

Cahier des charges techniques et note de cadrage à destination de **DIGICHEESE**

Blocs de compétences:

Bloc 3&4: Élaboration du cahier des

Charges techniques et d'une note de cadrage.

Bloc 5 : Projet : Pilotage et management des équipes projet

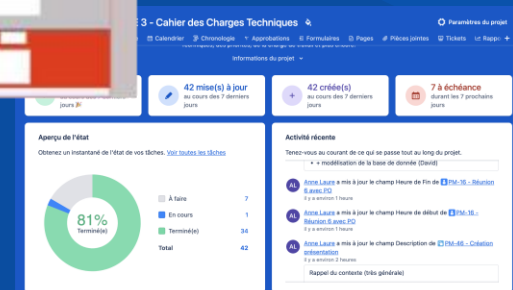
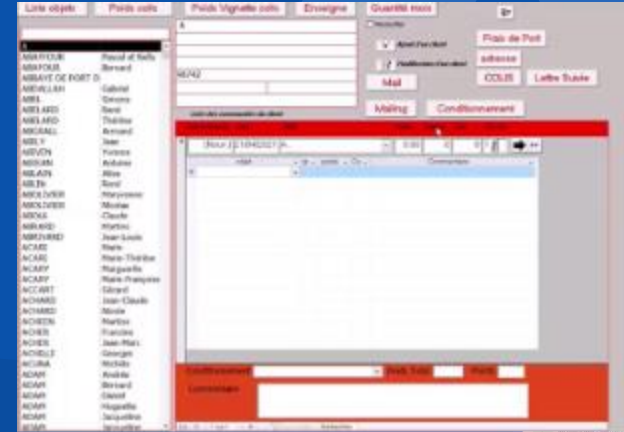


SOMMAIRE

- I. Contexte du projet
- II. Management du projet
- III. Note de cadrage
 - A. Objectifs
 - B. Base des Exigences techniques
 - C. Communication globale entre l'équipe et le client
 - D. Macro-Calendrier
 - E. Risque et contraintes
- IV. Cahier des charges techniques
 - A. Equipe du projet
 - B. Identification du besoin
 - C. Fonctionnalités principales pour les acteurs
 - D. Environnement technique
 - E. Calendrier détaillé
 - F. Budget
- V. Conclusion

PARTIE I

Contexte du projet et management des équipes



I. CONTEXTE DU PROJET



Refonte d'un Système Informatique (SI) pour une entreprise familiale DIGICHEESE.

Clients cibles de DIGICHEESE:

- distributeurs de grandes enseignes
- particuliers

Cœur du projet:

- Programme de fidélité sous forme de points à découper sur les produits, permettant aux clients de recevoir des goodies.
- La gestion de ce programme de fidélité nécessite une coordination entre différents intervenants (Opérateur colis, stock et un administrateur).

Objectif: réaliser une refonte complète d'une application de gestion pour répondre au besoin de maintenance et d'ergonomie.

Management de l'équipe

Concilier planification du cahier des charges techniques et de la note de cadrage et répartition des tâches.

Méthode utilisée : inspiré des méthodes agiles

- ➔ Kanban: Gestion de projet visuelle, organiser et de suivre les tâches de manière claire et efficace, gérer les priorités, affecter les tâches.
- ➔ Réunion tous les matins pour l'affectation des tâches

Outils utilisés:

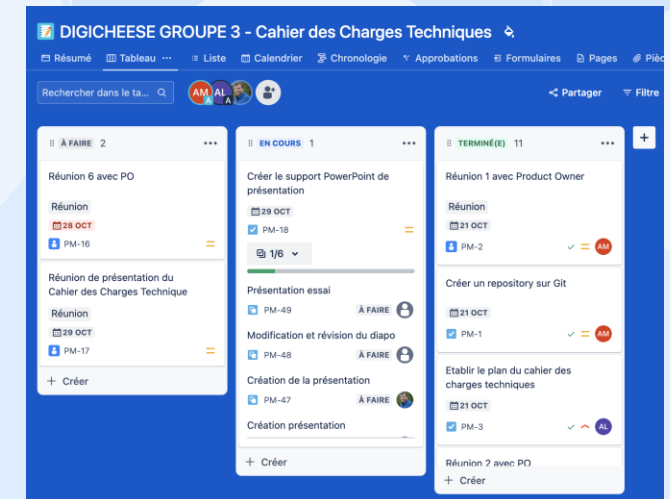
❖ Jira Software



❖ Jira Confluence



❖ Teams pour les réunions et échanges



II. MANAGEMENT DU PROJET

Kanban à date du 28 Octobre avant la réunion

DIGICHEESE GROUPE 3 - Cahier des Charges Techniques

Résumé Tableau Liste Calendrier Chronologie Approbations Formulaires Pages Pièces jointes

Rechercher dans le ta... Partager Filtre

À FAIRE 3

- Réunion 6 avec PO
 - Réunion
 - 28 OCT
 - PM-16
- Créer le support PowerPoint de présentation
 - 29 OCT
 - PM-18
 - 0/6
- Réunion de présentation du Cahier des Charges Technique
 - Réunion
 - 29 OCT
 - PM-17

EN COURS 0

+ Créer

TERMINÉ(E) 11

- Réunion 1 avec Product Owner
 - Réunion
 - 21 OCT
 - PM-2
- Créer un repository sur Git
 - 21 OCT
 - PM-1
- Etablir le plan du cahier des charges techniques
 - 21 OCT
 - PM-3
- Réunion 2 avec PO

+ Créer

Kanban à date du 28 Octobre dans la journée

DIGICHEESE GROUPE 3 - Cahier des Charges Techniques

Résumé Tableau Liste Calendrier Chronologie Approbations Formulaires Pages Pièces jointes

Rechercher dans le ta... Partager Filtre

À FAIRE 1

- Réunion de présentation du Cahier des Charges Technique
 - Réunion
 - 29 OCT
 - PM-17

+ Créer

EN COURS 1

- Créer le support PowerPoint de présentation
 - 29 OCT
 - PM-18
 - 3/6
- Présentation essai
 - PM-49
- Modification et révision du diapo
 - PM-48
- Création de la présentation
 - PM-47
- Création présentation

+ Créer

TERMINÉ(E) 12

- 25 OCT
 - PM-7
 - 11/11
- Réunion 5 avec PO
 - Réunion
 - 25 OCT
 - PM-15
- Rédaction de la note de cadrage
 - 28 OCT
 - PM-12
 - 9/9

+ Créer

Exemple de tâche: Rédaction du CDCT

PM-7

Rédiger le cahier des charges

Joindre

Ajouter un ticket enfant

Associer un ticket

Créer

Personnes

AM

 Alexis Murail

AL

 Anne Laure
 David Meaux

Description

Rédaction du cahier des charges en accord avec le plan pré-établi et validé par le PO

Tickets enfant

Progression : 100%

PM-21

 Création du sommaire

TERMINÉ(E)

PM-8

 Rédaction de la partie I

TERMINÉ(E)

PM-10

 Rédaction de la partie III

TERMINÉ(E)

AL

Ajouter un commentaire...

Joli !

Besoin d'aide ?

C'est bloqué...

Pourriez-vous clarifier... ?

C'

Conseil de pro : appuyez sur M pour commenter

Terminé(e)

Actions

Informations

Assigné(e)

David Meaux

Me l'assigner

Rapporteur

AL Anne Laure

Priorité

Medium

Étiquettes

Aucun

Date d'échéance

25 oct. 2024

Suivi temporel

Aucun temps enregistré

Inclure child issues

Start date

Aucun

Catégorie

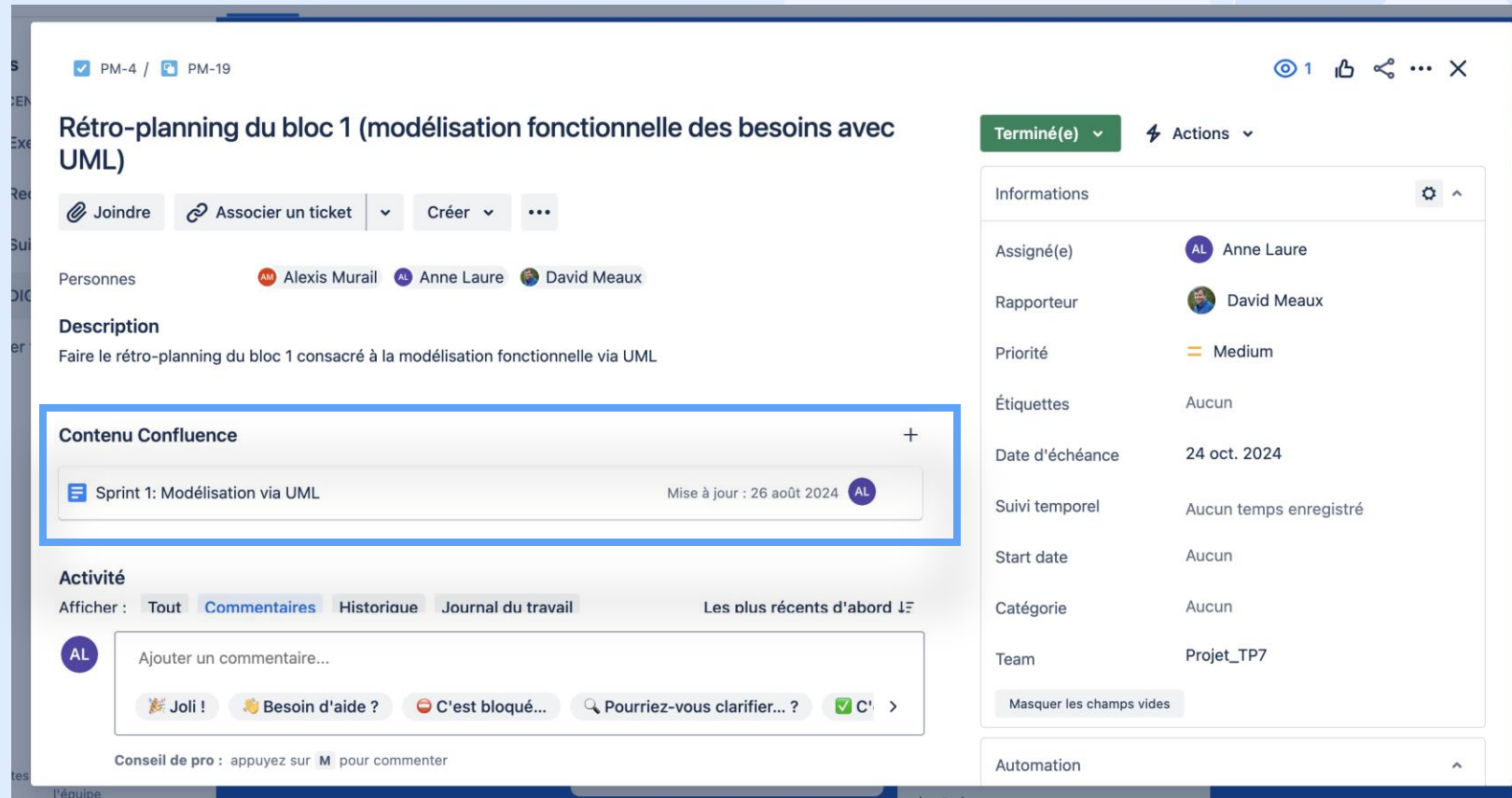
Aucun

Team

Projet_TP7

Masquer les champs vides

Connexion des tâches avec Jira confluence



PM-4 / PM-19

Rétro-planning du bloc 1 (modélisation fonctionnelle des besoins avec UML)

Joindre Associer un ticket Créer ...

Personnes Alexis Murail Anne Laure David Meaux

Description
Faire le rétro-planning du bloc 1 consacré à la modélisation fonctionnelle via UML

Contenu Confluence

Sprint 1: Modélisation via UML Mise à jour : 26 août 2024 AL

Activité
Afficher : Tout Commentaires Historique Journal du travail Les plus récents d'abord

AL Ajouter un commentaire...

Joli ! Besoin d'aide ? C'est bloqué... Pourriez-vous clarifier... ? C'est bon >

Conseil de pro : appuyez sur M pour commenter

Informations

Assigné(e) AL Anne Laure

Rapporteur David Meaux

Priorité Medium

Étiquettes Aucun

Date d'échéance 24 oct. 2024

Suivi temporel Aucun temps enregistré

Start date Aucun

Catégorie Aucun

Team Projet_TP7

Masquer les champs vides

Automation

PARTIE II

Note de cadragre



1. **Modélisation Fonctionnelle des besoins : Diagrammes UML**
2. **Développement BackEnd et API**
 - Gestion des Clients
 - Gestion des Commandes
 - Gestion des Colis
 - Gestion des Stocks
 - Gestion des Utilisateurs
3. **Conception du FrontEnd : Optimisation de l'expérience utilisateur**
4. **Déploiement sur site**

B. LES EXIGENCES TECHNIQUES



1. **Langages de Programmation** : Python et Java
2. **Base de données** : SQL
3. **Serveur SQL** : MariaDB
4. **Serveur Web** : Apache HTTP Server
5. **Outils de Développement** : Visual Studio Code et JetBrains
6. **Gestion de versions et de collaboration** : Github
7. **Tests** : PyTest, Swagger...
8. **Machine virtuelle Windows 2000**
9. **Infrastructure** : Déploiement sur infrastructure existante ou possibilité de migrer sur Cloud
10. **Sécurité** : Chiffrement des données sensibles et Gestion des rôles

- Communication par mail et par Teams
- **Product Backlog JIRA** pour la gestion du calendrier et l'attribution des tâches
- **Réunions de pilotage (COPIL)** : chaque vendredi à 16h
 - Via Teams le temps du développement
 - En présentiel lors du déploiement
 - Seront présents à minima :
 - Le Chef de Projet,
 - Le Lead Développeur,
 - Le Product Owner,
 - Un représentant DIGICHEESE

Semaines	Jours	S1 06/10 - 10/01	S2 13/01 - 17/01	S3 20/01 - 24/01	S4 27/01 - 31/01	S5 03/02 - 07/02	S6 10/02 - 14/02	S7 17/02 - 21/02	S8 24/02 - 28/02	S9 03/03 - 07/03
Fonctionnalités										
Analyse des besoins et Conception	10	SPRINT 1 Analyses et conception								
Design UX/UI et début Développement	10			SPRINT 2 UX et développement						
Développement	10					SPRINT 3 Développement				
Révision, Finalisation	5						SPRINT 4 Révision			
Déploiement	10							SPRINT 5 Déploiement		
	45									

- Budget alloué : 50 000€
- Date butoir : 31 mars 2025

1. Résistance au changement
2. Problèmes Techniques
3. Perte de données
4. Budget Insuffisant
5. Sécurité
6. Performance
7. Ralentissement de l'activité

PARTIE III

Cahier Des Charges Technique

III. CAHIER DES CHARGES TECHNIQUE

A. Equipe du Projet

(CDCT : § II)



Nom – Prénom	Rôle projet	Société	E-mail de contact
Fromagerie DIGICHEESE	Client principal	Fromagerie DIGICHEESE	contact@digicheese.com
Christophe GERMAIN	Directeur de Projets et PO	Diginamic.fr	cgermain@diginamic.fr
Robin HOTTON	PO	Diginamic.fr	rhotton@diginamic-formation.fr
Anne-Laure LEMAITRE	Cheffe de projet	Diginamic.fr	allemaitre@diginamic-formation.fr
Alexis MURAIL	Lead Dev / Développeur Full stack	Diginamic.fr	amurail@diginamic-formation.fr
David MEAUX	Développeur Backend	Diginamic.fr	dmeaux@diginamic-formation.fr
Marie CURIE	Testeuse	Diginamic.fr	mcurie@diginamic-formation.fr
Jules VERNE	Designer UI/UX	Diginamic.fr	jverne@diginamic-formation.fr

III. CAHIER DES CHARGES TECHNIQUE

B. Identification du besoin (CDCT : § III.1.C)

TROIS RÔLES (CUMULABLES) ONT ÉTÉ IDENTIFIÉS :

1. **Administrateur** : Initialement, la fonction d'administrateur du système avait un rôle de paramétrage (notamment la gestion des emballages, le calcul du montant de l'affranchissement, les poids des articles, etc.). **Dans la refonte de l'application**, l'administrateur devra pouvoir gérer les utilisateurs, les commandes, les objets (articles), les conditionnements, les poids, et le poids des vignettes.
2. **Opérateur colis** : L'opérateur colis traitait manuellement les colis réceptionnés et entrant les informations dans l'application Access. **Dans la refonte**, il devra pouvoir gérer les commandes, le mailing, les statistiques, créer des fiches clients et accéder à une interface d'impression.
3. **Opérateur stock** : Il était responsable de la gestion des stocks de goodies et des inventaires. **Lors de la refonte**, il devra pouvoir saisir les inventaires, gérer les stocks et accéder à une interface d'impression.

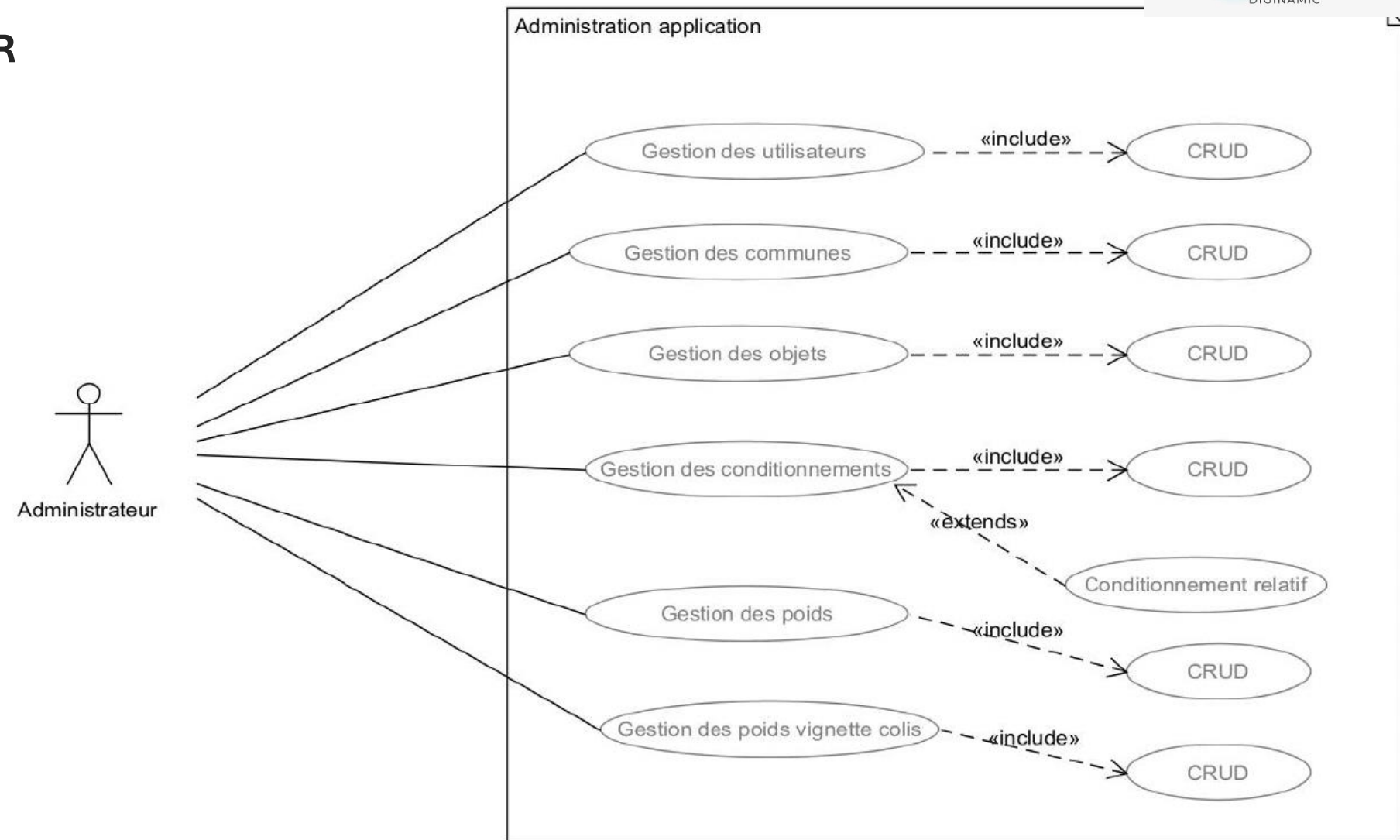
Ces rôles pouvant être cumulable, ce cas d'utilisation devra être pris en compte lors de la connexion à l'application. Une analyse des packages est disponible en Annexe 2.

DIGICHEESE fait donc appel à notre entreprise afin de réaliser une refonte complète d'une application de gestion en prenant en compte les problèmes précédemment cités mais également pour répondre au besoin de maintenance et d'ergonomie dans l'application.

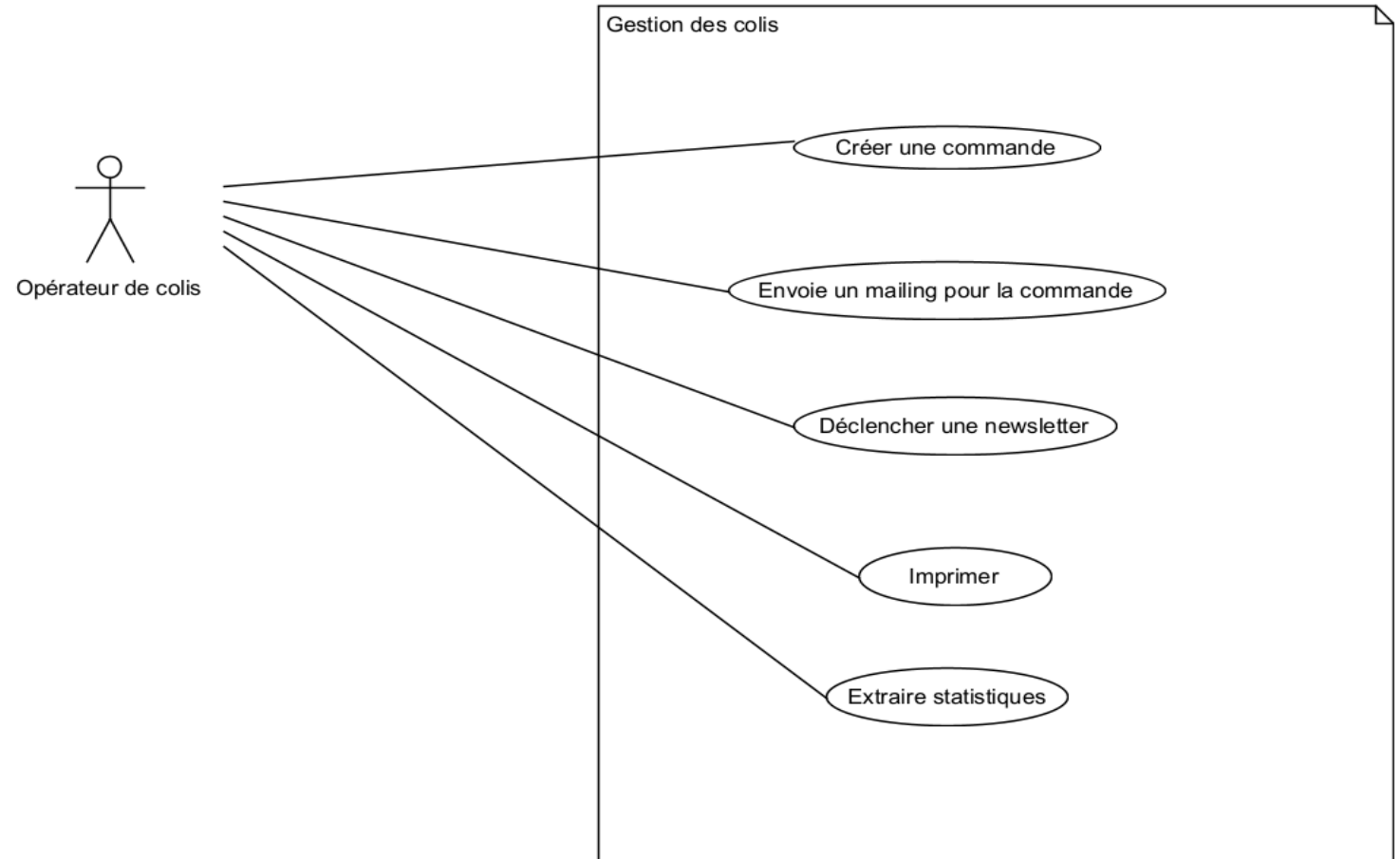
III. CAHIER DES CHARGES TECHNIQUE

C. Fonctionnalités principales pour les acteurs (CDCT : § IV.3.a)

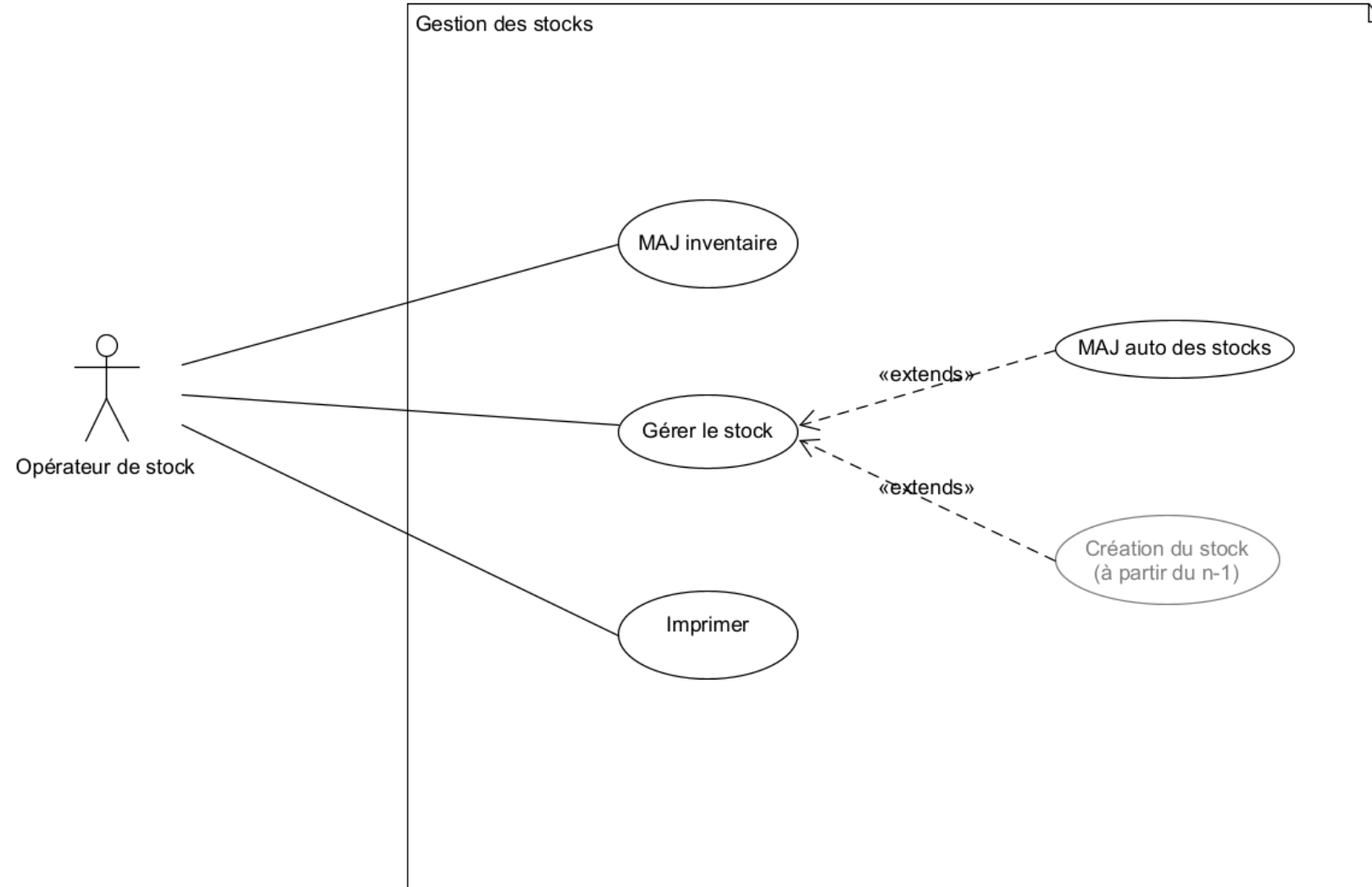
1. ADMINISTRATEUR



B. OPÉRATEUR COLIS



C. OPÉRATEUR STOCK



III. CAHIER DES CHARGES TECHNIQUE

D. Environnement technique

(CDCT: § IV.2.a, b)

Environnement informatique présentant les caractéristiques suivantes :

1. **Langage de programmation** : Python.
2. **Frameworks** : frontend (JavaScript), backend (Flask API, Python).
 - a. JavaScript est un langage de programmation supporté par tous les navigateurs web modern, un écosystème riche avec des bibliothèques comme bootstrap qui fournit des composants d'interface déjà stylisée.
 - b. L'API Flask est un micro-framework qui constituera le noyau de notre application web.
3. **Base de données** : Relationnelle (SQL) et utilisation de SQLAlchemy. SQLAlchemy est une bibliothèque Python qui établit une liaison entre le langage de programmation et les bases de données SQL.
4. **Infrastructure** : Les serveurs MariaDB et Apache HTTP seront hébergés par DIGICHEESE sur votre ordinateur Windows 2000. Cependant, nous pouvons proposer une infrastructure cloud géré soit par Microsoft (Azure) ou alors auprès d'autre service (AWS, Google).

Au niveau des outils de développements :

1. **Environnement de développement intégré (IDE)** : VS Code, JetBrains
2. **Outils de gestions des versions** : GitHub
3. **Environnement de test** : test unitaires (pytest), Swagger (vérification des routes de l'API), MyPy (tests statiques pour vérifier les types dans le code).
4. **Intégration continue et déploiement continu** : Tests unitaires et une machine virtuelle Windows 2000.
5. **Base de données (serveur SQL)** : MariaDB (gratuit) a des exigences matérielles très minimales et fonctionnera sur presque tout.
6. **Serveur web** : Le serveur HTTP Apache (gratuit) a des exigences matérielles très minimales et fonctionnera sur presque tout ce qui est équipé de Windows 2000 ou d'une version ultérieure.

ARCHITECTURE LOGICIELLE ET PROTOCOLE DE SÉCURITÉ

L'Architecture proposée :

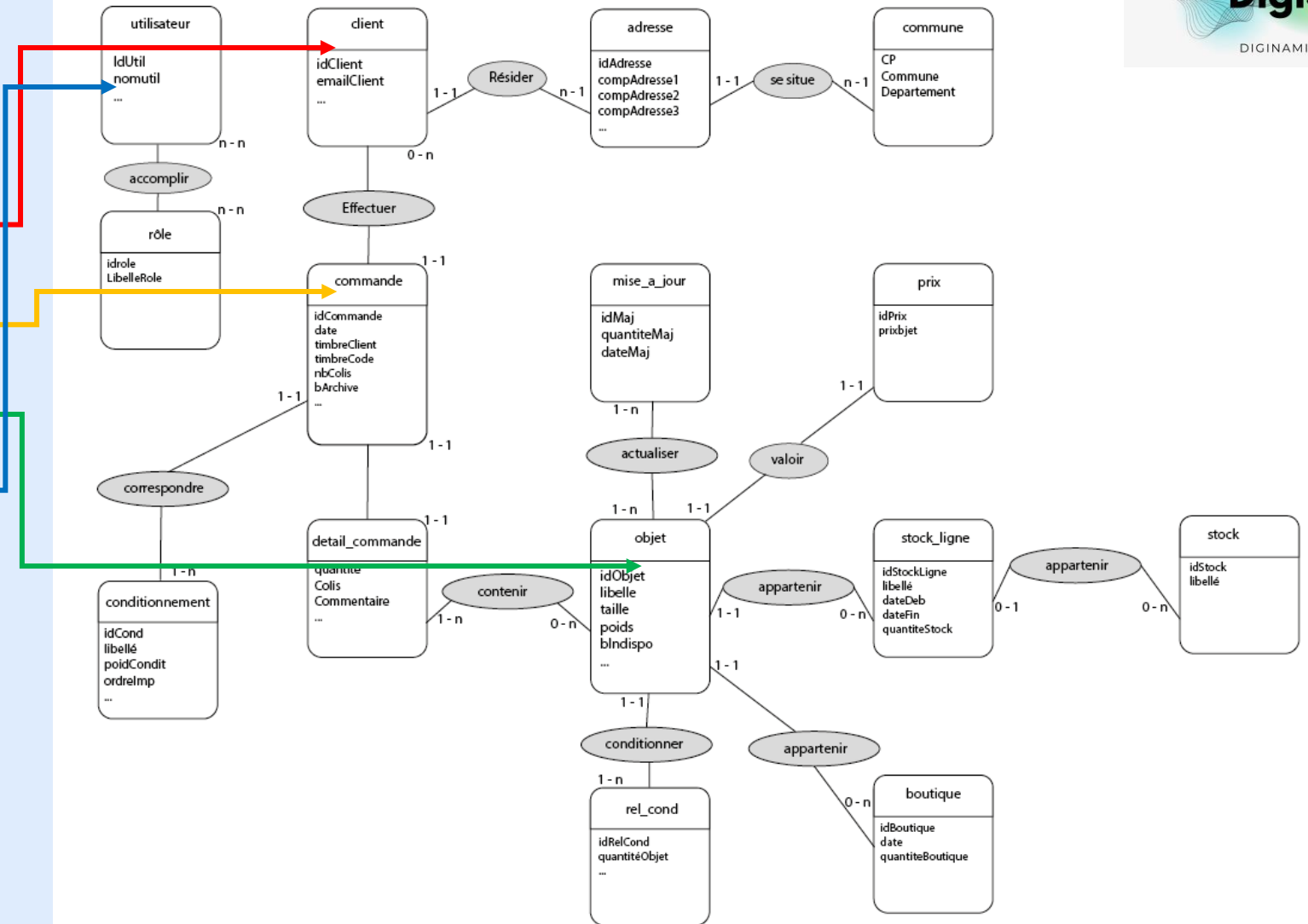
1. **Frontend** : l'utilisateur interagit avec une interface web développée JavaScript.
2. **Backend** : cette interface envoie des requêtes à l'API développée avec Flask API (Python) pour traiter les demandes de l'utilisateur.
3. **Accès aux données** : le backend envoie des requêtes à une base de données MariaDB (MySQL) pour récupérer ou enregistrer des informations.
4. **Sécurité** : l'authentification des utilisateurs est gérée via l'application et toutes les communications entre le client et le serveur sont chiffrées avec HTTPS.

L'Architecture logicielle :

1. **Modèle** : Architecture Microservice basée sur des API.
2. **Interaction entre les composants** : API (Flask API, SQLAlchemy)
3. **Sécurité** : communication sécurisée via HTTPS, chiffrement des données sensibles (données personnelles des clients), intégration de services tiers, par exemple pour l'authentification
4. **Gestion des rôles** : gestions différenciées selon les droits et le profils octroyé (Administrateur, Opérateur colis ou stock)

MODÉLISATION DE LA BASE DE DONNÉES

- Client
- Commande
- Objet
- Utilisateur



PLAN DE DEVELOPPEMENT

Méthode Agile : ➔ Méthode Scrum basée sur des sprints

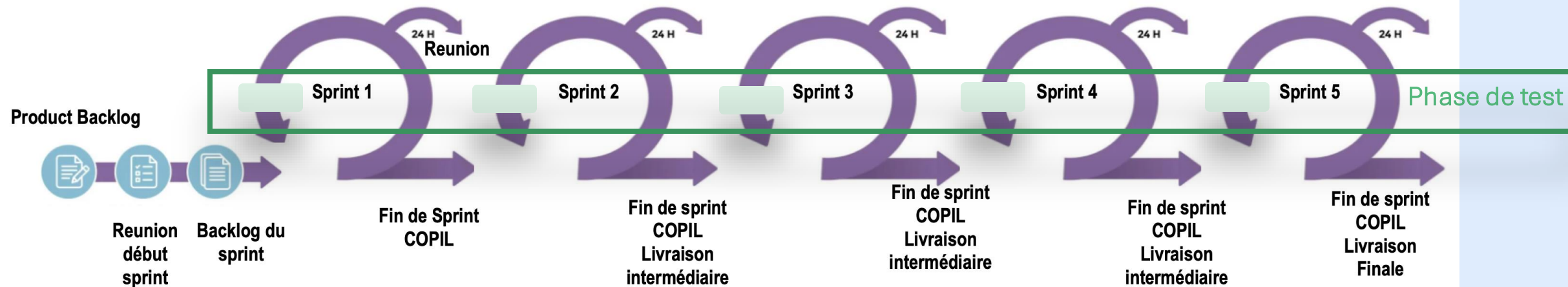
Sprint 1: Analyse des besoins et d'analyse fonctionnelle.

Sprint 2: Design et développement du backend et du frontend.

Sprint 3: Poursuite du développement du backend et du frontend.

Sprint 4: Phase de révision et de modification des développements

Sprint5: Déploiement et formation des nouveaux utilisateurs



PRODUCT BACKLOG

☐ ▾ Sprint 1 Analyses, Conception 6 janv. – 20 janv. (7 tickets)

SPRINT 1 Analyses et conception fonctionnelle

<input checked="" type="checkbox"/> DG-1	Reunion de cadrage		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-2	Analyses fonctionnelle		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-4	Définition des fonctionnalités		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-3	COPIL		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-5	Conception de l'architecture		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-6	Rédaction des documents		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-7	COPIL		À FAIRE ▾	-	

+ Créer un ticket

☐ ▾ Sprint 2 UX et développement 20 janv. – 3 févr. (11 tickets)

<input checked="" type="checkbox"/> DG-14	Conception et validation UX design		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-15	COPIL		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-16	Création de la base de données	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-45	Création de la Connexion à Flask API	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-21	Fonctionnalités Utilisateurs	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-20	Fonctionnalités Commandes	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-19	Fonctionnalités Objets	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-18	Fonctionnalités Clients	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-22	Front-end	FRONT-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-24	Session de test des développements réalisés		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-25	COPIL		À FAIRE ▾	-	

+ Créer un ticket

☐ ▾ Sprint 3 Développement 3 févr. – 17 févr. (8 tickets)

<input checked="" type="checkbox"/> DG-26	Poursuite du back-end	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-27	Swagger	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-32	Poursuite du front-end	FRONT-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-46	Intégration du front avec le back		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-28	Documentation Swagger	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-30	Session de test des développements réalisés	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-31	COPIL		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-43	Session de test des développements réalisés	FRONT-END	À FAIRE ▾	-	

+ Créer un ticket

☐ ▾ Sprint 4 Révision 17 févr. – 24 févr. (4 tickets)

<input checked="" type="checkbox"/> DG-33	Tâche et US à planifier selon les modification/ révisions à faire	BACK-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-34	Tâche et US à planifier selon les modification/ révisions à faire	FRONT-END	À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-35	Session de test des développements réalisés		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-36	COPIL		À FAIRE ▾	-	

+ Créer un ticket

☐ ▾ Sprint 5 Déploiement 24 févr. – 10 mars (7 tickets)

<input checked="" type="checkbox"/> DG-37	Déploiement		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-42	COPIL		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-38	Configuration utilisateurs		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-39	Formation		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-40	Suivi de fonctionnement		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-41	Test		À FAIRE ▾	-	
<input checked="" type="checkbox"/> DG-44	COPIL		À FAIRE ▾	-	

+ Créer un ticket

TEST:

Testeur externe au développeur :

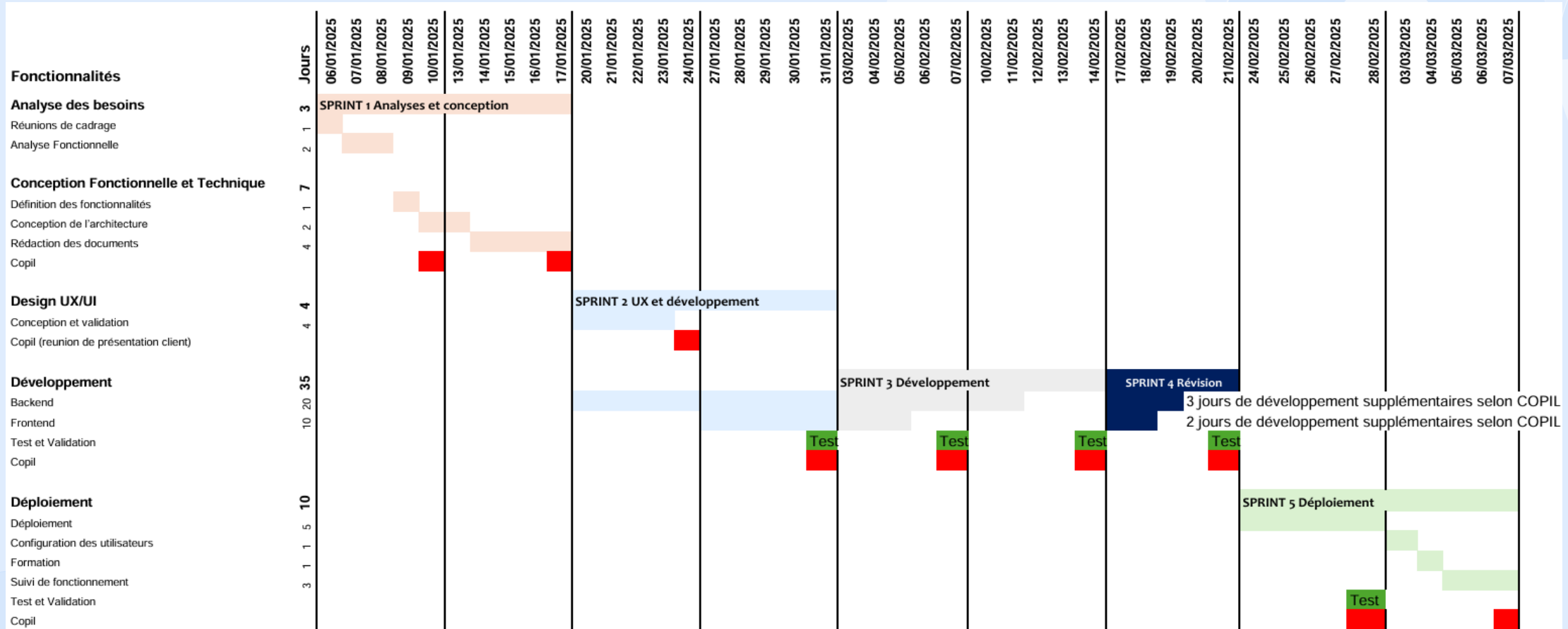
- Analyse objective et complète du développement effectué
- Éviter que les développeurs ne testent leurs propres développements, ce qui pourrait limiter la détection d'anomalies

Exemple de tests:

- Tests unitaires : Pour vérifier individuellement le fonctionnement d'une portion spécifique et isolée de code(fonction ou méthode):
- Tests fonctionnels : Pour s'assurer que l'application se comporte conformément aux spécifications fonctionnelles.
- Tests de non-régression : Pour garantir que les nouvelles fonctionnalités n'ont pas impacté les fonctionnalités existantes.
- Tests de performance : Pour évaluer la réactivité et la stabilité de l'application sous charge.

Test unitaires (pytest), MyPy (tests statiques pour vérifier les types dans le code).

CALENDRIER:



Tâches	Intervenants	Montant	Commentaires
Analyse des besoins	Chef de projet, Lead Dev, PO, utilisateurs finaux	2 400€	<ul style="list-style-type: none"> - Recueillir les besoins des utilisateurs finaux - Écrire les spécifications fonctionnelles
Conception Fonctionnelle et Technique	Chef de projet, Lead Dev	4 200€	<ul style="list-style-type: none"> - Définir les fonctionnalités minimales pour une première version - Concevoir l'architecture technique - Écriture du CDCT et de la Note de Cadrage
Design UX/UI	Chef de projet, Designer UX/UI	2 000€	<ul style="list-style-type: none"> - Design de l'interface utilisateur, Conception des écrans - Validation des maquettes
Développement	Chef de projet, Dev Backend, Dev Frontend	26 000€	<ul style="list-style-type: none"> - Gestion des données, API, Algorithme de calcul des coûts - Interface utilisateur, écrans de suivi des commandes et de gestion des stocks - Suivi du développement, Supervision
Tests et Validation	Chef de projet, Testeur	2 400€	<ul style="list-style-type: none"> - Tests unitaires, Tests d'intégration, Validation des utilisateurs - Coordination des tests
Déploiement	Chef de projet, Lead Dev	7 100€	<ul style="list-style-type: none"> - Déploiement sur infra client ou cloud - Configuration des utilisateurs - Formation des utilisateurs à l'outil - Assurance du bon fonctionnement de l'outil - Suivi du déploiement, Supervision
Comités de Pilotage	Chef de projet, Lead Dev, PO	2 400€	<ul style="list-style-type: none"> - Réunions de suivi après chaque grande étape
Maintenance (facultative)	Chef de projet, Développeur	(2 400€)	<ul style="list-style-type: none"> - Assurer une maintenance corrective sur la première année - Suivi des demandes et des correctifs
Cloud (facultatif)		(881.86€/mo.)	<ul style="list-style-type: none"> - Microsoft Azure SQL Server (758.96€/mo.) - App Service Basic Plan (122,90€/mo.)
Total		46 500€	(+ 2 400€ facultatifs) (+ 881,86€/mo. Cloud facultatifs)



1. Récapitulatif des Objectifs :

- Moderniser votre application actuelle
- Gestion optimisée des commandes, de la base clients
- Cadre sécurisé et conforme aux normes RGPD

2. Valeur Ajoutée :

- Architecture technique qui permettra d'avoir une interface moderne et intuitive
- Optimisation des performances
- Meilleure gestion de la Sécurité

3. Perspectives d'Évolution :

- En cas de futurs besoins, une nouvelle collaboration est possible pour améliorer le service

4. Prochaines Étapes :

- Après validation du cadre et des exigences techniques,
- À partir du début d'année prochaine, conformément à votre demande,
- Nous pourrons démarrer la conception détaillée et les premiers développements en suivant le calendrier fourni.

Merci de votre attention

DIGI3