mixte (élevage et culture)
2. D’autre part, les solutions spécialisées dans la gestion et le suivi d’une exploitation d’élevage de cheptel. Au sein de cette catégorie, on peut également noter plusieurs distinctions
   * Une différenciation par type de cheptel. Certaines solutions sont dédiées au suivi d’un type d’élevage unique (par exemple, un cheptel bovin). D’autres permettent la gestion de plusieurs types.
   * Pour les solutions concernées par l’élevage bovin, une différenciation par type de production : production laitière, production viandeuse, ou production mixte.

## Présentation des solutions existantes

Nous présentons ici une sélection de solutions logicielles existantes représentative des différentes formules, formes et fonctionnalités offertes actuellement sur le marché.

* ***FarmBrite***

[*https://www.farmbrite.com/*](https://www.farmbrite.com/)

*Solution tout-en-un de gestion de fermes*

* + Catégorie : Généraliste, tout-en-un
  + Type : Culture et élevage et élevage (bovin, ovin, porcin etc…)
  + Segment de marché : Petite/Moyenne exploitation
  + Zone géographique : Monde entier
  + Format : Application web et mobile
  + Jusqu’à 5000 tètes de bétail
  + Prix : de 30$ à 110 $ par mois
  + Clients actifs : 4000 +
* ***NavFarm***

[*https://www.navfarm.com/*](https://www.navfarm.com/)

*ERP de gestion de ferme universel*

* + Catégorie : Généraliste, tout-en-un
  + Type : Culture et élevage et élevage (bovin, ovin, porcin etc…)
  + Segment de marché : Petite/Moyenne exploitation
  + Zone géographique : Amérique du nord, Amérique du sude, Europe, Asie
  + Format : Application mobile
  + Prix : 20$ pour une petite exploitation individuelle, prix à la demande pour une offre plus complète.
  + Clients actifs : -
* ***Agrovision Cow***

[*https://agrovision.com/be-fr/*](https://agrovision.com/be-fr/)

* + Catégorie : Spécialisée, modulaire
  + Type : Elevage bovin/Production laitière
  + Segment de marché : Petite/Moyenne exploitation, Consultant en production laitière
  + Zone géographique : Europe
  + Format : Application web et mobile
  + Prix : -
  + Clients actifs : 19000 +
* ***HerdWarch***

[*https://herdwatch.com/*](https://herdwatch.com/)

* + Catégorie : Spécialisée, modulaire
  + Type : Elevage bovin et ovin/Production laitière
  + Segment de marché : Petite exploitation
  + Zone géographique : USA, Royaume-Uni, Irlande
  + Format : Application mobile
  + Prix : De 200€ à 400€ par an (fonction de la taille du cheptel)
  + Clients actifs : -
* ***CattleMax***

<https://www.cattlemax.com/>

*Logiciel de gestion des données spécialisée dans les ranchs.*

* Catégorie : Spécialisée
* Type : Elevage bovin
* Segment de marché : Moyenne/Grande exploitation
* Zone géographique : Amérique du nord
* Format : Application Destkop et mobile
* Prix : -
* Clients actifs 80000 +

## Fonctionnalités proposées

Parmi les solutions logicielles analysées dans cette section, un large panel de fonctionnalités est proposé. Le type et la nature de celles-ci pour chaque logiciel va dépendre de divers facteurs tels que le segment de marché ciblé ou la région géographique concernée. Selon les offres, les fonctionnalités font partie d’un bloc commun ou se présentent de manière modulaire. En outre, elles sont souvent interconnectées ou interdépendantes.

Cette section présente une liste non exhaustive de ces fonctionnalités, illustrées d’exemples visuels.

#### Gestion de troupeau

Cette fonctionnalité, commune à la majorité des applications analysées, regroupe la collecte et l’affichage d’informations collectives et individuelles sur les animaux du troupeau d’une exploitation. La gestion collective se présente généralement sous forme de tableaux, tandis que des fiches individuelles par animal permettent la gestion individuelle. En plus des caractéristiques liées à un animal, une fiche individuelle affiche des éléments tels que l’historique des actions liées à cet animal, sa généalogie, son statut au regard de la production laitière ou des informations commerciales.

Une image contenant texte, capture d’écran, nombre, logiciel

Le contenu généré par l’IA peut être incorrect.

Figure : Gestion du troupeau par FarmBrite

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Historique des actions pour un animal par CattleMax

#### Suivi de la production laitière

Le suivi de la production laitière comprend le suivi des traites et l’enregistrement des volumes laitiers. Cet enregistrement peut se faire via un encodage manuel ou de manière automatique via une connexion à l’API du logiciel d’une salle de traite. Cette fonctionnalité va généralement de pair avec des possibilités avancées de reporting statistique sur les volumes de production par vache et pour le troupeau dans son ensemble.

#### Informations d’achats et de ventes

Ce type de fonctionnalités regroupe l’enregistrement des données relatives aux achats et ventes de bétail, aux acheteurs et aux prix.

#### Suivi sanitaire

Le suivi sanitaire regroupe l’enregistrement de tous les types de données et évènements relatifs à la santé de chaque animal du troupeau : les maladies, les blessures, les traitements médicamenteux, les vaccins ainsi que les interventions vétérinaires.

#### Gestion des reproductions

La gestion des reproductions comprend le suivi des périodes de chaleur, l’enregistrement des reproductions, le calendrier des vêlages à venir.

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Gestion des vêlages par AgrovisionCow

#### Gestion de pâturages

Les fonctions relatives aux pâturages ont pour but de gérer la répartition du troupeau à travers les différents pâturages de l’exploitation, la planification des mouvements ou le contrôle de la qualité et de la longueur de l’herbe. Ces fonctions sont généralement couplées avec un mécanisme de représentation visuelle sur une carte (type Google map).

Une image contenant texte, capture d’écran, carte

Le contenu généré par l’IA peut être incorrect.

Figure : Gestion des pâturages avec FarmBrite

#### Gestion de l’alimentation

Les fonctionnalités liées à la gestion de l’alimentation combinent le suivi des stocks de fourrages et la planification des rations alimentaires en fonction de facteurs environnementaux ou des résultats de la production laitière.

#### Gestion d’inventaire

Il s’agit de la gestion des stocks de matériel et de consommables liés à l’activité d’élevage.

#### Organisation du travail

Cette catégorie de fonctionnalités a pour objet l’organisation du travail d’une exploitation, à savoir la planification, l’attribution et le suivi de tâches aux utilisateurs enregistrés de l’application. Cela va de pair avec la gestion des utilisateurs. On peut également mentionner les fonctions de calendrier des tâches et de gestion des horaires, de todo-list ou de tableau kanban, d’organisation en équipes et de time-tracking.

Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Figure : Gestion des tâches avec FarmBrite

#### Intégration logicielle

Les solutions analysées offrent différents niveaux d’intégration avec des outils logiciels et matériels externes.

* Avec les logiciels connectés des salles de traites pour l’enregistrement des volumes laitier.
* Avec des distributeurs automatiques de nourritures connectés à la gestion des stocks de nourriture et à la planification de l’alimentation pour chaque animal.
* Avec des capteurs matériels attachés directement aux animaux du troupeau.

Une image contenant oiseau, noir, bec, orange

Le contenu généré par l’IA peut être incorrect.Par exemple, des capteurs d’oreille ou des capteurs par collier enregistrent plusieurs types de données et prévoient des interfaces applicatives pour la récupération aisée de ces données par un logiciel de gestion.

## Tableaux comparatifs des solutions analysées

#### Caractéristiques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Solution*** | Catégorie | Type | Segment de marché | Zone Géographique | Format |
| *FarmBrite* | Généraliste | Culture/Elevage bovin, ovin | Petite/Moyenne exploitation | Monde | Application web et mobile |
| *NavFarm* | Généraliste | Culture/Elevage bovin, ovin, | Petite/Moyenne exploitation | Monde | Application mobile |
| *Agrovision Cows* | Spécialisée | Elevage Bovin, Laitier | Petite/Moyenne exploitation | Europe | Application web / mobile |
| *HerdWatch* | Spécialisée | Elevage Bovin, Ovin, Laitier | Petite exploitation | USA, UK, Irlande | Application mobile |
| *CattleFarm* | Spécialisée | Elevage Bovin | Moyenne/Grande exploitation | Amérique du nord | Application destkop/mobile |

#### Fonctionnalités

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Solution*** | *FarmBrite* | *NavFarm* | *Agrovision Cow* | *HerdWatch* | *CattleFarm* |
| ***Fonctionnalités*** |
| Gestion de troupeau | X | X | X | X | X |
| Suivi de production laitière | X |  | X | X |  |
| Informations d’achats et de ventes | X | X |  |  | X |
| Suivi sanitaire | X |  |  | X |  |
| Gestion des reproductions | X |  |  | X |  |
| Gestion de pâturages | X |  |  | X |  |
| Gestion de l’alimentation | X |  |  |  |  |
| Gestion d’inventaire | X | X |  |  |  |
| Organisation du travail | X |  |  |  |  |
| Intégration logicielle | X |  | X | X | X |

## Valeur ajoutée du projet

L’analyse de l’évolution du secteur agricole en Europe et en Belgique montre que les petites et moyennes exploitations familiales constituent encore une grande partie de l’ensemble du secteur. Cependant, une logique croissante de concentration accentue la pression concurrentielle sur les exploitations de taille plus réduite. Il est de ce fait indispensable pour les gérants de celles-ci de se doter des outils digitaux leur permettant d’optimiser l’organisation du travail et la rentabilité.

Par rapport aux logiciels existants sur le marché, Le projet Cattle Manager se positionne comme une solution :

* Orientée pour les petites exploitations.
* Spécialisée sur la gestion de troupeau et le suivi de la production laitière.

En comparaison avec les fonctionnalités proposées par les solutions existantes, Cattle Manager offre les avantages suivants :

* Une grande facilité d’utilisation, grâce à une interface simplifiée et le regroupement d’un maximum d’informations sur un nombre restreint d’écrans.
* Une extensibilité future des fonctionnalités. Grâce à une structure de données claire et cohérente et une architecture logicielle découplée, l’extension ou l’ajout de nouvelles fonctionnalités est relativement aisée.
* Gratuité. Cattle Manager est gratuit d’utilisation.

# Analyse Métier

## Description des intervenants

**Les utilisateurs « gérants »** : Il s’agit du ou des responsables de l’exploitation agricole, ou de toute personne responsable de la gestion du bétail. Un gérant est en mesure d’ajouter, modifier ou supprimer les informations relatives aux animaux, aux groupes, aux enclos, aux stocks de nourriture. Il peut également encoder tout type d’évènement (maladie, soin, gestation, traite, …). Un gérant peut créer des tâches et les assigner aux travailleurs de son choix.

**Les utilisateurs « travailleurs »** : Il s’agit de toute personne en charge d’effectuer les tâches manuelles relatives à l’entretien du bétail et de l’exploitation. Un travailleur peut être assigné par un gérant à des tâches et pourra afficher les tâches lui étant attribuées pour une journée sur l’application. Il pourra ajouter ou modifier certaines informations et évènements.

## Exigences Fonctionnelles

Les exigences fonctionnelles de l’application web sont présentées ci-dessous. Les fonctionnalités communes concernent touts les utilisateurs, quel que soit leur rôle (« travailleur » ou « gérant »). Les autres fonctionnalités sont spécifiques à un rôle particulier (travailleur ou gérant).

Il est à noter que les rôles attribués ne sont pas mutuellement exclusifs et qu’un utilisateur peut être gérant et travailleur. Dans ce cas, il aura accès à toutes les fonctionnalités.

Les fonctionnalités de l’application mobile constituent un sous-ensemble des fonctionnalités de l’application web.

### *Fonctionnalités Communes*

|  |  |
| --- | --- |
| *Code* | *Description* |
| CM-C001 | S’authentifier |
| CM-C002 | Visualiser la liste des animaux |
| CM-C003 | Visualiser les détails d’un animal |
| CM-C004 | Visualiser la liste des groupes |
| CM-C005 | Visualiser la liste des enclos |
| CM-C006 | Visualiser la liste des évènements sur le troupeau |

### *Fonctionnalités Gérant*

|  |  |
| --- | --- |
| *Code* | *Description* |
| CM-G001 | Visualiser la liste des tâches |
| CM-G002 | Créer une tâche |
| CM-G003 | Modifier une tâche |
| CM-G004 | Assigner des travailleurs à une tâche |
| CM-G005 | Visualiser la liste des travailleurs |
| CM-G006 | Créer un utilisateur travailleur |
| CM-G007 | Créer un animal |
| CM-G008 | Modifier un animal |
| CM-G009 | Créer un groupe |
| CM-G010 | Modifier un groupe |
| CM-G011 | Créer un enclos |
| CM-G012 | Modifier un enclos |
| CM-G013 | Visualiser les statistiques du troupeau |

### *Fonctionnalités Travailleur*

|  |  |
| --- | --- |
| *Code* | *Description* |
| CM-T001 | Visualiser la liste des tâches assignées |
| CM-T002 | Mettre à jour le statut d’une tâche |
| CM-T003 | Transférer des animaux entre groupes |
| CM-T004 | Transférer des animaux entre enclos |
| CM-T005 | Encoder une traite |
| CM-T006 | Encoder une alerte pour un animal |
| CM-T007 | Encoder une alerte pour un enclos |
| CM-T008 | Encoder une gestation |
| CM-T009 | Encoder une intervention |
| CM-T010 | Encoder une affection |
| CM-T011 | Encoder un traitement |
| CM-T012 | Encoder une vaccination |

### *Fonctionnalités Application mobile*

|  |  |
| --- | --- |
| *Code* | *Description* |
| CM-C001 | S’authentifier |
| CM-C002 | Visualiser la liste des animaux |
| CM-C003 | Visualiser les détails d’un animal |
| CM-T001 | Visualiser les tâches assignées |
| CM-T002 | Mettre à jour le statut d’une tâche |
| CM-T005 | Encoder une traite |

## Exigences non-fonctionnelles

|  |  |
| --- | --- |
| *Code* | *Description* |
| CM-NF001 | L’application doit être sécurisée au moyen d’un système d’authentification |
| CM-NF002 | L’application doit présenter une cohérence visuelle (couleurs, polices, thème) |
| CM-NF003 | Les fonctionnalités doivent être facilement accessibles |

# Analyse fonctionnelle

## Sitemap de l’application

Une image contenant texte, capture d’écran, diagramme, Plan

Le contenu généré par l’IA peut être incorrect.

Figure : Sitemap de l'application web

## Diagramme de cas d’utilisation

Une image contenant texte, noir et blanc, document

Le contenu généré par l’IA peut être incorrect.

Figure : Diagramme de cas d’utilisation

## Description détaillée des cas d’utilisation

Ci-dessous sont décrits les cas d’utilisation associés aux exigences fonctionnelles, accompagnés de wireframes. Pour les fonctionnalités qui concernent également l’application mobile, Les wireframes sont présentées groupées.

#### CM-C001 S’authentifier

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de s’authentifier sur le portail web. |
| ***Acteurs*** | Gérant, Travailleur |
|  | |
| ***Préconditions*** | * L’utilisateur n’est pas encore authentifié sur le portail. |
| ***Postconditions*** | * L’utilisateur est authentifié sur le portail. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur arrive sur la page d’accueil. 2. Le système vérifie si l’utilisateur possède déjà un token. 3. S’il ne possède pas de token ou que le token présent est échu, l’utilisateur est redirigé vers la page de login administrée par Auth0. 4. L’utilisateur remplit ses informations d’authentification (adresse email et mot de passe) dans les champs du formulaire prévus à cet effet. 5. L’utilisateur sélectionne le bouton de confirmation. 6. L’utilisateur reçoit un token valide de la part d’Auth0. 7. L’utilisateur est redirigé vers le dashboard. |
| ***Scénario alternatif*** | Ce scénario alternatif commence au point 3 du scénario nominal.  3a. L’utilisateur possède déjà un token valide. L’utilisateur est redirigé automatiquement vers son dashboard en fonction de son rôle. |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 6 du scénario nominal.  6a. Les informations d’authentification sont incorrectes. Un message d’erreur apparaît et l’utilisateur est invité à réessayer de s’authentifier. |

Une image contenant texte, capture d’écran, logiciel, affichage

Le contenu généré par l’IA peut être incorrect.

Figure : Page de login

#### CM-C002 Visualiser la liste des animaux

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des animaux. |
| ***Acteurs*** | Gérant, Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur ou gérant. * Au moins un animal existe dans la base de données. |
| ***Postconditions*** | * La liste des animaux est affichée sous forme de liste ou de grille |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Bétail » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des animaux. Les animaux sont affichés sous forme de liste ou de grille. 3. L’utilisateur peut filtrer la liste des animaux en fonction d’un groupe, d’un enclos, ou d’une recherche textuelle sur les différents champs. La liste des animaux affichées est mise à jour en fonction du filtrage effectué. |
| ***Scénario alternatif*** | Ce scénario commence au point 2 du scénario nominal.  2a. L’utilisateur est redirigé vers la page de la liste des animaux. Un message indique qu’aucun animal n’est actuellement enregistré en base de données. |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, nombre, Parallèle

Le contenu généré par l’IA peut être incorrect.

Figure : Page Liste du bétail

#### CM-C003 Visualiser les détails d’un animal

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la fiche de détails d’un animal du troupeau. |
| ***Acteurs*** | Gérant, Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur ou gérant. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * La page de détail d’un animal est affichée pour l’animal sélectionné. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne dans la liste des animaux l’option de visualisation des détails pour l’animal souhaité. 2. L’utilisateur est redirigé vers la page de détails de l’animal sélectionné. 3. Sont affichées directement les caractéristiques, la photo et la position sur la carte de l’animal. Les affections, traitements, interventions et gestations sont également affichés sous forme de colonnes distinctes. 4. L’utilisateur peut visualiser des informations supplémentaires relatives à l’animal sous forme de popup en sélectionnant les options prévues à cet effet.  * Statistiques de production laitière * Historique * Arbre généalogique |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, diagramme, logiciel

Le contenu généré par l’IA peut être incorrect.

Figure : Page Détails d'un animal

#### CM-C004 Visualiser la liste des groupes

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des groupes. |
| ***Acteurs*** | Gérant, Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur ou gérant. |
| ***Postconditions*** | * La liste des groupes est affichée. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Groupes » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des groupes. Les groupes sont affichés sous forme de grille. |
| ***Scénario alternatif*** | Ce scénario commence au point 2 du scénario nominal.  2a. L’utilisateur est redirigé vers la page de la liste des groupes. Un message indique qu’aucun groupe n’est actuellement enregistré en base de données. |
| ***Scénario d’erreur*** | / |

Une image contenant capture d’écran, diagramme, ligne, texte

Le contenu généré par l’IA peut être incorrect.

Figure : Page des groupes

#### CM-C005 Visualiser la liste des enclos

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des enclos. |
| ***Acteurs*** | Gérant, Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur ou gérant. |
| ***Postconditions*** | * La liste des enclos est affichée. |
|  | |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Enclos » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des enclos. Les enclos sont affichés sous forme de grille. |
| ***Scénario alternatif*** | Ce scénario commence au point 2 du scénario nominal.  2a. L’utilisateur est redirigé vers la page de la liste des enclos. Un message indique qu’aucun enclos n’est actuellement enregistré en base de données. |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Figure : Page des enclos

#### CM-C006 Visualiser la liste des activités pour le troupeau

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des différentes activités pour l’ensemble des animaux du troupeau (Interventions, gestations, affections, traitements, vaccinations). |
| ***Acteurs*** | Travailleur, Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur ou gérant |
| ***Postconditions*** | * La liste des activités pour l’ensemble du troupeau est affichée, sous forme de colonnes. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Activités » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des activités, les activités pour l’ensemble des animaux du troupeau sont affichées sous forme de colonnes pour chaque type d’activités. 3. L’utilisateur peut filtrer les activités selon :  * La date * Les enclos * Les groupes |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, nombre, affichage

Le contenu généré par l’IA peut être incorrect.

Figure : Page des activités

#### CM-G001 Visualiser la liste des tâches

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des tâches. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. |
| ***Postconditions*** | * La liste des tâches est affichée. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Tâches » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des tâches. Les tâches du jour sont affichées sous forme de liste. 3. L’utilisateur peut filtrer les taches selon :  * La date * Le statut * L’enclos concerné * Le travailleur assigné |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, nombre, affichage

Le contenu généré par l’IA peut être incorrect.

Figure : Page de la liste des tâches

#### CM-G002 Créer une tâche

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’ajouter une tâche à la liste des tâches |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est connecté sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des tâches. |
| ***Postconditions*** | * Une nouvelle tâche a été enregistrée dans la base de données. Cette tâche est en état « à réaliser ». |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option d’ajout d’une tâche. 2. Un pop-up apparait contenant un formulaire d’ajout de tâche. 3. L’utilisateur complète les champs du formulaire avec les informations nécessaires pour la création d’une nouvelle tâche et enregistre la nouvelle tâche en cliquant sur le bouton de confirmation. 4. Le pop-up contenant le formulaire disparait et un message de confirmation apparait, mentionnant que le la tâche a été crée avec succès. 5. La liste des tâches est mise à jour avec la nouvelle tâche créée. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario commence au point 4 du scénario nominal.  4a. Un message d’erreur apparaît, mentionnant qu’une erreur s’est produite lors de l’enregistrement de la nouvelle tâche.  Ce scénario commence au point 5 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs incorrects sont mis en évidence. |

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Figure : Création d'une tâche

#### CM-G003 Modifier une tâche

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de modifier une tâche. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est connecté sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des tâches. * Il existe au moins une tâche en état « à réaliser » dans la base de données ». |
| ***Postconditions*** | * La tâche a été modifiée en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option de modification d’une tâche pour la tâche de son choix dans la liste. 2. Un pop-up apparait contenant un formulaire pré complété avec les informations de la tâche choisie. 3. L’utilisateur modifie les champs souhaités et enregistre les modifications en cliquant sur le bouton de confirmation. 4. Le pop-up contenant le formulaire disparait et un message de confirmation apparait, mentionnant que le la tâche a été modifiée avec succès. 5. La liste des tâches est mise à jour avec les informations mises à jour pour la tâche modifiée. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario commence au point 4 du scénario nominal.  4a. Un message d’erreur apparaît, mentionnant qu’une erreur s’est produite lors de l’enregistrement des modifications.  Ce scénario commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs incorrects sont mis en évidence. |
|  | |
| ***Contraintes*** | Seules les tâches en état « à réaliser » peuvent être modifiées. |

#### CM-G004 Assigner des travailleurs à une tâche

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’assigner un ou plusieurs travailleurs à une tâche. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est connecté sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des tâches. * Il existe au moins une tâche en état « à réaliser » dans la base de données ». * Il existe au moins un travailleur disponible dans la base de données. |
| ***Postconditions*** | * Un lien d’assignation a été créé en base de données entre un ou plusieurs travailleurs et une tâche. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option d’assignation de travailleur pour la tâche de son choix dans la liste. 2. Un pop-up apparaît avec une liste déroulante contenant le nom des différents travailleurs. 3. L’utilisateur choisit dans cette liste le ou les travailleurs à assigner à cette tâche. 4. L’utilisateur enregistre son choix en cliquant sur le bouton de confirmation. 5. Le pop-up disparait et un message de confirmation apparait, mentionnant que le la tâche a été modifiée avec succès. 6. La liste des tâches est mise à jour avec les changements enregistrés pour la tâche en question. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario commence au point 5 du scénario nominal.  5a. Un message d’erreur apparaît, mentionnant qu’une erreur s’est produite lors de l’enregistrement des modifications. |
|  | |
| ***Contraintes*** | Seule une tâche en état « à réaliser » peut se voir assigner des travailleurs. |

#### CM-G005 Visualiser la liste des travailleurs

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des travailleurs. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. |
| ***Postconditions*** | * La liste des travailleurs est affichée. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Travailleurs » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page de la liste des travailleurs. Les travailleurs sont affichés sous forme de liste. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Page de la liste des travailleurs

#### CM-G006 Créer un utilisateur travailleur

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de créer un utilisateur « travailleur ». |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des travailleurs. |
| ***Postconditions*** | * Un nouvel utilisateur avec le rôle de travailleur a été créé en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option de création d’un utilisateur travailleur, en cliquant sur le bouton prévu à cet effet. 2. Un pop-up apparaît, contenant un formulaire pour la création d’un utilisateur travailleur. 3. L’utilisateur remplit les champs du formulaire avec les informations du nouveau travailleur à créer et enregistre le nouveau travailleur en cliquant sur le bouton de confirmation. 4. Le pop-up de création disparait et un message de confirmation apparaît, mentionnant qu’un nouveau travailleur a été créé avec succès. 5. La liste des travailleurs est mise à jour avec le nouveau travailleur. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés du formulaire sont mis en évidence. |

Une image contenant texte, capture d’écran, ligne, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure : Créer un travailleur

#### CM-G007 Créer un animal

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de créer un nouvel animal en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des animaux (bétail). |
| ***Postconditions*** | * Un nouvel animal a été ajouté en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option de création d’un nouvel animal. 2. Un popup apparaît, contenant un formulaire pour la création d’un animal. 3. L’utilisateur remplit les champs du formulaire et enregistre le nouvel animal en cliquant sur le bouton de confirmation. 4. Le popup de création disparaît et un message de confirmation est affiché, mentionnant que le nouvel animal a été créé avec succès. 5. La liste des animaux est mise à jour avec le nouvel animal. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que l’animal n’a pas été créé.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs incorrects sont mis en évidence. |

Une image contenant texte, capture d’écran, diagramme, nombre

Le contenu généré par l’IA peut être incorrect.

Figure : Créer un animal

#### CM-G008 Modifier un animal

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de modifier un animal présent en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste du bétail. * Il existe au moins un animal en base de données |
| ***Postconditions*** | * Un animal a été modifié en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option de modification d’un animal. 2. Un popup apparaît, contenant un formulaire pour la modification de l’animal souhaité. Les champs du formulaire sont prés complétés avec les données actuelles de l’animal. 3. L’utilisateur modifie les champs souhaités du formulaire et enregistre les modifications en cliquant sur le bouton de confirmation. 4. Le popup de modification disparaît et un message de confirmation est affiché, mentionnant que l’animal a été modifié avec succès. 5. La liste des animaux est mise à jour avec les données modifiées de l’animal. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que les modifications de l’animal n’ont pas été enregistrées.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

#### CM-G009 Créer un groupe

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de créer un nouveau groupe en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des groupes. |
| ***Postconditions*** | * Un nouveau groupe a été ajouté en base de données |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option de création d’un nouveau groupe. 2. Un popup apparaît, contenant un formulaire pour la création d’un groupe. 3. L’utilisateur remplit les champs du formulaire et enregistre le nouveau groupe en cliquant sur le bouton de confirmation. 4. Le popup de création disparaît et un message de confirmation est affiché, mentionnant que le nouveau groupe a été créé avec succès. 5. La liste des groupes est mise à jour avec le nouveau groupe. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que le groupe n’a pas été créé.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |



Figure : Créer un groupe

#### CM-G010 Modifier un groupe

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de modifier un groupe présent en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des groupes. * Il existe au moins un groupe en base de données |
| ***Postconditions*** | * Un groupe a été modifié en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour le groupe souhaité l’option de modification d’un groupe. 2. Un popup apparaît, contenant un formulaire pour la modification du groupe souhaité. Les champs du formulaire sont prés complétés avec les données actuelles du groupe. 3. L’utilisateur modifie les champs souhaités du formulaire et enregistre les modifications en cliquant sur le bouton de confirmation. 4. Le popup de modification disparaît et un message de confirmation est affiché, mentionnant que le groupe a été modifié avec succès. 5. La liste des groupes est mise à jour avec les données modifiées du groupe. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que les modifications n’ont pas été enregistrées.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

#### CM-G011 Créer un enclos

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de créer un nouvel enclos en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des enclos. |
| ***Postconditions*** | * Un nouvel enclos a été ajouté en base de données |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’option de création d’un nouvel enclos. 2. Un popup apparaît, contenant un formulaire pour la création d’un enclos. 3. L’utilisateur remplit les champs du formulaire et enregistre le nouvel enclos en cliquant sur le bouton de confirmation. 4. Le popup de création disparaît et un message de confirmation est affiché, mentionnant que le nouvel enclos a été créé avec succès. 5. La liste des enclos est mise à jour avec le nouvel enclos. |
| ***Scénario alternatif*** |  |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que l’enclos n’a pas été créé.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

Une image contenant texte, capture d’écran, ligne, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure : Créer un enclos

#### CM-G012 Modifier un enclos

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de modifier un enclos présent en base de données. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. * L’utilisateur est présent sur la page de la liste des enclos. * Il existe au moins un enclos en base de données |
| ***Postconditions*** | * Un enclos a été modifié en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’enclos souhaité l’option de modification d’un enclos. 2. Un popup apparaît, contenant un formulaire pour la modification de l’enclos souhaité. Les champs du formulaire sont prés complétés avec les données actuelles de l’enclos. 3. L’utilisateur modifie les champs souhaités du formulaire et enregistre les modifications en cliquant sur le bouton de confirmation. 4. Le popup de modification disparaît et un message de confirmation est affiché, mentionnant que l’enclos a été modifié avec succès. 5. La liste des enclos est mise à jour avec les données modifiées de l’enclos. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que les modifications de l’enclos n’ont pas été enregistrées.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

#### CM-G013 Visualiser les statistiques du troupeau

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser les statistiques du troupeau et de la production laitière. |
| ***Acteurs*** | Gérant |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que gérant. |
| ***Postconditions*** | * Les statistiques du troupeau sont affichées. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Statistiques » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page des statistiques du troupeau. Sont affichées :  * Statistiques de répartition du troupeau par enclos, groupe, groupe d’âge, sexe. Ces statistiques sont affichées sous forme de diagrammes circulaires. * Statistiques de production laitière pour le mois en cours, affichée sous forme de diagramme linéaire.  1. L’utilisateur modifie la période d’affichage des statistiques de production laitière en modifiant les dates dans les champs prévus à cet effet. 2. Le diagramme de production laitière est mis à jour avec les données calculées pour la nouvelle période choisie. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur est affiché, mentionnant qu’aucune donnée de production laitière n’est disponible pour la période choisie. |

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Figure : Page des statistiques du troupeau

#### CM-T001 Visualiser la liste des tâches assignées

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de visualiser la liste des tâches qui lui sont assignées pour la journée. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. |
| ***Postconditions*** | * La liste des tâches du jour assignée à l’utilisateur est affichée. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Dashboard » dans le menu de navigation. 2. L’utilisateur est redirigé vers son dashbaord. Les tâches du jour assignées à cet utilisateur sont affichées sous forme de liste. 3. L’utilisateur peut filtrer les taches selon le statut. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

#### CM-T002 Mettre à jour le statut d’une tâche

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de mettre à jour le statut d’une tâche |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur son dashboard ou sur la page de la liste des tâches. |
| ***Postconditions*** | * Le statut de la tâche a été mis à jour en base de données. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour la tâche souhaitée l’option de mise à jour d’une tache. 2. Un pop-up apparaît comprenant une liste déroulante avec les différents statuts possibles pour une tâche. 3. L’utilisateur sélectionne dans la liste déroulante le nouveau statut de la tâche et confirme son choix en cliquant sur le bouton de confirmation. 4. Le popup disparaît et un message de confirmation est affiché, mentionnant que le la tâche à été mise à jour avec succès. 5. La liste des tâches est mise à jour avec le nouveau statut de la tâche concernée. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparaît, mentionnant que la tâche a été clôturée entre-temps.  Ce scénario d’erreur commence au point 4 du scénario nominal.  4b. Un message d’erreur apparaît, mentionnant que le travailleur n’est plus assigné à la tâche. |
|  | |
| ***Contraintes*** | Un travailleur ne peut mettre à jour qu’une tâche sur laquelle il est assigné. |

#### CM-T003 Transférer des animaux entre groupes

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de transférer un ou plusieurs animaux d’un groupe vers un autre. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des groupes |
| ***Postconditions*** | * Les animaux sélectionnés du premier groupe ont été transférés dans le second groupe. Les animaux sélectionnés du second groupe ont été transférés dans le second groupe. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour le groupe souhaité dans la liste des groupes l’option de transfert d’animaux. 2. Un pop-up en deux parties apparait. La partie gauche contient la liste des animaux composant le groupe choisi. La partie droite comprend une liste déroulante des autres groupes et une liste vide. 3. L’utilisateur sélectionne dans la liste déroulante le second groupe. La liste des animaux de ce groupe s’affiche dans l’encart prévu à cet effet. 4. L’utilisateur choisit un ou des animaux du premier groupe et les déplace dans le second groupe en cliquant sur les boutons prévus à cet effet. Les animaux sont visuellement transférés de la liste de gauche vers la liste de droite. 5. L’utilisateur enregistre son choix en cliquant sur le bouton de confirmation. 6. Le popup disparaît et un message de confirmation est affiché, mentionnant que les groupes concernés ont été mis à jour avec succès. 7. La liste des groupes est mise à jour avec la nouvelle composition des deux groupes concernés. |
| ***Scénario alternatif*** | Ce scénario alternatif commence au point 5 du scénario nominal.  5a. L’utilisateur choisit un ou des animaux du second groupe et les déplace dans le premier groupe en cliquant sur les boutons prévus à cet effet. Les animaux sont visuellement transférés de la liste de droite vers la liste de gauche.   * Reprise du scénario au point 5 du scénario nominal. |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 6 du scénario nominal.  6a. Un message d’erreur apparaît, mentionnant que l’un des groupes concernés a été modifié entre-temps et que l’action ne peut plus être réalisée. |

Une image contenant texte, capture d’écran, diagramme, Rectangle

Le contenu généré par l’IA peut être incorrect.

Figure : Transférer des animaux

#### CM-T004 Transférer des animaux entre enclos

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur de transférer un ou plusieurs animaux d’un enclos vers un autre. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des enclos |
| ***Postconditions*** | * Les animaux sélectionnés du premier enclos ont été transférés dans le second enclos. Les animaux sélectionnés du second enclos ont été transférés dans le second enclos. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour le enclos souhaité dans la liste des enclos l’option de transfert d’animaux. 2. Un pop-up en deux parties apparait. La partie gauche contient la liste des animaux composant le enclos choisi. La partie droite comprend une liste déroulante des autres enclos et une liste vide. 3. L’utilisateur sélectionne dans la liste déroulante le second enclos. La liste des animaux de cet enclos s’affiche dans l’encart prévu à cet effet. 4. L’utilisateur choisit un ou des animaux du premier enclos et les déplace dans le second enclos en cliquant sur les boutons prévus à cet effet. Les animaux sont visuellement transférés de la liste de gauche vers la liste de droite. 5. L’utilisateur enregistre son choix en cliquant sur le bouton de confirmation. 6. Le popup disparaît et un message de confirmation est affiché, mentionnant que les enclos concernés ont été mis à jour avec succès. 7. La liste des enclos est mise à jour avec la nouvelle composition des deux enclos concernés. |
| ***Scénario alternatif*** | Ce scénario alternatif commence au point 5 du scénario nominal.  5a. L’utilisateur choisit un ou des animaux du second enclos et les déplace dans le premier enclos en cliquant sur les boutons prévus à cet effet. Les animaux sont visuellement transférés de la liste de droite vers la liste de gauche.   * Reprise du scénario au point 5 du scénario nominal. |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 6 du scénario nominal.  6a. Un message d’erreur apparaît, mentionnant que l’un des enclos concernés a été modifié entre-temps et que l’action ne peut plus être réalisée. |

#### CM-T005 Encoder une traite

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder les volumes de traite pour un ou plusieurs animaux à une date sélectionnée. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * Au moins un animal de la base de données est une vache en statut allaitante. |
| ***Postconditions*** | * Un volume laitier a été encodé pour la date sélectionnée pour les animaux concernés. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne l’onglet « Traites » dans le menu de navigation. 2. L’utilisateur est redirigé vers la page d’encodage des traites, à la date du jour. La liste des vaches allaitantes est affichée sous forme de champs multiples. 3. L’utilisateur remplit le volume laitier pour les animaux voulus, en utilisant les champs ou les sliders prévus à cet effet. 4. L’utilisateur enregistre la traite du jour, complété pour tout ou partie des animaux, en cliquant sur le bouton de confirmation général. 5. La page de traite est mise à jour avec les volumes laitiers enregistrés pour les animaux concernés. |
| ***Scénario alternatif*** | Ce scénario alternatif commence au point 4 du scénario nominal  4a. L’utilisateur enregistre la traite du jour pour un seul animal, en cliquant sur le bouton de confirmation pour cet animal.  5a. La page de traite est mise à jour avec le volume laitier enregistré pour l’animal concerné.  Ce scénario alternatif commence au point 3 du scénario nominal.  3b. L’utilisateur sélectionne une autre date en utilisant le champ de sélection de date et confirme son choix.  4b. La page d’encodage des traites est mise à jour avec les données des volumes laitiers de la date sélectionnée.  => Reprise du scénario nominal au point 3. |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 4 du scénario nominal.  4a. Un message d’erreur apparaît, indiquant que la traite a déjà été enregistrée à cette date pour un ou plusieurs animaux concernés. |
|  | |
| ***Contraintes*** | Une seule traite par animal et par date peut être encodée. |

Une image contenant texte, capture d’écran, diagramme, nombre

Le contenu généré par l’IA peut être incorrect.

Figure : Page d'encodage des traites

#### CM-T006 Encoder une alerte pour un enclos

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une alerte pour un enclos. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la liste des enclos. |
| ***Postconditions*** | * Une alerte a été ajoutée en base de données pour un enclos. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’enclos souhaité l’option d’ajout d’une alerte. 2. Un popup apparaît contenant un formulaire pour l’ajout d’une alerte. 3. L’utilisateur remplit les champs du formulaire et enregistre l’alerte en cliquant sur le bouton de confirmation. 4. Le popup d’ajout disparaît. Un message de confirmation est affiché, mentionnant que l’alerte a été créée avec succès. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une alerte pour un enclos

#### CM-T007 Encoder une alerte pour un animal

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une alerte pour un animal. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la liste des animaux. |
| ***Postconditions*** | * Une alerte a été ajoutée en base de données pour un animal. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’ajout d’une alerte. 2. Un popup apparaît contenant un formulaire pour l’ajout d’une alerte. 3. L’utilisateur remplit les champs du formulaire et enregistre l’alerte en cliquant sur le bouton de confirmation. 4. Le popup d’ajout disparaît. Un message de confirmation est affiché, mentionnant que l’alerte a été créée avec succès. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | / |

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une alerte pour un animal

#### CM-T008 Encoder une intervention

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une intervention pour un animal. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * Une intervention a été ajoutée en base de données pour l’animal souhaité. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’encodage d’un nouvel évènement. 2. Une liste déroulante des différents types d’évènement apparaît. L’utilisateur choisit dans cette liste l’option « Intervention ». 3. Un popup apparaît, contenant un formulaire pour l’ajout d’une intervention. 4. L’utilisateur remplit les champs du formulaire et enregistre la nouvelle intervention en cliquant sur le bouton de confirmation. 5. Le popup d’ajout disparaît et un message de confirmation est affiché, mentionnant que l’intervention a été ajoutée avec succès. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 5 du scénario nominal.  5a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que l’intervention n’a pas été ajoutée.  Ce scénario d’erreur commence au point 4 du scénario nominal.  5b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

Une image contenant texte, capture d’écran, diagramme, nombre

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une intervention

#### CM-T009 Encoder une gestation

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une gestation pour un animal. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * Une gestation a été ajoutée en base de données pour l’animal souhaité. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’encodage d’un nouvel évènement. 2. Une liste déroulante des différents types d’évènement apparaît. L’utilisateur choisit dans cette liste l’option « Gestation ». 3. Un popup apparaît, contenant un formulaire pour l’ajout d’une gestation. 4. L’utilisateur remplit les champs du formulaire et enregistre la nouvelle gestation en cliquant sur le bouton de confirmation. 5. Le popup d’ajout disparaît et un message de confirmation est affiché, mentionnant que la gestation a été ajoutée avec succès. |
| ***Scénario alternatif*** |  |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 5 du scénario nominal.  5a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que la gestation n’a pas été ajoutée.  Ce scénario d’erreur commence au point 4 du scénario nominal.  5b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |
|  | |
| ***Contraintes*** | * Une gestation ne peut être encodée que pour un animal femelle * Une gestation ne peut être encodée que pour un animal de minimum 2 ans |

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une gestation

#### CM-T010 Encoder une affection

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une affection pour un animal. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * Une affection a été ajoutée en base de données pour l’animal souhaité. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’encodage d’un nouvel évènement. 2. Une liste déroulante des différents types d’évènement apparaît. L’utilisateur choisit dans cette liste l’option « Affection ». 3. Un popup apparaît, contenant un formulaire pour l’ajout d’une affection. 4. L’utilisateur remplit les champs du formulaire et enregistre la nouvelle affection en cliquant sur le bouton de confirmation. 5. Le popup d’ajout disparaît et un message de confirmation est affiché, mentionnant que l’affection a été ajoutée avec succès. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 5 du scénario nominal.  5a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que l’affection n’a pas été ajoutée.  Ce scénario d’erreur commence au point 4 du scénario nominal.  5b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

Une image contenant texte, capture d’écran, nombre, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une affection

#### CM-T011 Encoder un traitement

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder un traitement relatif à une affection pour un animal. |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * Un traitement a été ajoutée en base de données pour l’animal souhaité. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’encodage d’un nouvel évènement. 2. Une liste déroulante des différents types d’évènement apparaît. L’utilisateur choisit dans cette liste l’option « Traitement ». 3. Un popup apparaît, contenant un formulaire pour l’ajout d’un traitement. 4. L’utilisateur choisit dans une liste déroulante l’affection à laquelle se rapporte le traitement, remplit les champs du formulaire et enregistre le nouveau traitement en cliquant sur le bouton de confirmation. 5. Le popup d’ajout disparaît et un message de confirmation est affiché, mentionnant que le traitement a été ajouté avec succès. |
| ***Scénario alternatif*** |  |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 5 du scénario nominal.  5a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que le traitement n’a pas été ajouté.  Ce scénario d’erreur commence au point 4 du scénario nominal.  5b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

Une image contenant texte, capture d’écran, ligne, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder un traitement

#### CM-T012 Encoder une vaccination

|  |  |
| --- | --- |
| ***Description*** | Ce cas d’utilisation permet à un utilisateur d’encoder une vaccination |
| ***Acteurs*** | Travailleur |
|  | |
| ***Préconditions*** | * Un utilisateur est authentifié sur le portail en tant que travailleur. * L’utilisateur est présent sur la page de la liste des animaux. |
| ***Postconditions*** | * Une vaccination a été ajoutée en base de données pour l’animal souhaité. |
|  | |
| ***Scénario nominal*** | 1. L’utilisateur sélectionne pour l’animal souhaité l’option d’encodage d’un nouvel évènement. 2. Une liste déroulante des différents types d’évènement apparaît. L’utilisateur choisit dans cette liste l’option « Vaccination ». 3. Un popup apparaît, contenant un formulaire pour l’ajout d’une vaccination. 4. L’utilisateur remplit les champs du formulaire et enregistre la nouvelle vaccination en cliquant sur le bouton de confirmation. 5. Le popup d’ajout disparaît et un message de confirmation est affiché, mentionnant que la vaccination a été ajoutée avec succès. |
| ***Scénario alternatif*** | / |
| ***Scénario d’erreur*** | Ce scénario d’erreur commence au point 5 du scénario nominal.  5a. Un message d’erreur apparait, mentionnant qu’une erreur s’est produite et que la vaccination n’a pas été ajoutée.  Ce scénario d’erreur commence au point 4 du scénario nominal.  5b. Un message d’erreur apparaît, mentionnant qu’un ou plusieurs champs du formulaire sont incorrects. Les champs concernés sont mis en évidence. |

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Figure : Encoder une vaccination

## Wireframes de l’application mobile

Une image contenant texte, capture d’écran, Téléphone mobile, smartphone

Le contenu généré par l’IA peut être incorrect.

Figure : App mobile - Page de login

Une image contenant texte, Téléphone mobile, Appareil mobile, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Figure : App mobile - Page de liste du bétail

Une image contenant texte, capture d’écran, Téléphone mobile, Appareil mobile

Le contenu généré par l’IA peut être incorrect.

Figure : App mobile - Page détails d'un animal

Une image contenant texte, capture d’écran, Téléphone mobile, Appareil mobile

Le contenu généré par l’IA peut être incorrect.

Figure : App mobile - Encoder une traite

# Description des entités

**Utilisateurs**: Entité regroupant les informations des utilisateurs de l’application. Un utilisateur à le rôle de gérant ou de travailleur.

**Tâches** : Entité regroupant les informations relatives aux tâches quotidiennes à réaliser au sein de l’exploitation. Une tâche concerne une date précise, elle peut être générique impliquer un animal ou un enclos. Un ou plusieurs utilisateurs travailleurs peuvent être assignés à une tâche.

**Animaux**: Entité regroupant les informations relatives à un animal. Il s’agit d’une entité majeure de l’application. Un animal peut être lié à un ou deux autres animaux en tant que descendant.

**Groupes** : Entité décrivant un groupe d’animaux. Chaque animal peut être relié à un et un seul groupe.

**Enclos** : Entité décrivant les informations d’un enclos. Chaque animal peut être relié à un et un seul enclos.

**Traites** : Entité décrivant le volume laitier obtenu lors d’une traite. Une traite se réfère à un animal et à une date.

**Alertes** : Entité regroupant des informations temporaires à afficher afin d’informer les autres utilisateurs d’un problème relatif à l’exploitation, à un animal ou à un enclos. Une alerte possède également un niveau de priorité.

**Gestations** : Entité décrivant une gestation d’un animal. Elle possède une data de début et une date de vêlage calculée automatiquement sur base de la date initiale.

**Interventions**: Entité regroupant les informations relatives à une intervention extérieure sur un animal.

**Affections**: Entité décrivant une affection pour un animal telle que maladie ou une blessure.

**Traitements**: Entité décrivant un traitement administré à un animal. Un traitement peut se référer à une affection.

**Vaccinations**: Entité décrivant une vaccination administrée à un animal.

# Contraintes fonctionnelles

## Règles de structure

#### Utilisateurs

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS001 | Un utilisateur a un Id unique |
| CM-RS002 | Un utilisateur a un prénom, obligatoire |
| CM-RS003 | Un utilisateur a un nom, obligatoire |
| CM-RS004 | Un utilisateur a une adresse email unique, obligatoire |
| CM-RS005 | Un utilisateur peut avoir de 1 à plusieurs rôles |

#### Tâches

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS006 | Une tâche a un Id unique |
| CM-RS007 | Une tâche a un nom |
| CM-RS008 | Une tâche peut avoir une description |
| CM-RS009 | Une tâche a une date, obligatoire |
| CM-RS010 | Une tâche a un statut, obligatoire |
| CM-RS011 | Une tâche peut concerner un enclos |
| CM-RS012 | Une tâche peut concerner un animal |
| CM-RS013 | Une tâche peut être assignée à 0 ou plusieurs travailleurs |

#### Animaux

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS014 | Un animal a un Id unique |
| CM-RS015 | Un animal a un nom obligatoire |
| CM-RS016 | Un animal a un identifiant unique, obligatoire |
| CM-RS017 | Un animal peut être lié à un autre animal, qui est son père |
| CM-RS018 | Un animal peut être lié à un autre animal, qui est sa mère |
| CM-RS019 | Un animal a une date de naissance, obligatoire |
| CM-RS020 | Un animal a une race, obligatoire |
| CM-RS021 | Un animal a un sexe, obligatoire (mâle ou femelle) |
| CM-RS022 | Un animal peut être marqué comme allaitant ou non (booléen) |
| CM-RS023 | Un animal peut être lié à un groupe |
| CM-RS024 | Un animal peut être lié à un enclos |

#### Enclos

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS025 | Un enclos a un Id unique |
| CM-RS026 | Un enclos a un nom unique, obligatoire |
| CM-RS027 | Un enclos possède une capacité maximale (entier) |
| CM-RS028 | Un enclos peut posséder des coordonnées géographiques, stockées sous forme textuelle |

#### Groupes

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS029 | Un groupe a un Id unique |
| CM-RS030 | Un groupe a un nom unique, obligatoire |
| CM-RS031 | Un groupe peut avoir une description |

#### Traites

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS032 | Une traite a un Id unique |
| CM-RS033 | Une traite est liée à un et un seul animal |
| CM-RS034 | Une traite a une date, obligatoire |
| CM-RS035 | Une traite a un volume, obligatoire (Double) |
| CM-RS036 | Une traite peut être annulée ou non |

#### Alertes

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS037 | Une alerte a un Id unique |
| CM-RS038 | Une alerte a un nom unique, obligatoire |
| CM-RS039 | Une alerte peut avoir une description |
| CM-RS040 | Une alerte peut être liée à un animal |
| CM-RS041 | Une alerte peut être liée un enclos |
| CM-RS042 | Une alerte possède un type, obligatoire |
| CM-RS043 | Une alerte a un statut, obligatoire |

#### Gestations

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS044 | Une gestation a un Id unique |
| CM-RS045 | Une gestation est liée à un et un seul animal |
| CM-RS046 | Une gestation a une date de début, obligatoire |
| CM-RS047 | Une gestation a une date de vêlage, qui peut être nulle |
| CM-RS048 | Une gestation a un statut, obligatoire |

#### Interventions

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS049 | Une intervention a un Id unique |
| CM-RS050 | Une intervention est liée à un et un seul animal |
| CM-RS051 | Une intervention a un type, obligatoire |
| CM-RS052 | Une intervention a une date, obligatoire |

#### Affections

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS053 | Une affection a un Id unique |
| CM-RS054 | Une affection est liée à un et un seul animal |
| CM-RS055 | Une affection a un nom, obligatoire |
| CM-RS056 | Une affection a un type, obligatoire |
| CM-RS057 | Une affection a un statut, obligatoire |
| CM-RS058 | Une affection peut avoir une description textuelle |

#### Traitements

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS059 | Un traitement a un Id unique |
| CM-RS060 | Un traitement est lié à un et un seul animal |
| CM-RS061 | Un traitement peut être lié à une affection de cet animal |
| CM-RS062 | Un traitement à un nom, obligatoire |

#### Vaccination

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RS063 | Une vaccination a un Id unique |
| CM-RS064 | Une vaccination est liée à un et un seul animal |
| CM-RS065 | Une vaccination à un nom, obligatoire |
| CM-RS066 | Une vaccination possède un dosage, obligatoire |

## Règles de validation

#### Tâches

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RV001 | Une tâche en statut « Terminée » ne peut plus être modifiée |
| CM-RV002 | Seule une tâche en statut « A réaliser » peut se voir assigner des travailleurs |
| CM-RV003 | Le statut d’une tâche ne peut pas être mis à jour si la date de la tâche est supérieure à la date du jour |

#### Animaux

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RV004 | Seule un animal femelle peut être marqué comme « Allaitant » |

#### Enclos

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RV005 | Si un animal est assigné à un enclos dont la capacité maximale est déjà atteinte, une alerte est déclenchée |

#### Traites

|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RV006 | Une traite ne peut être encodée qu’une fois pour un animal à une date donnée |
| CM-RV007 | Aucun volume ne peut être encodé pour une traite annulée |
| CM-RV008 | Une traite ne peut être encodée que pour un animal marqué comme « allaitant » |

#### Gestations

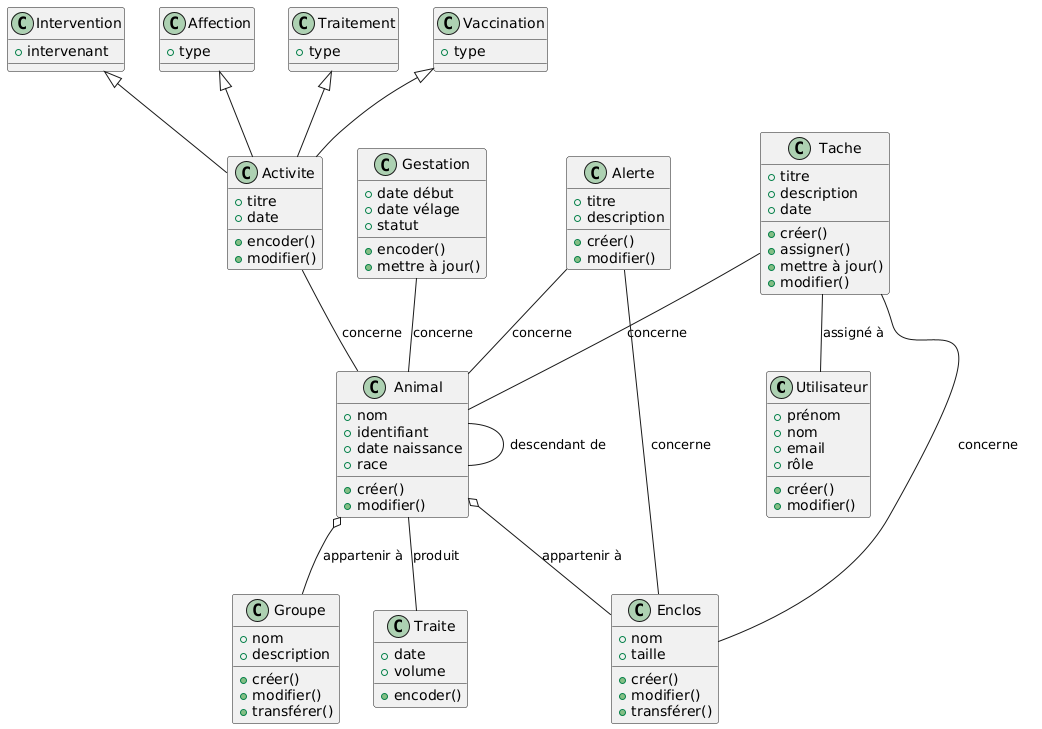
|  |  |
| --- | --- |
| ***Code*** | ***Description*** |
| CM-RV009 | Une gestation ne peut être ajouté que pour un animal femelle |
| CM-RV010 | Une gestation ne peut pas être ajouté pour un animal âgé de moins de 2 ans |
| CM-RV011 | Une gestation ne peut pas être ajouté pour un animal possédant déjà une gestation active |

# Schéma relationnel de la solution

Une image contenant texte, diagramme, Parallèle, Plan

Le contenu généré par l’IA peut être incorrect.

# Diagramme de classe



# Analyse Technique

## Choix technologiques

* La base de données relationnelle utilisera Microsoft SQL Server (Version 2022).
* La base de données NoSql utilisera MongoDB.
* Le Backend sera réalisé dans le framework .Net (.Net 8).
* L’accès aux données SQL Server se fera via l’ORM Entity Framework.
* L’application web sera réalisée avec le framework javascript Angular (Version 19).
* L’application mobile sera réalisée avec le framework Flutter (Version 3).

## Architecture applicative

### Base de données

Les données relatives aux entités telles que décrites dans le schéma relationnel seront contenues dans une base de données relationnelle SQL Server. L’intégrité relationnelle des données sera garantie par l’utilisation de clés étrangères.

Des données annexes, susceptible de changer avec le temps et dont l’intégrité référentielle n’est pas importante pour l’activité, seront stockées dans une base de données NoSql de type « documents ».

### Back-end

L’architecture back-end du projet sera structurée selon les principes de la « Clean Architecture ».

* Une couche « Domain » contient les classes représentant les entités liées au domaine d’activité. Dans ce type d’architecture, cette couche est le point de départ dont dépendent les autres couches. Elle ne dépend pas d’autres couches.
* Une couche « Application » contient la logique business. Elle dépend de la couche « Domain ».
* Une couche « Presentation » contient toute la partie liée à l’interface utilisateur. Elle dépend de la couche « Application ».
* Une couche « Persistence » contient la logique d’accès aux données. Elle dépend de la couche « Domain ».

Cette architecture présente plusieurs avantages :

* Le domaine d’activité est au centre du processus.
* Un couplage faible entre la logique business et les autres couches qui relèvent du détail d’implémentation : l’interface utilisateur et le stockage des données.

Description détaillée des couches architecturales

#### La couche « Persistence »

Cette couche permettra l’accès aux données. L’accès aux données relationnelles sera réalisé au moyen d’un ORM. Cette couche aura une dépendance sur la couche application de par l’implémentation d’interfaces qui seront définies dans la couche application. Elle aura également une dépendance directe sur la couche « Domain », nécessaire dans ce cas-ci du fait de l’utilisation d’un ORM spécifique (Entity Framework).

#### La couche « Domain »

Cette couche contiendra les entités dont dépendent les autres couches.

#### La couche « « Application »

Cette couche aura une dépendance directe sur la couche « Domain. Elle aura également une dépendance sur la couche Persistence. Les interfaces définies dans la couche application seront implémentées dans la couche Persitence et injectées par injection de dépendance.

#### La couche « Presentation »

Cette couche sera divisée entre une partie Backend et une partie FrontEnd.

Au niveau du backend, une Web API permettra de répondre à des requêtes HTTP au moyen de classes de contrôleurs. La Web API aura une dépendance directe sur la couche Application.

Une image contenant texte, diagramme, capture d’écran, Plan

Le contenu généré par l’IA peut être incorrect.

### Front-end

Pour la partie Frontend, une application web et une application mobile serviront d’interface utilisateur. Ces deux interfaces recevront les données via des requêtes HTTPS adressées à la Web API. Les contrôleurs de la Web API en charge de ces requêtes seront communs aux deux applications de front-end.

#### Application web

L’application web se présentera sous la forme d’une SPA (Single Page Application) et sera réalisée avec le Framework JavaScript Angular. Ce Framework fonctionne sur base de composants modulaires et intègre une gestion aisée des formulaires. Ces avantages le rendent parfaitement adapté aux exigences du projet. En outre, Angular est associé au langage Typescript, ce qui permet d’améliorer la qualité du code et de réduire les risques d’erreur d’exécution inhérents au langage Javascript. Enfin, il existe pour Angular des bibliothèques complètes de composants graphiques pré implémentés, qui permettent de garantir l’accessibilité utilisateur et la cohérence visuelle recherchée. Dans notre cas, nous utiliserons principalement les composants de la bibliothèque PrimeNG, laquelle se veut complète et robuste.

La gestion d’état (state management) de l’application web sera basée sur la bibliothèque NGXS, laquelle implémente une version simplifiée du design pattern CQRS pour la gestion du state global de l’application. Les composants Angular interagissent avec le back-end via des classes intermédiaires, appelées actions, lesquelles réalisent les appels vers le back-end au travers de services. Les données récupérées sont stockées dans le store. Les composants peuvent ensuite lire les parties du store nécessaires via des sélecteurs.

Une image contenant diagramme, texte, capture d’écran, cercle

Le contenu généré par l’IA peut être incorrect.

#### Application mobile

L’application mobile sera réalisée à l’aide du Framework Flutter et du langage de programmation Dart. Une application en flutter est composée de blocs appelés widgets, qui combinent le rendu de l’UI et la logique de code pour ce bloc. Ces widgets sont organisés sous forme d’arbre hiérarchique. Chaque widget gère son propre local state, et lorsque que celui-ci est modifié, l’application va regénérer la vue liée à ce widget et tous les widgets qui en dépendent.

Pour la gestion du global state de l’application mobile, il sera fait utilisation du package « provider », lequel permet de gérer un global state en tenant compte du fonctionnement hiérarchique des widgets en Flutter. Avec « Provider », chaque Widget va pouvoir directement interagir avec l’état global contenu dans un widget spécifique (ici appelé le contexte). Ce widget, situé au sommet de la hiérarchie, va transmettre les changements d’états aux widgets concernés, lesquels vont à leur tour déclencher une reconstruction des composants situés plus bas dans la hiérarchie.

Une image contenant texte, diagramme, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.

# Bilan et perspectives

## Bilan du projet

Nous pouvons évaluer la réalisation des objectifs du projet en fonction de différents axes majeurs : Les fonctionnalités, l’interface utilisateur et l’architecture logicielle.

Les fonctionnalités relatives aux exigences fonctionnelles ont été implémentées en respectant les contraintes fixées. Ces fonctionnalités sont en outre encadrées par des mécanismes d’authentification et d’autorisation. Il faut néanmoins signaler que la gestion des enclos et des prairies devrait idéalement être accompagnée d’un mécanisme de création cartographique. Celui-ci n’a pas été implémenté du fait de contraintes temporelles.

Par l’utilisation du Framework Angular et le choix d’une bibliothèque de composants graphiques, l’interface utilisateur mise en place est réactive, simple d’utilisation et présente la cohérence visuelle recherchée.

L’architecture logicielle choisie et implémentée pour les différentes composantes du projet s’est révélée très satisfaisante et a apporté un haut degré de flexibilité. Cela a permis un développement progressif et itératif des fonctionnalités.

Pour le back-end, l’implémentation des principes de la clean architecture dans le cadre d’une Web API s’est avéré être un bon choix pour rendre le projet modulaire et indépendant du reste des composants de l’application. Pour l’application web et l’application mobile, l’utilisation de mécanismes de gestion d’état a permis de garantir la cohérence des données entre le back-end et le front-end.

Plusieurs obstacles sont apparus au cours de la réalisation du projet. Le modèle de données initial a été remanié à plusieurs reprises car il s’est avéré inadéquat. Plusieurs pages et composants Angular de l’application web ont également dû être retravaillés en profondeur à la moitié du projet car ils n’étaient pas suffisamment réutilisables pour les fonctionnalités restantes.

La mise en œuvre d’une méthodologie agile a permis une planification efficace de l’implémentation des fonctionnalités et a apporté de la flexibilité face aux obstacles rencontrés tout au long du développement.

## Perspectives

L’architecture logicielle découplée du projet et la structure du modèle de données rendent l’application adaptable et extensible. En fonction des besoins des exploitants, des groupes de fonctionnalités pourraient être ajoutés progressivement et de manière indépendante.

* Gestion des stocks de nourritures. Ceux-ci pourraient être liés aux groupes de bétail ou à des enclos.
* Gestion des ventes de bétail, avec une éventuelle intégration à des solutions externes de comptabilité.
* Intégration directe avec des données provenant de solutions matérielles spécialisées, tels que des capteurs de température pour le bétail, des logiciels de salle de traite…

Afin d’améliorer la sécurité de l’application, nous recommandons également de mettre en place un mécanisme transversal de logging des actions des utilisateurs.

L’application mobile pourrait être étendue en y transposant l’ensemble des fonctionnalités de l’application web.

# Conclusion

Ce travail de fin d’étude constitue aussi bien une mise en pratique qu’une synthèse complète des compétences acquises et des méthodologies de travail enseignées tout au long de mon cursus à l’EPHEC. La réalisation de ce projet est une étape importante qui permet à l’étudiant de valider ses connaissances et de démontrer son aptitude à exercer les compétences requises dans un cadre professionnel.

Les objectifs du travail de fin d’étude donnent de l’importance aux différents aspects impliqués dans la réalisation d’un projet de développement informatique, et pas uniquement à la partie développement en tant que tel : l’analyse, la compréhension des besoins. Ces exigences constituent selon moi un point positif majeur de ce projet car elles nous préparent à la réalité du monde informatique, dans lequel l’analyse des besoins du client et la communication avec celui-ci se révèlent des aspects tout aussi importants que la connaissance des langages de programmation et les compétences de codage.

A titre personnel, cela m’a permis de faire le point sur mes compétences tant en termes de développement qu’en termes d’analyse. J’ai réalisé quels étaient mes points forts et j’ai pu déceler les compétences pour lesquelles il me faut encore combler des lacunes.

L’organisation du projet à l’aide la méthodologie Scrum a constitué pour moi un excellent exercice pratique et une illustration concrète des réalités d’un projet. Le fait de devoir constamment réévaluer les objectifs et les priorités s’est avéré d’une importance cruciale afin de pouvoir présenter à l’échéance un projet complet et cohérent.

En conclusion, je suis satisfait de pouvoir parachever mon apprentissage à travers ce projet, qui représente l’aboutissement de plusieurs années de défis personnels et professionnels. L’expérience et les compétences acquises durant mon cursus me permettent d’envisager sereinement et avec confiance la suite de ma carrière.

Deconninck Corentin

# Bibliographie

Eurostat (Novembre 2024). *A look at European Farms, European Census Results* (ISBN 978-92-68-23188-3)*.* Luxembourg Publications Office of the European Union. <https://ec.europa.eu/eurostat/web/products-catalogues/w/ks-01-24-024>

Parlement Européen, Direction générale des politiques internes (2017). *Recherche pour la Commission AGRI - Le secteur de l’élevage bovin dans l’Union Européenne, défis et perspectives : Lait et viande* (ISBN 978-92-846-1209-3). Union Européenne. <https://www.europarl.europa.eu/RegData/etudes/STUD/2017/585911/IPOL_STU(2017)585911_FR.pdf>

Statbel (05/12/2024). *Etat de l’agriculture Wallonne, Exploitations agricoles*. Statbel. [https://etat-agriculture.wallonie.be/contents/indicatorsheets/EAW-A\_II\_b\_1-1.html#](https://etat-agriculture.wallonie.be/contents/indicatorsheets/EAW-A_II_b_1-1.html)

Romane Bonnemé (29/07/2023). Des exploitations agricoles "à taille humaine" ? De moins en moins, surtout en Wallonie. *RTBF*. <https://www.rtbf.be/article/pourquoi-les-plus-grandes-exploitations-agricoles-du-pays-sont-elles-en-wallonie-11233498>

Roques P (2018). *UML 2.5 par la pratique* (8ème édition). Eyrolles

Angular team (2024). Angular. <https://angular.dev/>

PrimeNg (2025). PrimeNg V19. <https://primeng.org/>

Jimmy Bogard (2025). MediatR. <https://github.com/jbogard/MediatR>

OpenApiGenerator team. OpenApi Generator. <https://openapi-generator.tech/>

Auth0 (2025). Auth0. <https://auth0.com/docs>

Google (2025). Flutter. <https://docs.flutter.dev/>

Volodymyr Agafonkin (2023). Leaflet. <https://leafletjs.com/reference.html>