République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de M'hamed Bougara

Département Informatique



Mémoire de Fin d'Études

Pour l'obtention du diplôme de Licence en Informatique

Conception et Développement d'une Application Web de Gestion de Maintenance des Équipements

Cas : Société SARPI

Réalisé par :

BELAID Kheireddine
ALSID CHIKH Sid ahmed
BOUDJERADA Ahmed zinelabidine
MERMOURI Abdessalem

Encadré par :

Benyoucef Aicha

Promotion : 2025/2026

Résumé

Ce mémoire présente la conception et le développement d'une application web de gestion de la maintenance corrective des équipements pour la société SARPI, filiale du groupe SONATRACH. Le processus de maintenance, bien que formalisé dans des procédures internes, repose aujourd'hui sur des échanges papier qui entraînent des pertes d'informations, un manque de traçabilité et des délais de validation trop longs.

L'application développée couvre l'intégralité du processus, de la réception de la Demande De Réparation (DDR) jusqu'à la mise à jour de la Fiche Historique de l'équipement, en passant par le diagnostic, l'établissement du devis, la planification des Ordres De Travail (ODT), l'exécution et le contrôle qualité. Chaque acteur dispose d'une interface adaptée à son rôle avec des droits d'accès stricts.

La solution a été réalisée avec Laravel 12, Tailwind CSS et Alpine.js, hébergée sous XAMPP, et modélisée grâce à PowerDesigner. Elle remplace les formulaires papier par des formulaires numériques interactifs avec visas électroniques et calcul automatique des coûts d'intervention.

Mots-clés : maintenance corrective, application web, Laravel, gestion des équipements, SARPI, traçabilité, visas électroniques.

Abstract

This thesis presents the design and development of a web application for managing corrective equipment maintenance at SARPI, a subsidiary of the SONATRACH group. Although the maintenance process is formally defined in internal procedures, it currently relies on paper-based exchanges that lead to information loss, lack of traceability and slow validation circuits.

The developed application covers the entire process, from the reception of the Repair Request (DDR) to the update of the equipment's History Sheet, including diagnosis, quotation, Work Order (ODT) planning, execution and quality control. Each actor has a role-specific interface with strict access rights.

The solution was built with Laravel 12, Tailwind CSS and Alpine.js, hosted under XAMPP, and modelled using PowerDesigner. It replaces paper forms with interactive digital forms featuring electronic signatures and automatic cost calculation.

Keywords : corrective maintenance, web application, Laravel, equipment management, SARPI, traceability, electronic signatures.

ملخص

تقدم هذه المذكرة تصميم وتطوير تطبيق ويب لإدارة الصيانة التصحيحية لمعدات شركة ساربي، الفرع التابع لمجموعة سوناطراك. على الرغم من أن عملية الصيانة موثقة في إجراءات داخلية رسمية، إلا أنها تعتمد حالياًا على تبادل الوثائق الورقية، مما يؤدي إلى ضياع المعلومات وغياب قابلية التتبع وطول دورات المصادقة.

يغطي التطبيق المطور العملية بأكملها، من استلام طلب الإصلاح حتى تحديث بطاقة سجل المعدة، مروراً بالتشخيص وإعداد عرض الأسعار وتخطيط أوامر العمل وتنفيذها ومراقبة الجودة. يتمتع كل مستخدم بواجهة مخصصة لدوره مع صلاحيات وصول صارمة.

Table des matières

Résumé **i**

Abstract **ii**

Résumé (Molakhas) **iii**

Table des matières **5**

Liste des tableaux **5**

Liste des figures **1**

1 Présentation de l'Organisme d'Accueil et Problématique **2**

Introduction 2

1.1 Présentation de la Société SARPI 2

1.1.1 Historique et Identité 2

1.1.2 Activités 2

1.1.3 Organisation 3

1.1.4 Parc Matériel 3

1.2 Le Processus de Maintenance Corrective 3

1.2.1 Les Types de Maintenance 3

1.2.2 Le Processus de Maintenance Corrective 3

1.2.3 Documents Officiels du Processus 4

1.3 Problématique et Objectifs 4

1.3.1 Limites du Processus Actuel 4

1.3.2 Question de Recherche 5

1.3.3 Objectifs du Projet 5

Conclusion 5

2 Analyse, Conception et Modélisation **6**

Introduction 6

2.1 Analyse des Besoins 6

2.1.1 Acteurs du Système 6

2.1.2 Besoins Fonctionnels 6

2.1.3 Besoins Non Fonctionnels 7
2.2 Conception de la Solution 8
2.2.1 Problèmes et Solutions 8
2.2.2 Architecture et Périmètre 8
2.3 Modélisation UML 8
2.3.1 Diagramme des Cas d'Utilisation 8
2.3.2 Diagrammes de Séquence 13
2.3.3 Diagramme de Classes 21
Conclusion 26

3 Réalisation et Mise en œuvre **27**
Introduction 27
3.1 Environnement de Développement 27
3.1.1 Visual Studio Code 27
3.1.2 XAMPP 27
3.1.3 PowerDesigner 27
3.2 Technologies Utilisées 28
3.2.1 Backend 28
3.2.2 Frontend 28
3.3 Structure de la Base de Données 28
Conclusion 29

Liste des tableaux

1.1 Documents officiels du processus de maintenance 4

2.1 Acteurs du système et leurs rôles 6
2.2 Problèmes identifiés et solutions retenues 8
2.3 Classes principales du système 26

3.1 Rôles définis dans le système 28
3.2 Principales tables de la base de données 29

Table des figures

2.1 Diagramme des cas d'utilisation – Vue globale 9
2.2 Cas d'utilisation : Gestionnaire Matériel / Chef Équipe (Chantier) 10
2.3 Cas d'utilisation : SDGMM 10
2.4 Cas d'utilisation : Bureau des Méthodes 11
2.5 Cas d'utilisation : Ingénieur d'Ordonnancement 12
2.6 Cas d'utilisation : Chef d'Atelier 13
2.7 Cas d'utilisation : Gestionnaire Matériel (DR) 13
2.8 Séquence 1 : Signalement et Demande de Réparation 14
2.9 Séquence 2 : Diagnostic de l'Équipement 15
2.10 Séquence 3 : Établissement et Validation du Devis 16
2.11 Séquence 4 : Planification des Travaux 17
2.12 Séquence 5 : Émission et Gestion des ODT 18
2.13 Séquence 6 : Exécution et Contrôle des Travaux 19
2.14 Séquence 7 : Réception et Clôture 20
2.15 Diagramme de classes – Vue globale 21
2.16 Zoom 1 : Utilisateur et ses Relations 22
2.17 Zoom 2 : Équipement et Demandes 22
2.18 Zoom 3 : Diagnostic et Devis 23
2.19 Zoom 4 : ODT et Exécution 24
2.20 Zoom 5 : Rapport d'Intervention et Clôture 25

Chapitre 1

Présentation de l'Organisme d'Accueil et Problématique

Introduction

Ce chapitre présente SARPI, l'entreprise dans laquelle s'inscrit notre projet. On y décrit son activité, son organisation et le processus de maintenance de ses équipements tel qu'il fonctionne aujourd'hui, avant d'exposer les problèmes concrets que notre application cherche à régler.

1.1 Présentation de la Société SARPI

1.1.1 Historique et Identité

La Société Algérienne de Réalisation de Pipelines et d'Infrastructures (SARPI) est une entreprise publique algérienne filiale du groupe SONATRACH. Elle opère dans le secteur des travaux pétroliers et gaziers depuis sa création, et s'est progressivement imposée comme un acteur incontournable dans la construction et la maintenance des infrastructures énergétiques en Algérie [1].

1.1.2 Activités

SARPI intervient dans quatre domaines principaux liés aux hydrocarbures : la construction de pipelines (réseaux de transport de pétrole, gaz et condensat), les travaux de génie civil pétrolier (stations de pompage, bacs de stockage), la maintenance des infrastructures existantes, et les travaux d'électricité et d'instrumentation.

1.1.3 Organisation

L'entreprise est organisée autour de plusieurs entités complémentaires [1] :

— **Direction Générale (PDG)** : pilotage stratégique.
— **Direction Logistique (DLG)** : gestion du parc matériel.
— **SDGMM** : définit la politique de maintenance, élabore les procédures et les plans d'entretien préventif (P.E.P).
— **Directions Régionales (DR)** : DR Hassi Messaoud et DR Hassi R'Mel, chacune avec ses propres ateliers de maintenance.
— **Projets** : équipes opérationnelles sur les chantiers.

1.1.4 Parc Matériel

Le parc comprend des engins de terrassement (bulldozers, pelles mécaniques), des engins de levage (grues, chariots élévateurs), des véhicules de chantier, des équipements de soudage et de contrôle. La disponibilité de ce matériel conditionne directement la capacité de SARPI à tenir ses délais contractuels [1].

1.2 Le Processus de Maintenance Corrective

La maintenance chez SARPI est régie par la procédure interne **PRO-SMI-PS2-03** (révision 05, mars 2025) [2]. Elle distingue deux types d'interventions.

1.2.1 Les Types de Maintenance

Maintenance préventive. Actions programmées selon les préconisations des constructeurs, planifiées dans un Plan d'Entretien Préventif (P.E.P) annuel. Elle comprend l'entretien journalier (vérification des niveaux, contrôle visuel), le contrôle périodique (graissage, réglage sans démontage) et la révision générale (démontage complet et passage sur banc d'essais).

Maintenance corrective. Interventions suite à une panne. L'objectif est de remettre l'équipement en service au plus vite, en limitant l'impact sur le chantier.

1.2.2 Le Processus de Maintenance Corrective

Quand une panne dépasse les capacités de l'équipe sur site, le processus formel suivant est déclenché [2] :

1. **Signalement et DDR.** Le Gestionnaire Matériel émet une Demande d'Intervention (DI). La SDGMM l'analyse puis émet une Demande De Réparation (DDR) à la DR compétente.

2. **Réception et Diagnostic.** Le Bureau des Méthodes ouvre un dossier. Le Chef d'Atelier examine l'équipement, identifie la panne et les pièces nécessaires. Les résultats sont consignés dans une Fiche Diagnostic.
3. **Devis.** Le Bureau des Méthodes établit un devis (travaux, délais, coût). La SDGMM le valide ou le refuse. En cas de refus, une proposition de réforme est établie.
4. **Planification.** L'Ingénieur d'Ordonnancement planifie les phases de travaux, estime les charges par qualification et vérifie la disponibilité des Pièces De Rechange (PDR). Si les PDR manquent, l'ODT est mis en instance.
5. **Ordres De Travail et Exécution.** L'ODT est émis par phase dès confirmation des PDR. Le Chef d'Atelier supervise les travaux et renseigne les relevés journaliers.
6. **Contrôle, Réception et Clôture.** Le Bureau des Méthodes contrôle la qualité des travaux. Le Gestionnaire Matériel effectue les essais. L'Ingénieur d'Ordonnancement clôture l'ODT en calculant les temps et coûts réels. Le Bureau des Méthodes établit le Rapport d'Intervention et met à jour la Fiche Historique.

1.2.3 Documents Officiels du Processus

TABLE 1.1: Documents officiels du processus de maintenance

N°	Document	Rôle
1	Demande d'Intervention (DI)	Signalement initial
2	Demande De Réparation (DDR)	Déclenchement du processus
3	Fiche Diagnostic	Résultats de l'examen
4	Devis de réparation	Coût et délais estimés
5	Ordre De Travail (ODT)	Planification et suivi
6	Rapport d'Intervention	Bilan des travaux
7	Fiche Historique	Traçabilité par équipement

1.3 Problématique et Objectifs

1.3.1 Limites du Processus Actuel

Le processus est bien défini sur le papier, mais son exécution repose entièrement sur des échanges manuels. Concrètement, ça pose plusieurs problèmes :

— **Perte et dispersion d'informations :** les dossiers papier se perdent ou se retrouvent mal classés. La dispersion géographique des chantiers rend la consolidation difficile.

— **Pas de suivi en temps réel :** pour connaître l'état d'une réparation, il faut appeler ou attendre qu'un document circule.

— **Validations lentes :** chaque visa nécessite la transmission physique du dossier entre services, ce qui allonge les délais.

— **Historique incomplet :** la mise à jour de la Fiche Historique est souvent négligée dans un environnement papier.
— **Suivi des coûts impossible :** calculer le coût réel d'une intervention demande de regrouper manuellement des données issues de plusieurs documents.

1.3.2 Question de Recherche

Comment concevoir une application web qui permette à SARPI de gérer le processus de maintenance corrective de façon centralisée et traçable, depuis la réception de la DDR jusqu'à la clôture de l'ODT, en remplaçant les supports papier par des formulaires numériques accessibles à chaque acteur selon son rôle ?

1.3.3 Objectifs du Projet

— Numériser tous les formulaires du processus et les rendre accessibles via une interface web.
— Centraliser les données dans une base unique.
— Permettre un suivi en temps réel de chaque intervention, du signalement à la remise en service.
— Gérer les droits d'accès par rôle.
— Donner aux responsables une vue claire sur l'état du parc et les coûts de maintenance.

Conclusion

Ce premier chapitre a posé le cadre du projet : une entreprise avec un processus de maintenance bien défini mais entièrement manuel, et des problèmes concrets qui justifient le développement d'une solution informatique. Le chapitre suivant présente l'analyse des besoins et la conception de cette solution.

Chapitre 2

Analyse, Conception et Modélisation

Introduction

Ce chapitre couvre l'analyse des besoins, la conception de la solution et la modélisation UML du système. C'est ici qu'on définit ce que l'application doit faire, qui l'utilise et comment elle est structurée.

2.1 Analyse des Besoins

2.1.1 Acteurs du Système

Six acteurs ont été identifiés, chacun avec un rôle précis dans le processus [2] :

Table 2.1: Acteurs du système et leurs rôles

Acteur	**Entité**	**Rôle**
Gestionnaire Matériel / Chef Équipe	Chantier	Signale les pannes, émet les DI, réceptionne l'équipement réparé
SDGMM	Direction centrale	Traite les DI, émet les DDR, valide les devis
Bureau Méthodes	DR	Gère le diagnostic, les devis, le rapport et le contrôle qualité
Chef Atelier	DR	Crée la fiche diagnostic et supervise les travaux
Ingénieur Ordonnancement	DR	Planifie et émet les ODT
Gestionnaire Matériel (DR)	DR	Effectue les essais et réceptionne l'équipement

2.1.2 Besoins Fonctionnels

Les fonctionnalités sont organisées par phase du processus :

— **Demandes de réparation :** signalement de panne, émission de DI, création de DDR avec numérotation automatique, visas électroniques.
— **Diagnostic :** création de la Fiche Diagnostic par le Chef d'Atelier après visa du Bureau des Méthodes sur la DDR, saisie des PDR, visas.
— **Devis :** établissement par le Bureau des Méthodes, validation ou refus par la SDGMM, proposition de réforme en cas de refus.
— **Planification et ODT :** planification par phases, charges par qualification, gestion de la disponibilité des PDR, émission de DST si nécessaire.
— **Exécution :** saisie des exécutants et relevés journaliers, contrôle qualité, gestion des reprises en cas de non-conformité.
— **Réception et clôture :** enregistrement des essais, clôture de l'ODT avec calcul automatique des temps et coûts réels, rapport final, mise à jour de la Fiche Historique.
— **Gestion des accès :** authentification, droits stricts par rôle.

2.1.3 Besoins Non Fonctionnels

— **Sécurité :** authentification et contrôle des accès par rôle sur chaque route.
— **Performance :** temps de réponse acceptable même avec plusieurs utilisateurs simultanés.
— **Disponibilité :** accessible en permanence pendant les heures de travail.
— **Maintenabilité :** code structuré et documenté pour faciliter les évolutions.
— **Ergonomie :** interface adaptée au profil de chaque acteur.
— **Traçabilité :** toute action (création, modification, visa) est horodatée et associée à l'utilisateur.

2.2 Conception de la Solution

2.2.1 Problèmes et Solutions

TABLE 2.2: Problèmes identifiés et solutions retenues

Problème	**Solution**
Gestion papier, risques de perte	Numérisation de tous les formulaires (DDR, Fiche Diagnostic, ODT, Rapport)
Pas de traçabilité en temps réel	Statuts dynamiques pour chaque document et intervention
Validations lentes par transmission physique	Visas électroniques en ligne
Historique incomplet et difficile à consulter	Base centralisée, Fiche Historique mise à jour automatiquement
Suivi des coûts impossible	Calcul automatique à la clôture de l'ODT
Erreurs de transcription entre documents	Saisie unique, propagée dans tous les documents liés

2.2.2 Architecture et Périmètre

L'application repose sur une architecture trois tiers : couche présentation (navigateur web), couche métier (serveur Laravel) et couche données (base MySQL). Le périmètre couvre la maintenance corrective au niveau des DR, de la réception de la DDR à la clôture de l'ODT, en huit modules : Authentification, DDR, Diagnostic, Devis, ODT, Exécution, Réception et Historique.

2.3 Modélisation UML

2.3.1 Diagramme des Cas d'Utilisation

Le diagramme des cas d'utilisation représente les interactions entre les six acteurs et le système. Deux relations sont utilisées : ≪**include**≫ (inclusion obligatoire) et ≪**extend**≫ (extension conditionnelle). Le diagramme est organisé par acteur, chacun regroupant les cas d'utilisation qui lui sont propres.

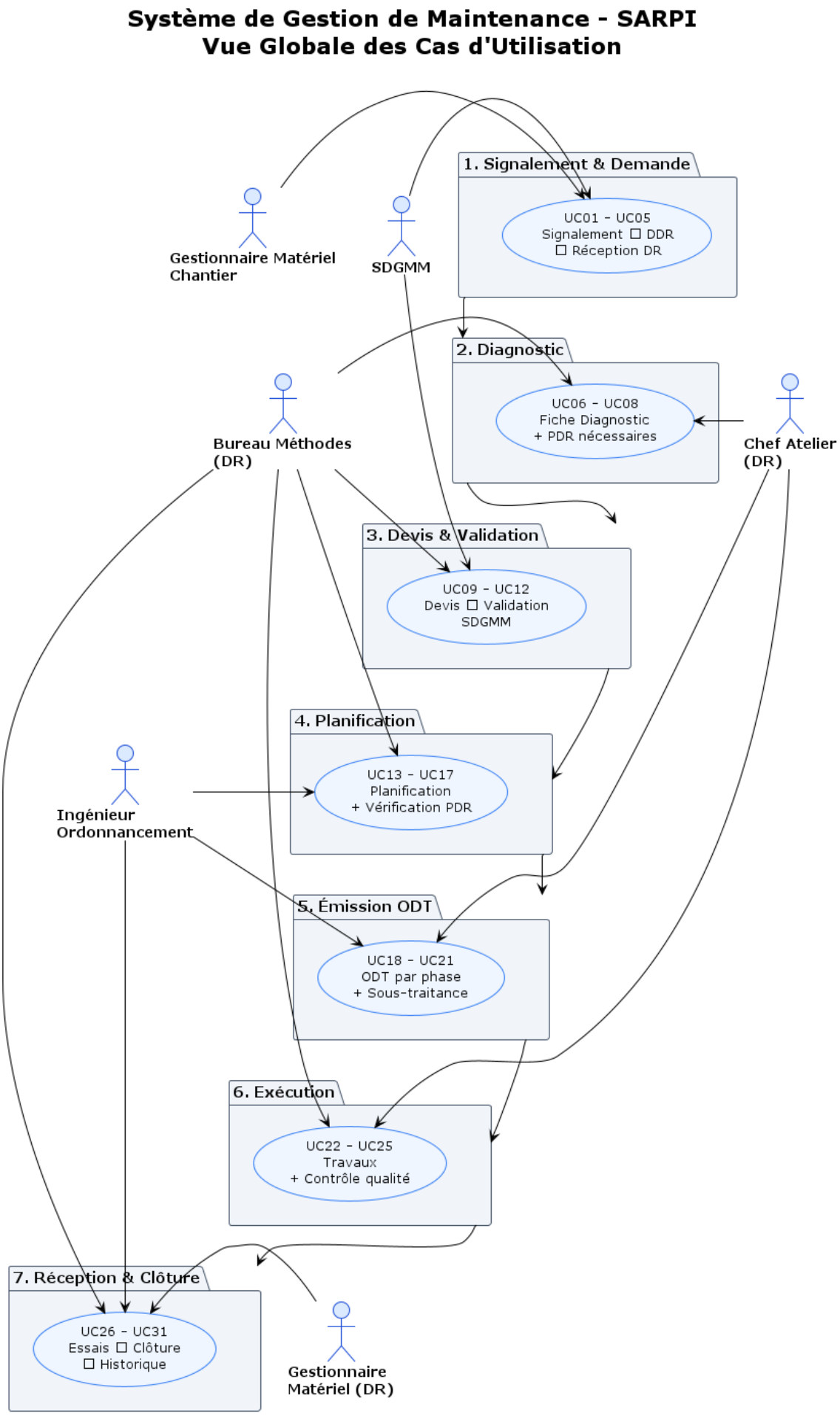


FIGURE 2.1: Diagramme des cas d'utilisation – Vue globale

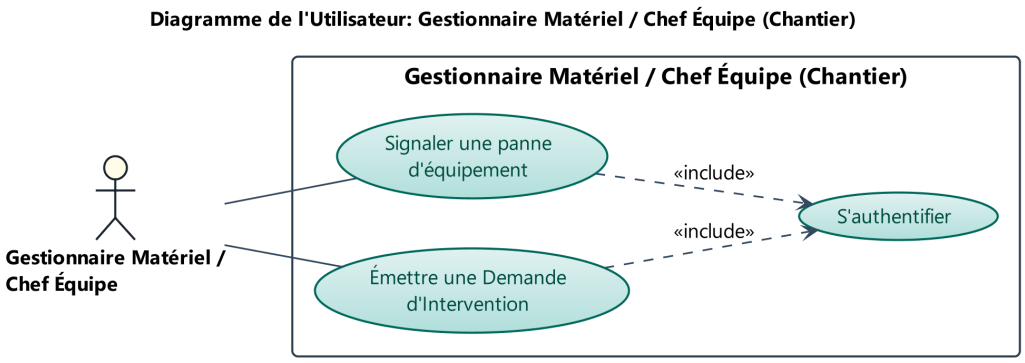


FIGURE 2.2: Cas d'utilisation : Gestionnaire Matériel / Chef Équipe (Chantier)

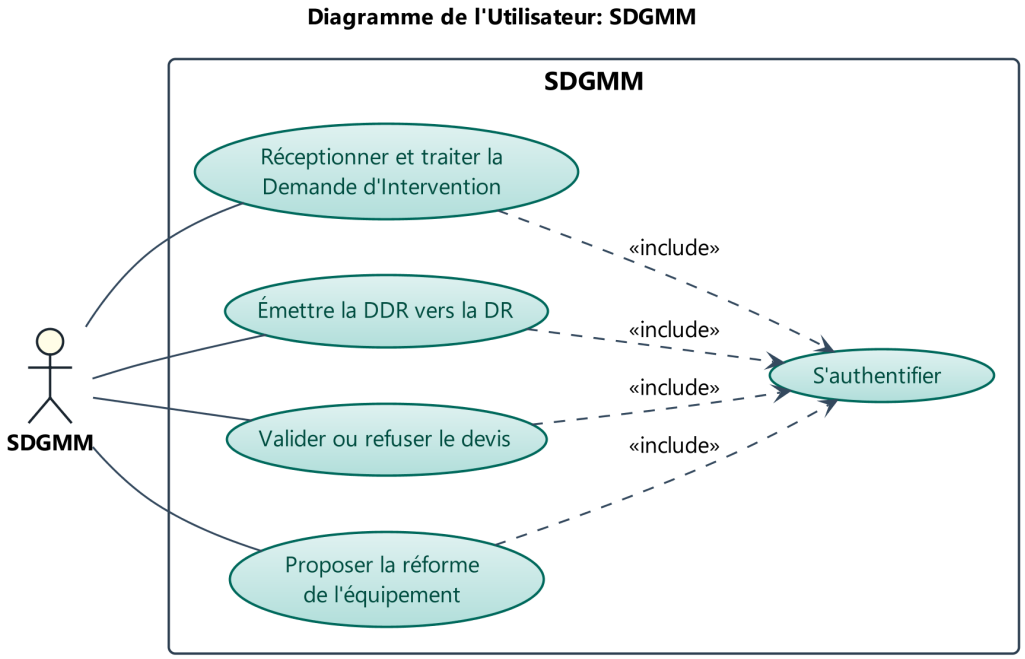


FIGURE 2.3: Cas d'utilisation : SDGMM

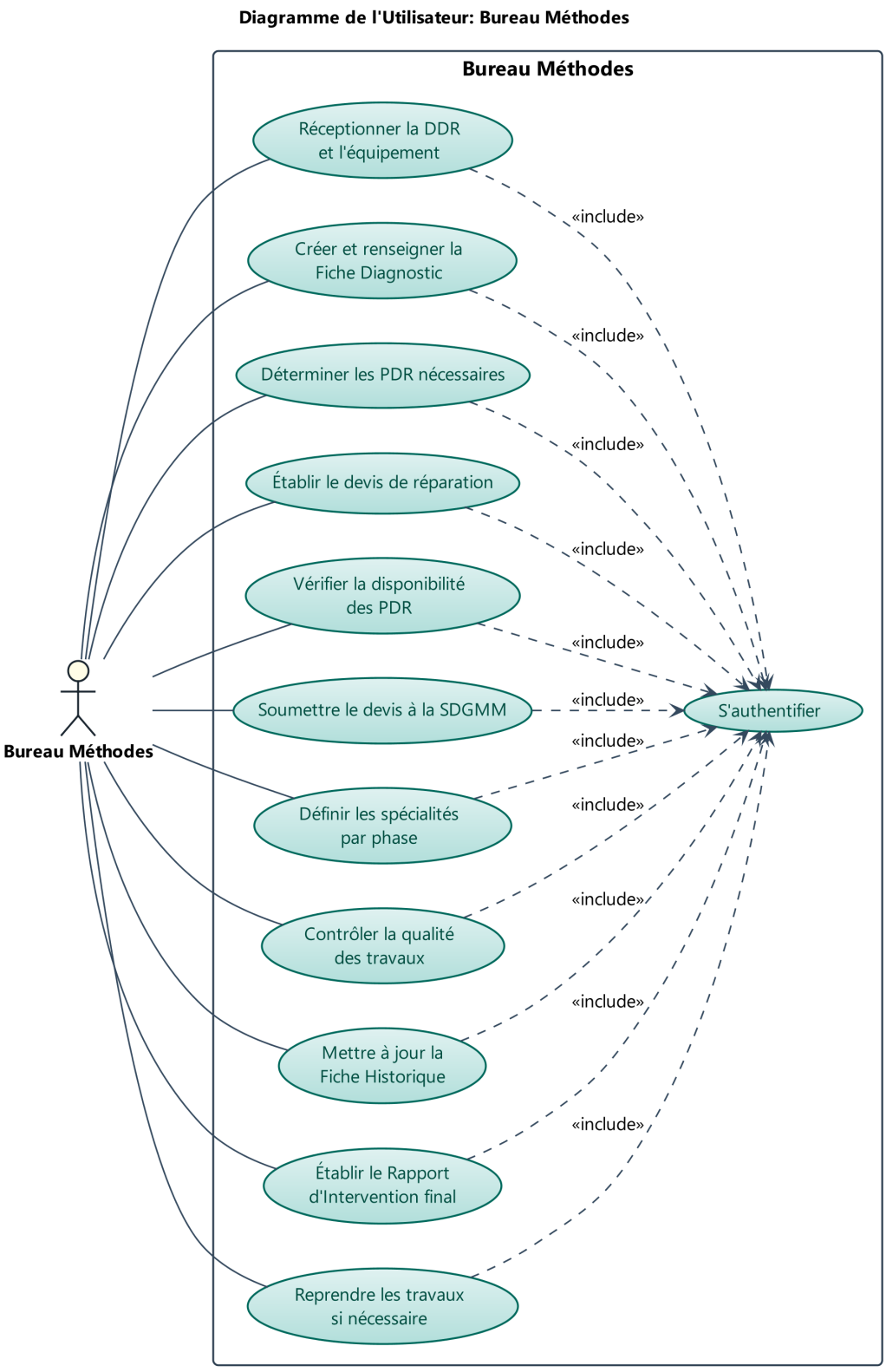


FIGURE 2.4: Cas d'utilisation : Bureau des Méthodes

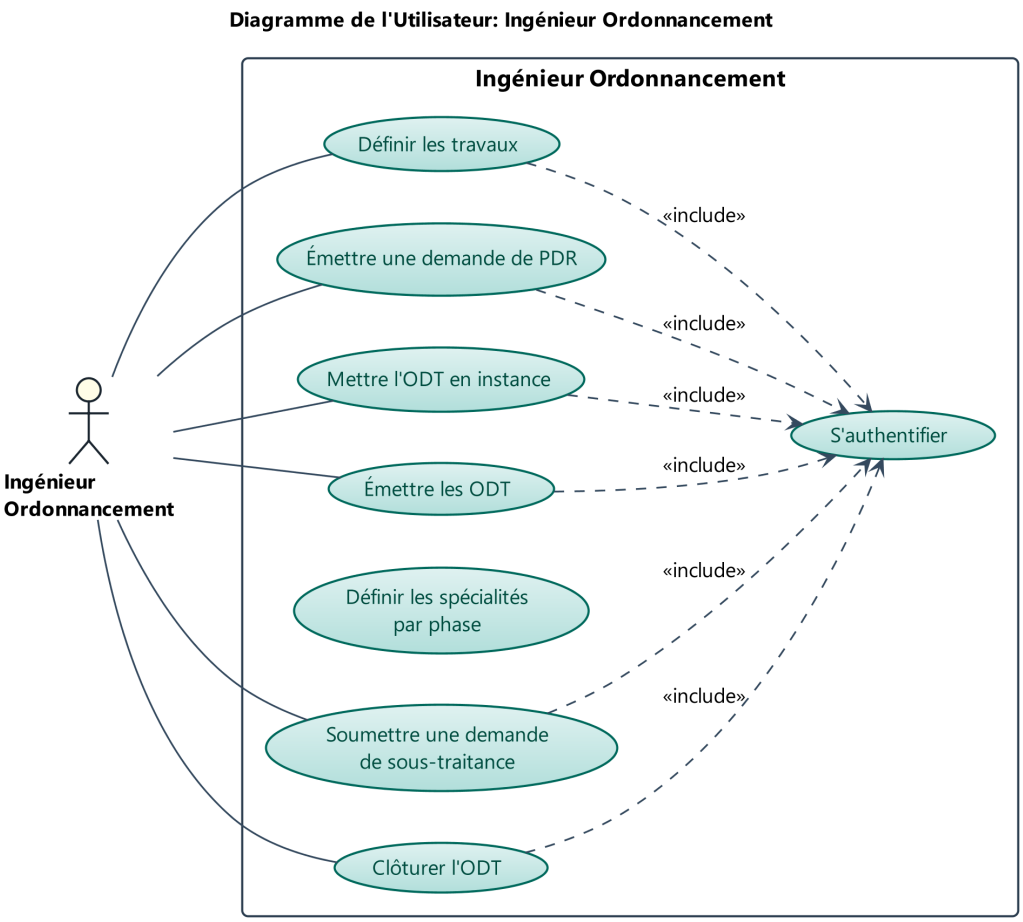


FIGURE 2.5: Cas d'utilisation : Ingénieur d'Ordonnancement

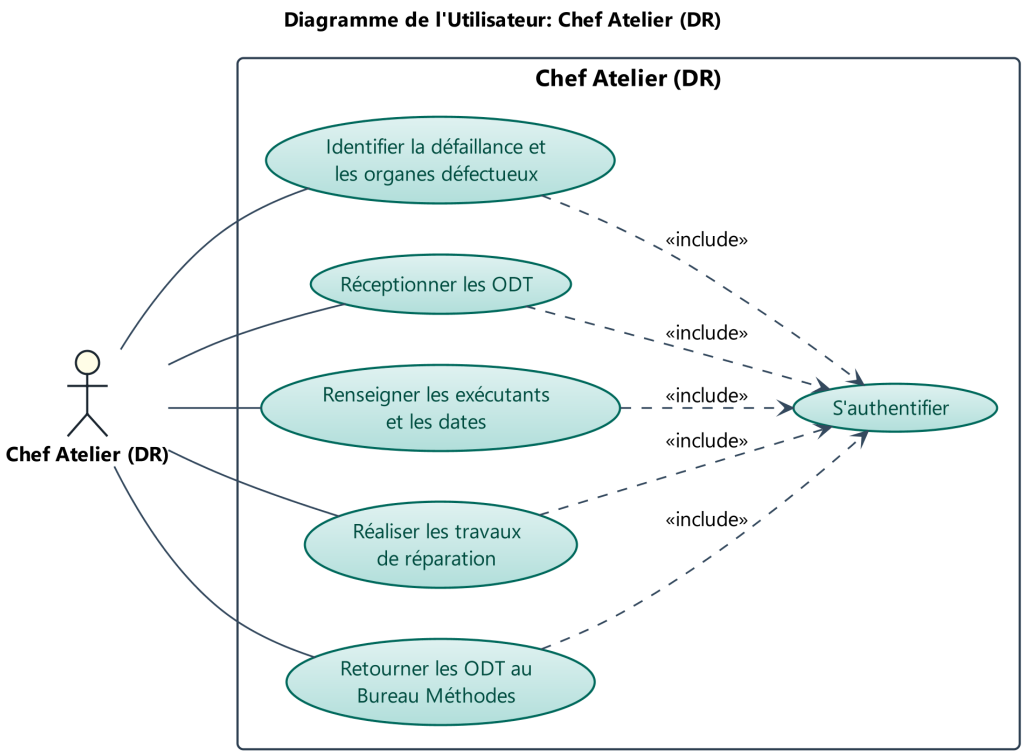


FIGURE 2.6: Cas d'utilisation : Chef d'Atelier

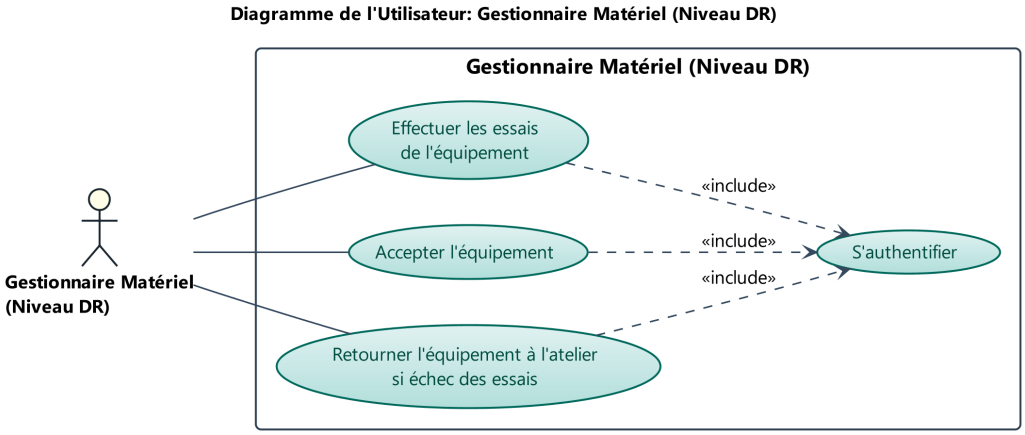


FIGURE 2.7: Cas d'utilisation : Gestionnaire Matériel (DR)

2.3.2 Diagrammes de Séquence

Sept diagrammes de séquence couvrent les phases du processus, en montrant les échanges chronologiques entre acteurs et système.

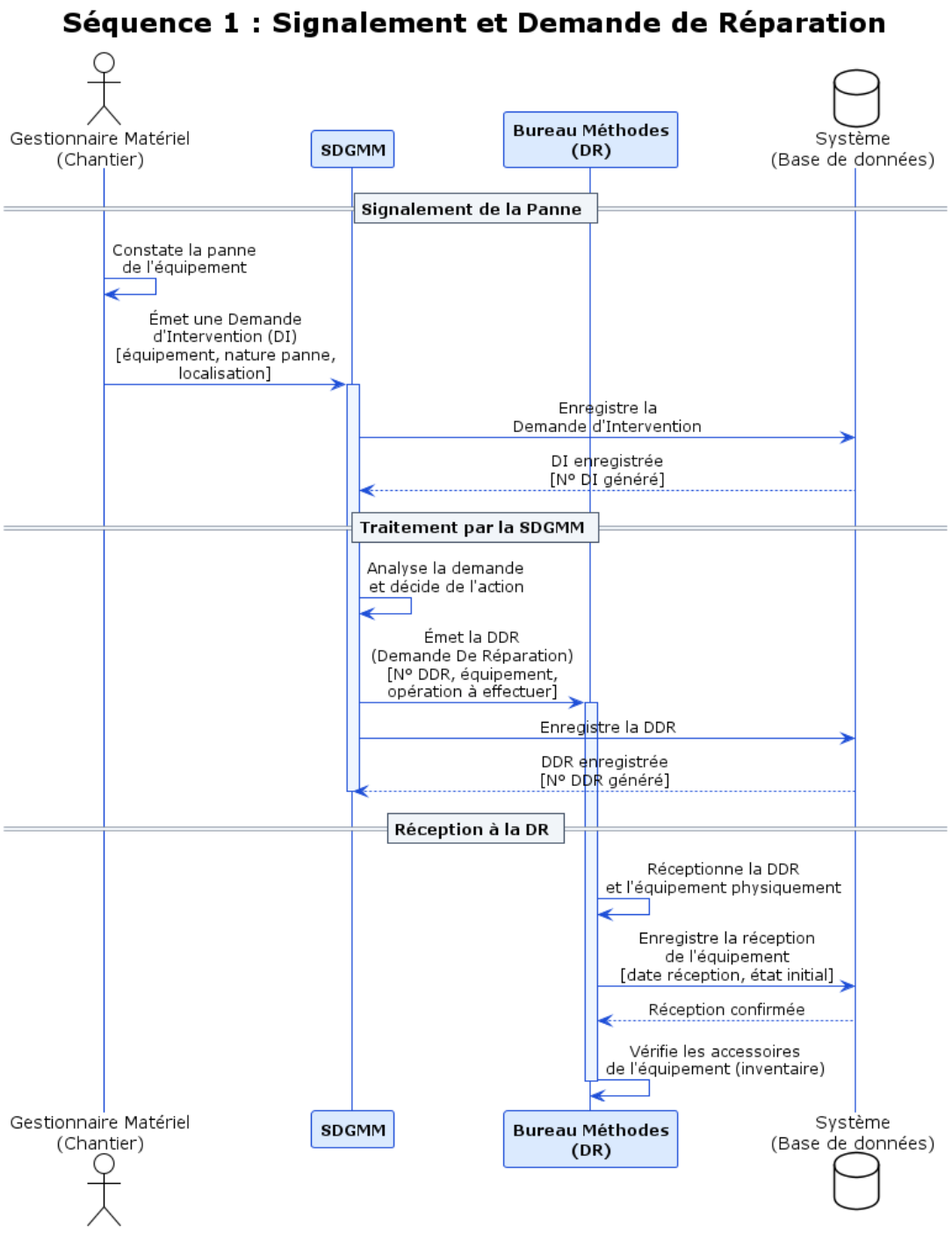


FIGURE 2.8: Séquence 1 : Signalement et Demande de Réparation

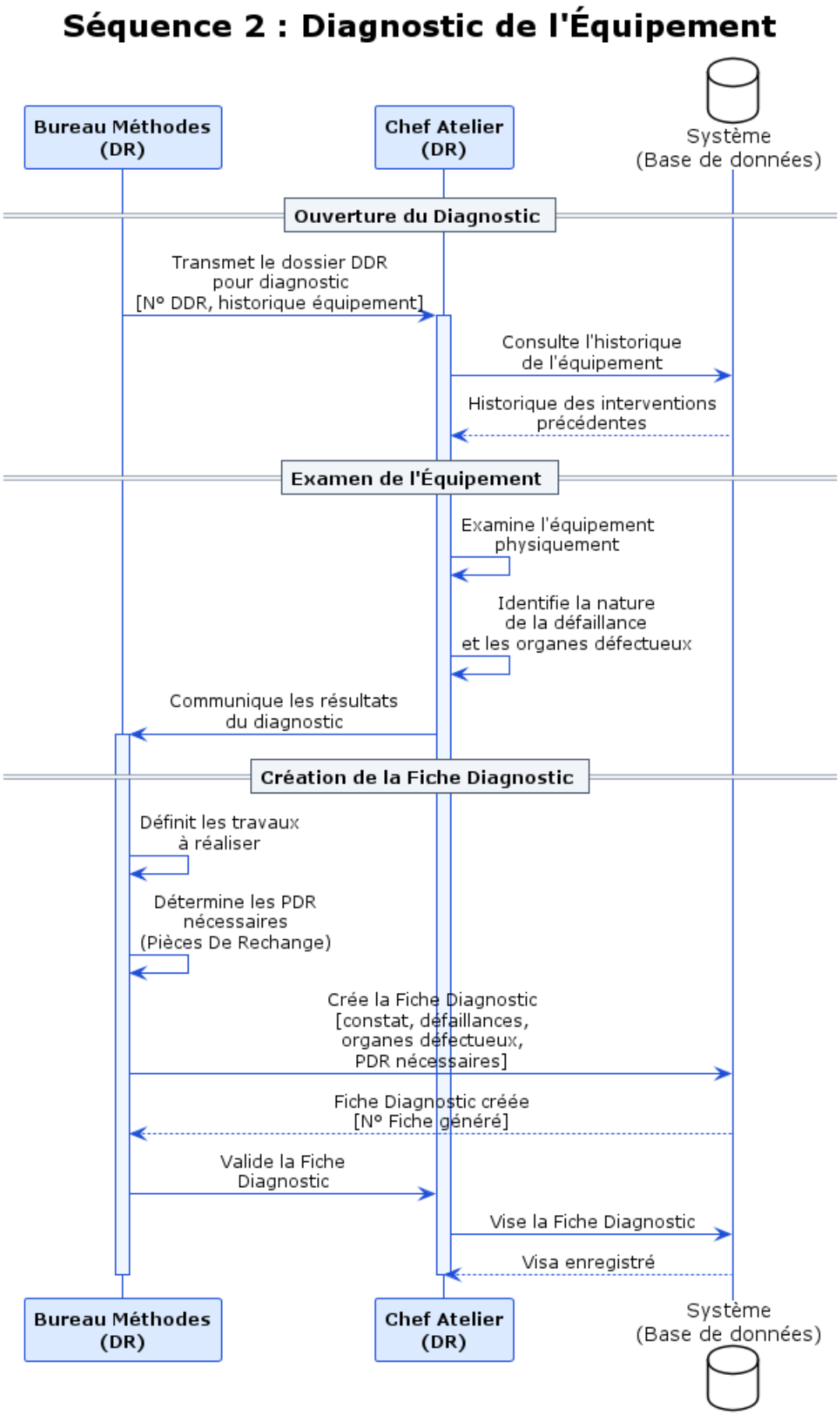


FIGURE 2.9: Séquence 2 : Diagnostic de l'Équipement

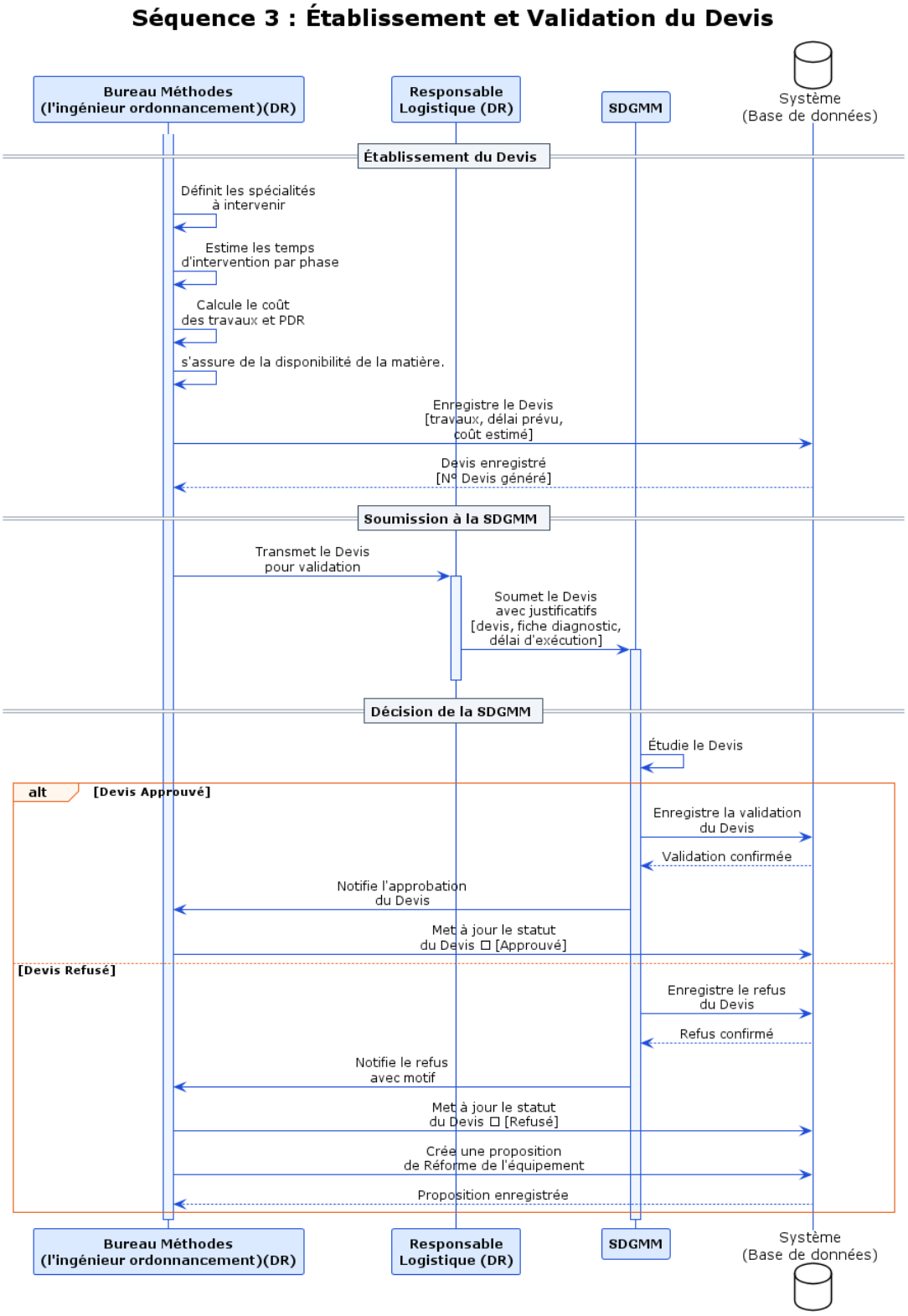


FIGURE 2.10: Séquence 3 : Établissement et Validation du Devis

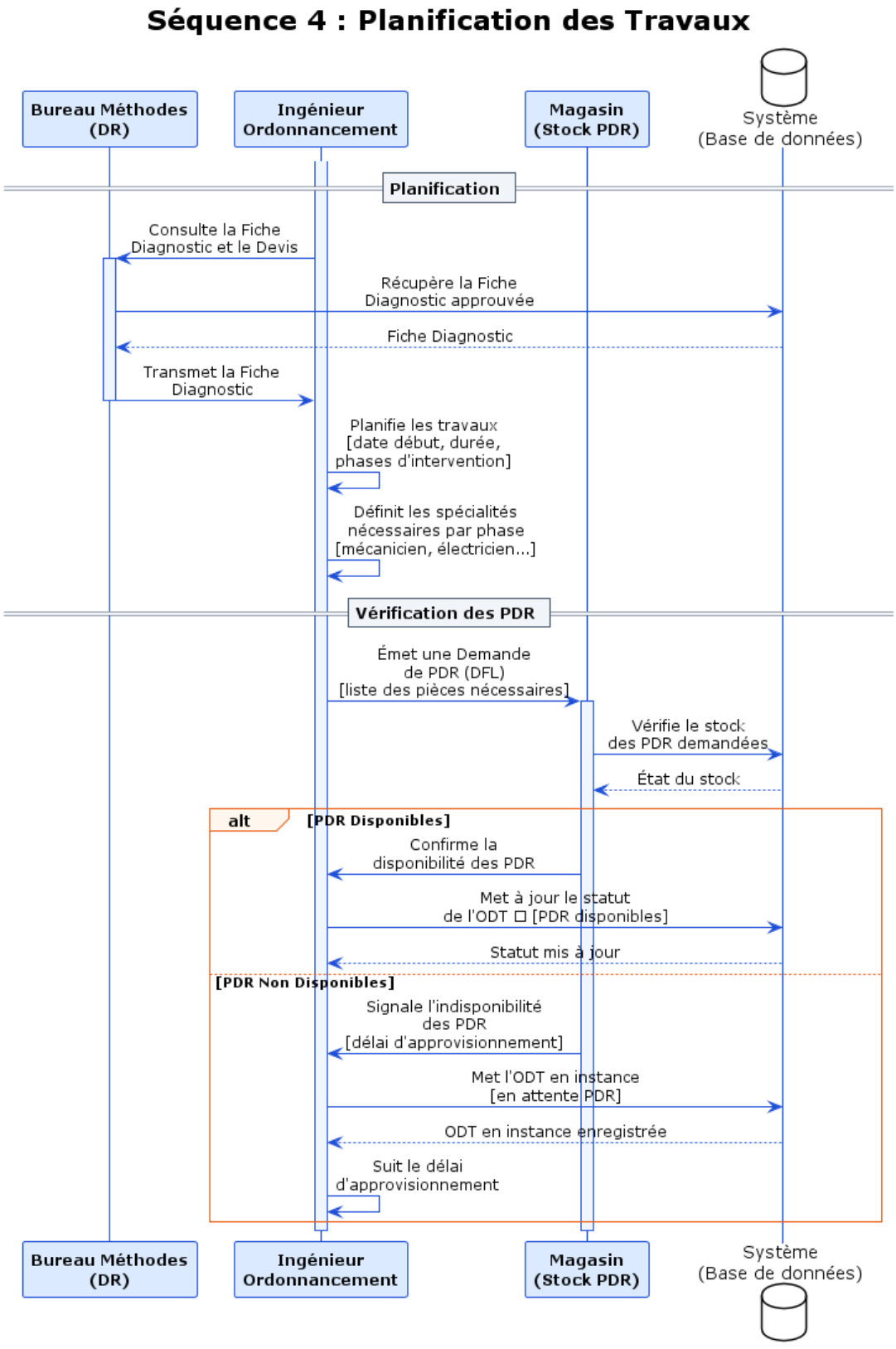


FIGURE 2.11: Séquence 4 : Planification des Travaux

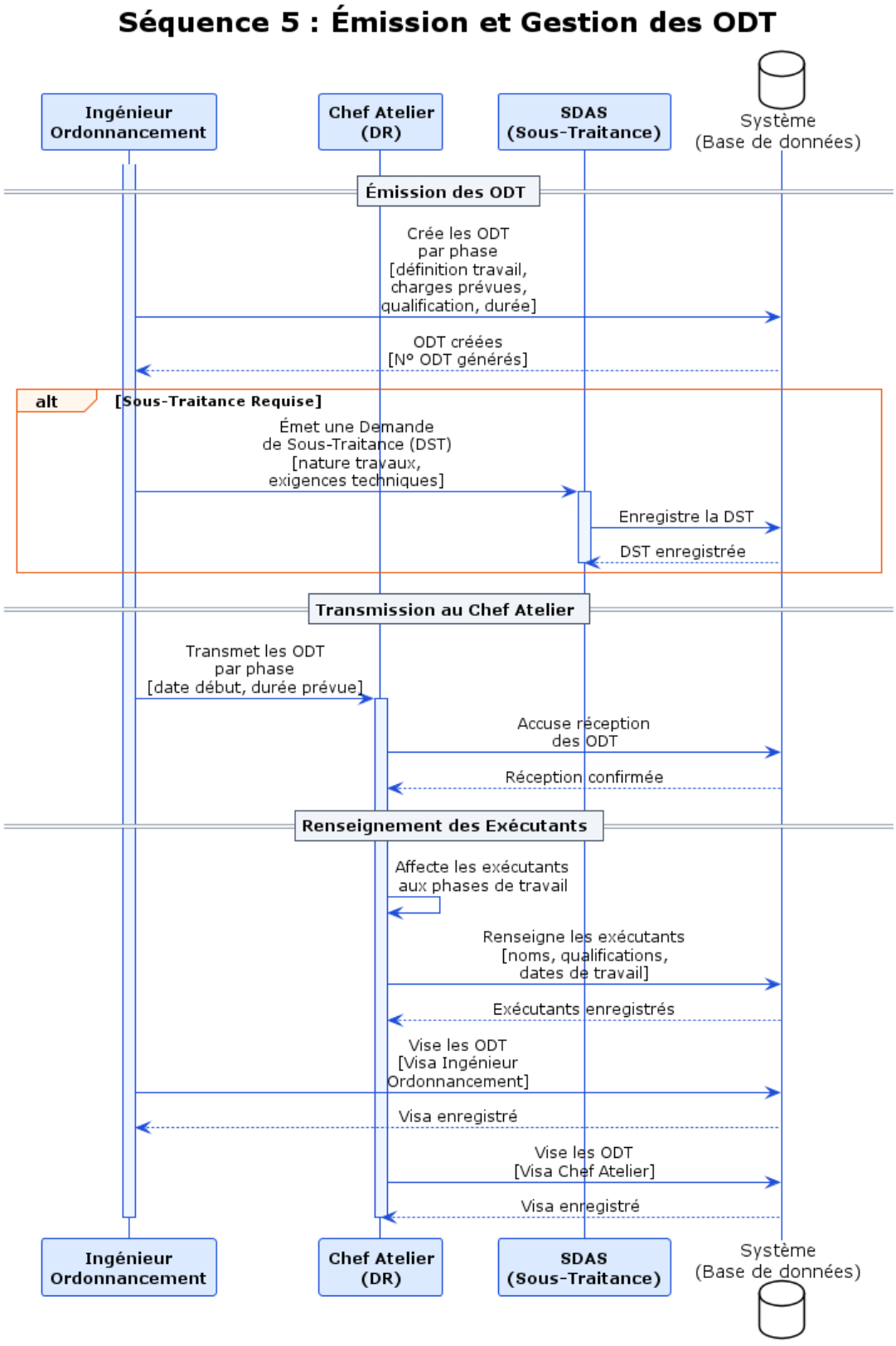


FIGURE 2.12: Séquence 5 : Émission et Gestion des ODT

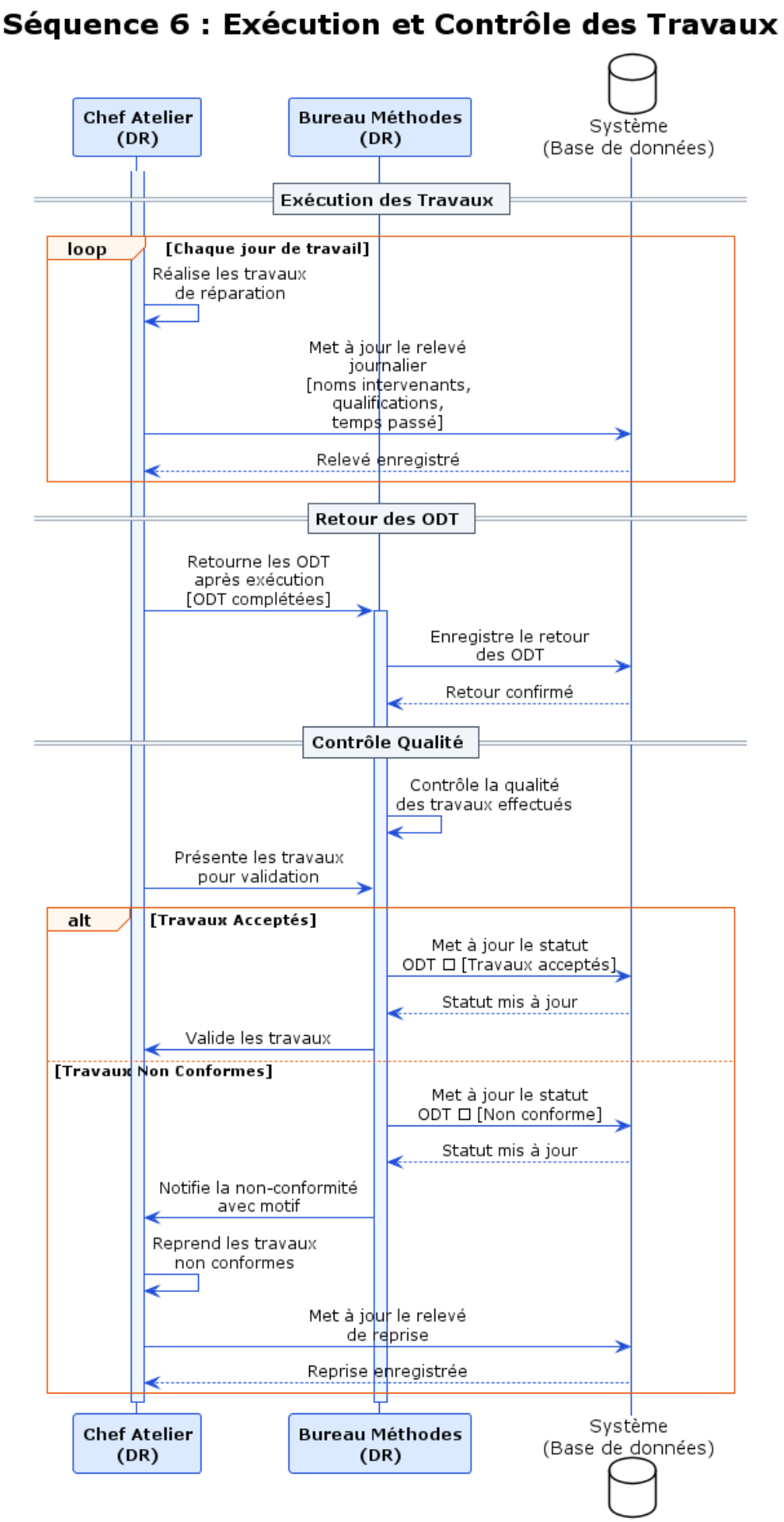


FIGURE 2.13: Séquence 6 : Exécution et Contrôle des Travaux

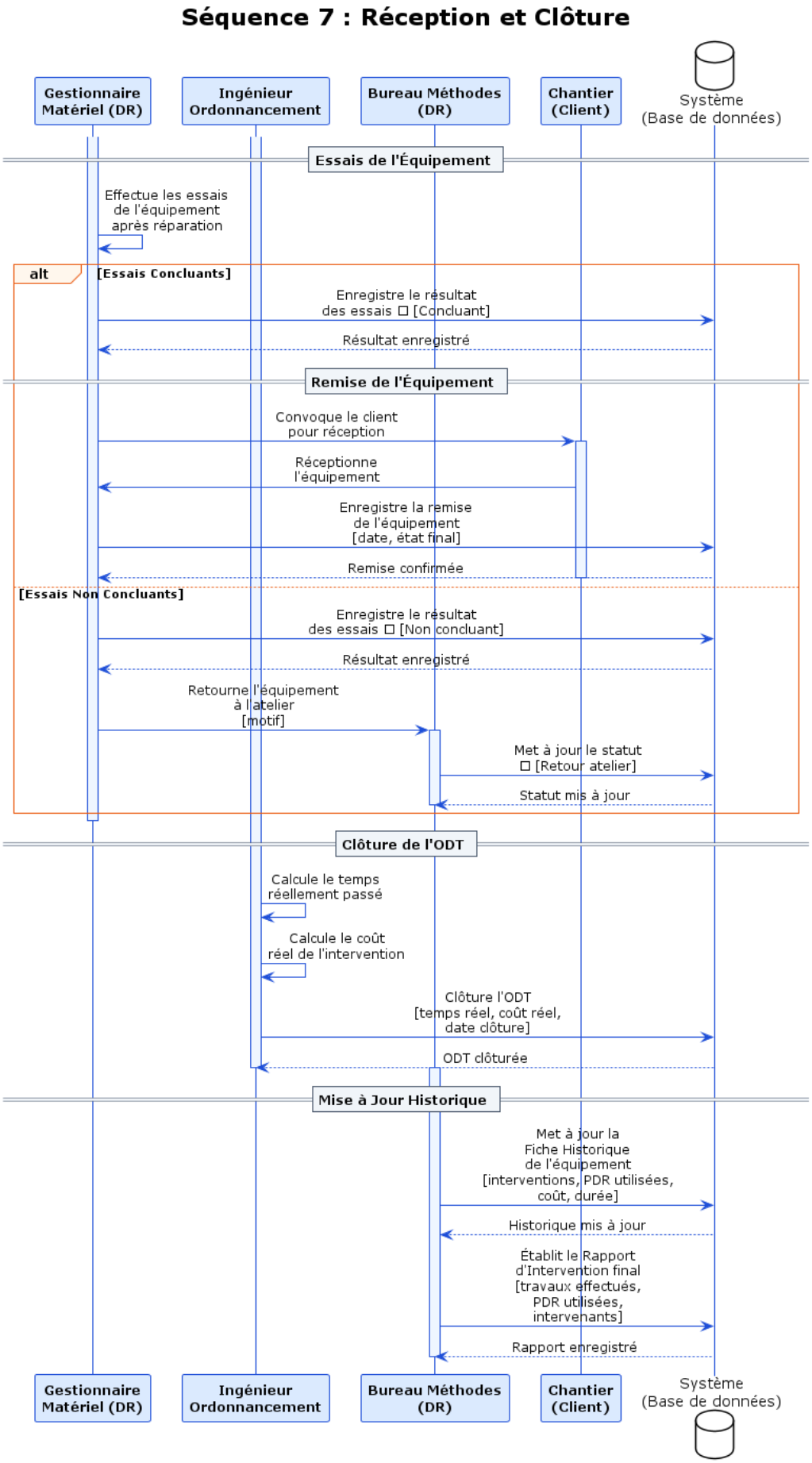


FIGURE 2.14: Séquence 7 : Réception et Clôture

2.3.3 Diagramme de Classes

Le diagramme de classes décrit la structure statique du système. Trois types de relations sont utilisés : l'association ($\rightarrow$), la composition ($*$) et la multiplicité (1, 0..*, 1..*, 0..1).

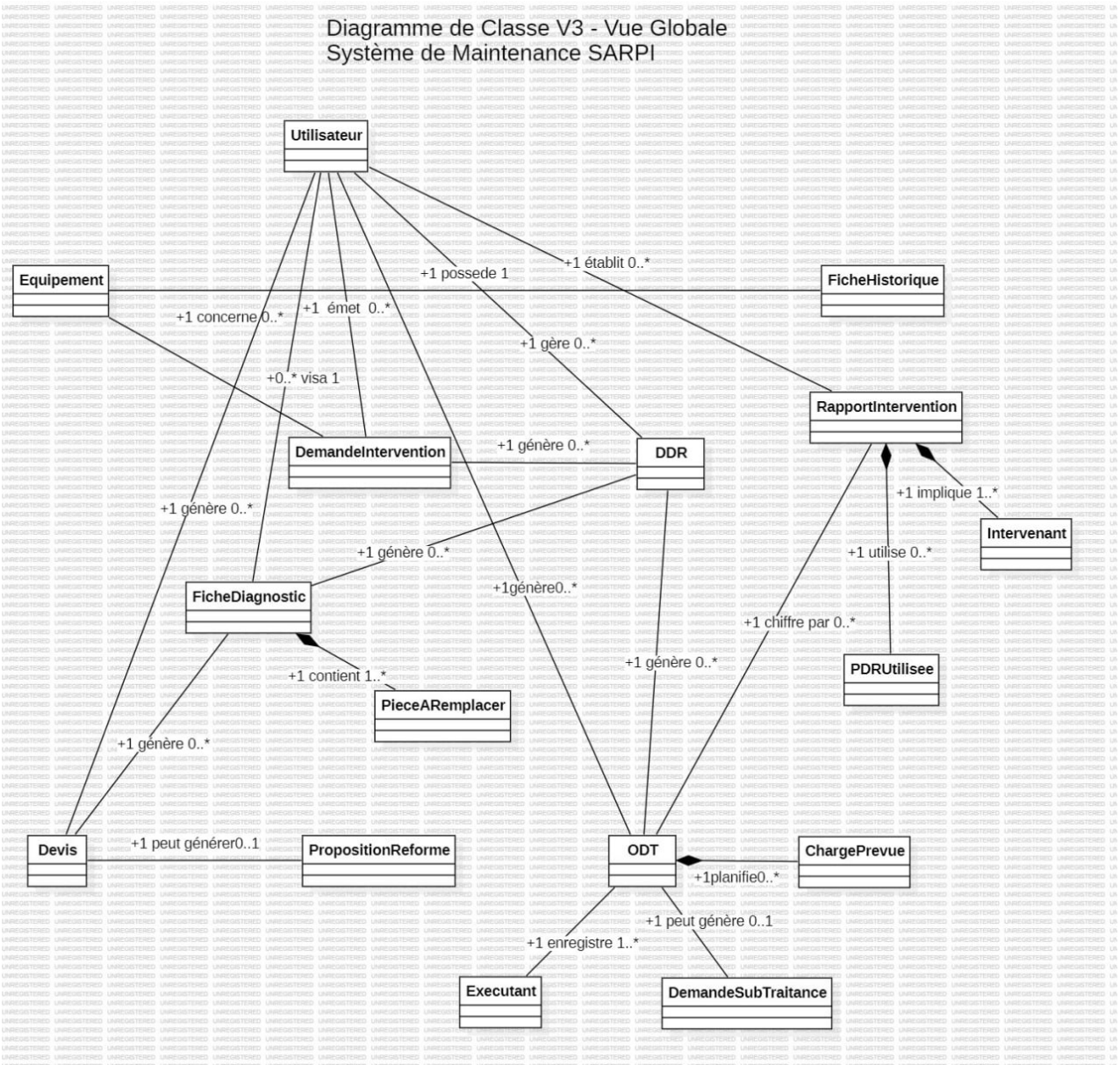


FIGURE 2.15: Diagramme de classes – Vue globale

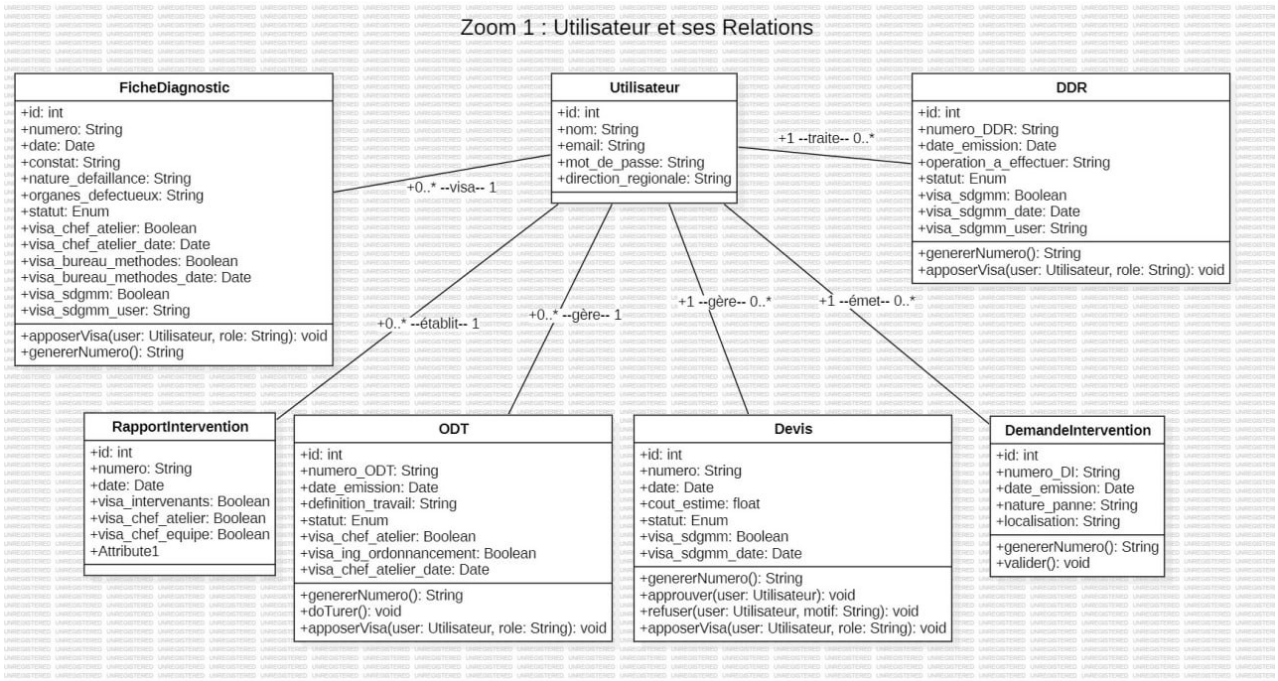


FIGURE 2.16: Zoom 1 : Utilisateur et ses Relations

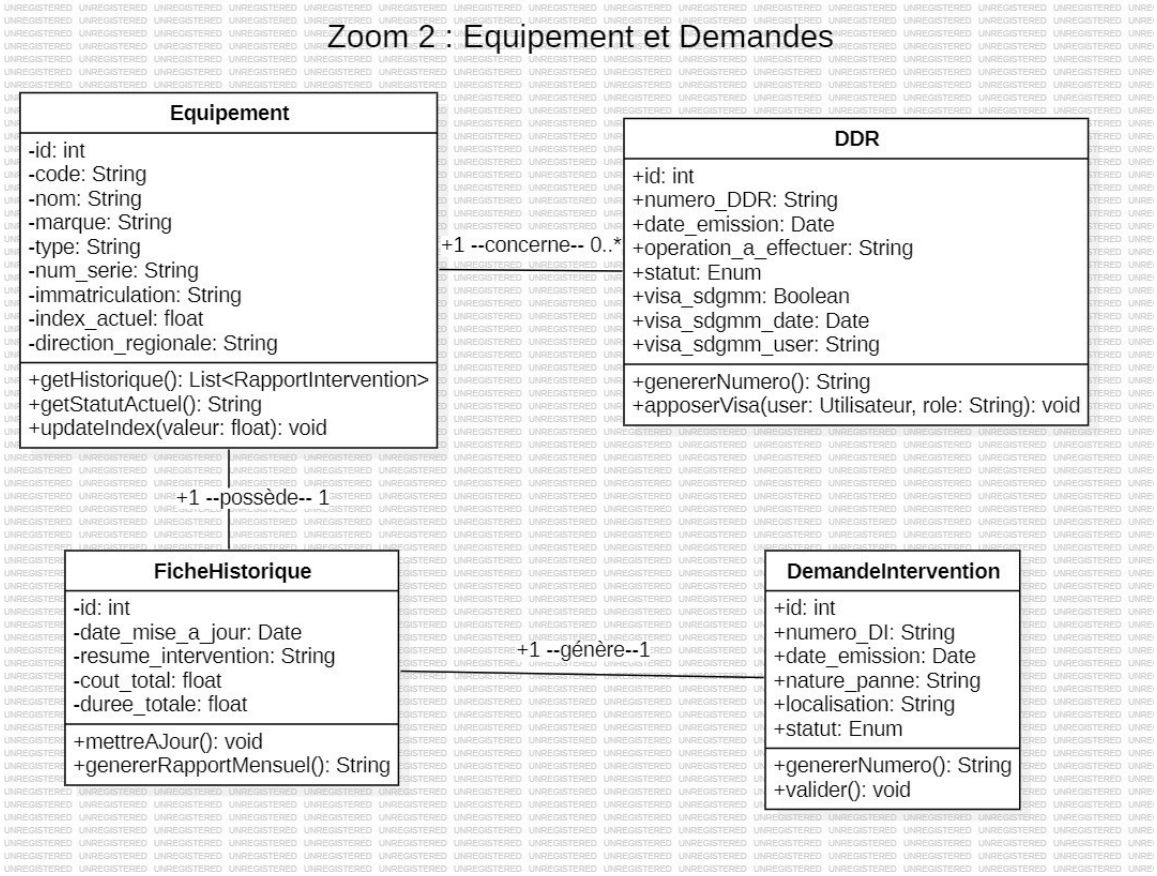


FIGURE 2.17: Zoom 2 : Équipement et Demandes

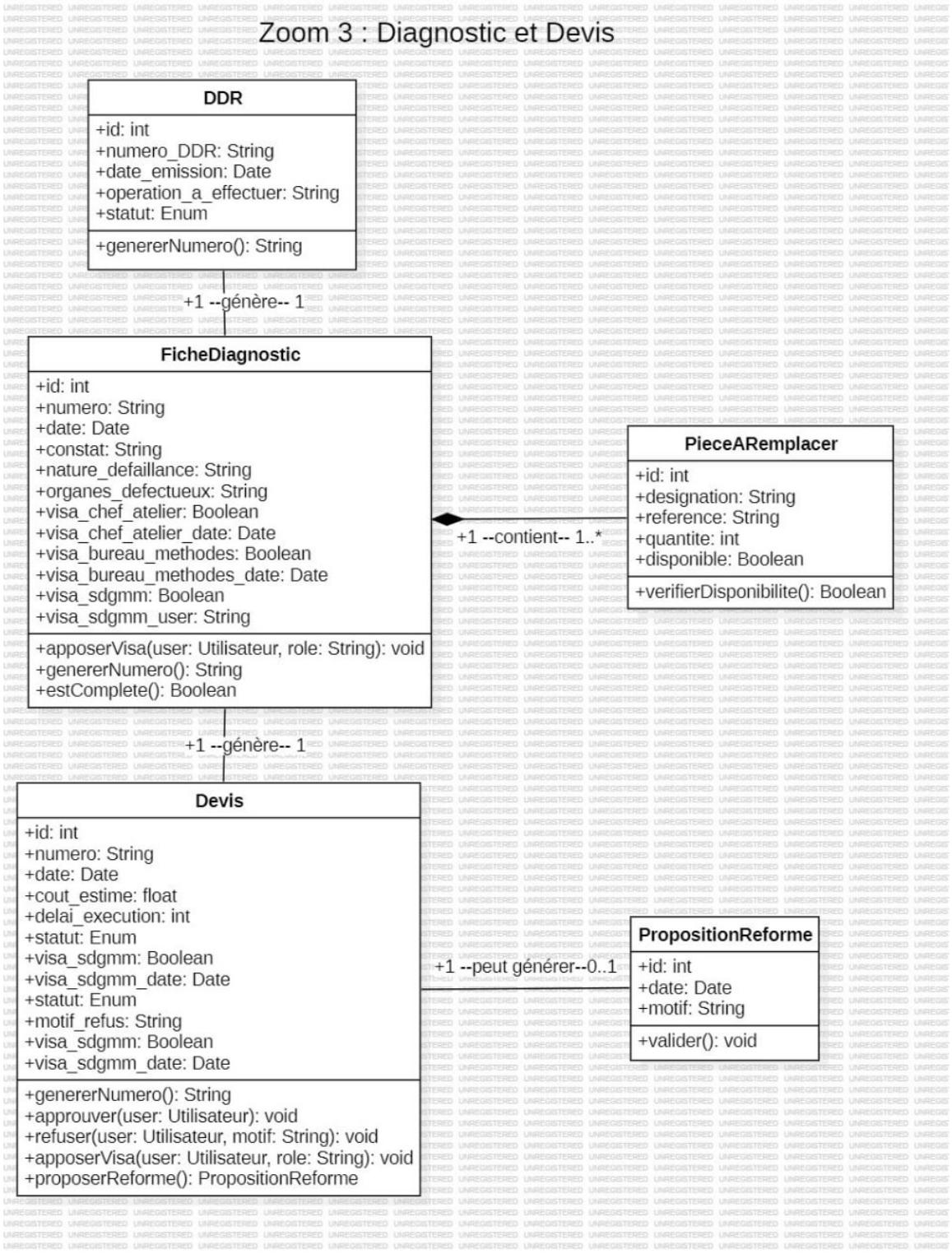


FIGURE 2.18: Zoom 3 : Diagnostic et Devis

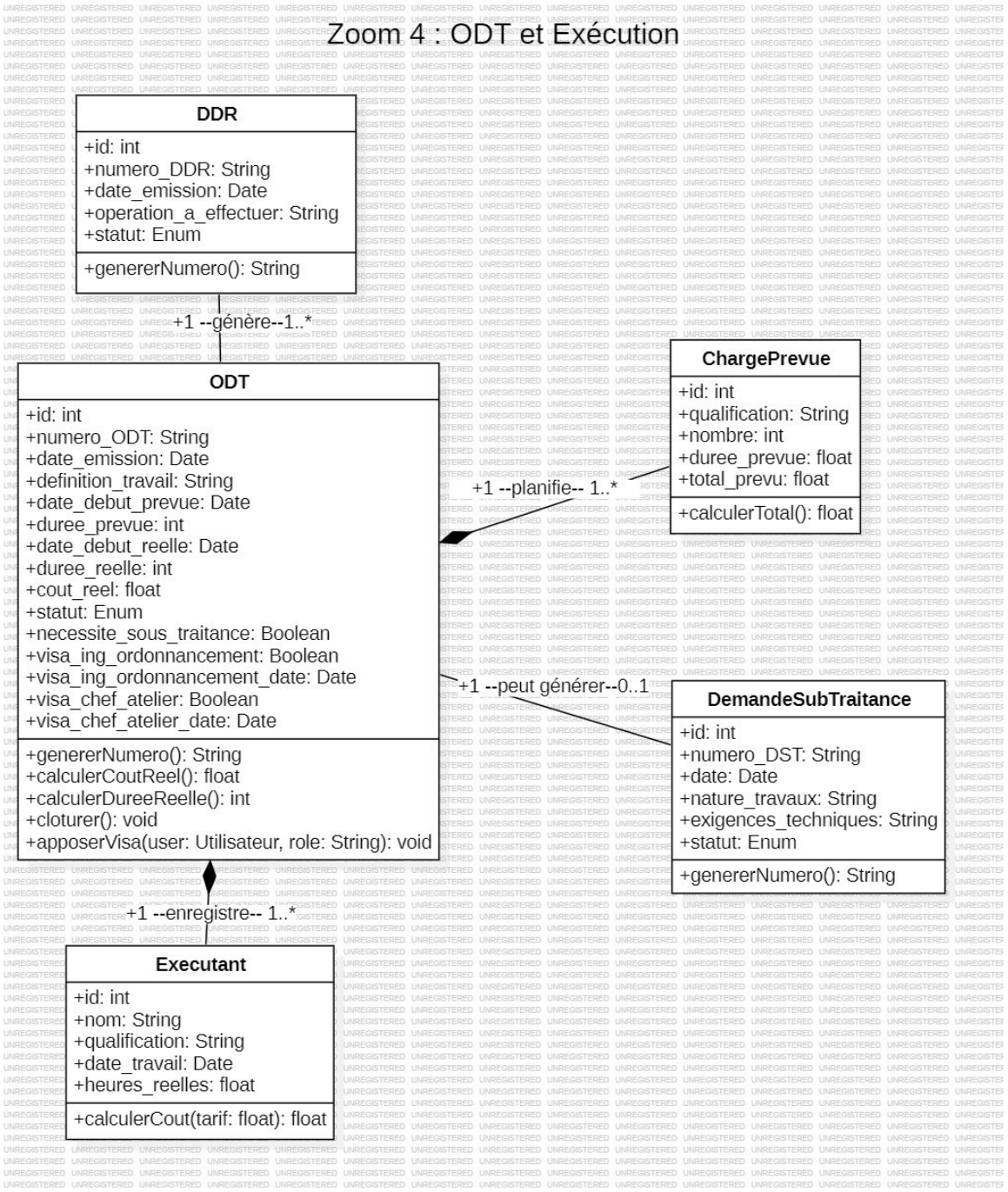


FIGURE 2.19: Zoom 4 : ODT et Exécution

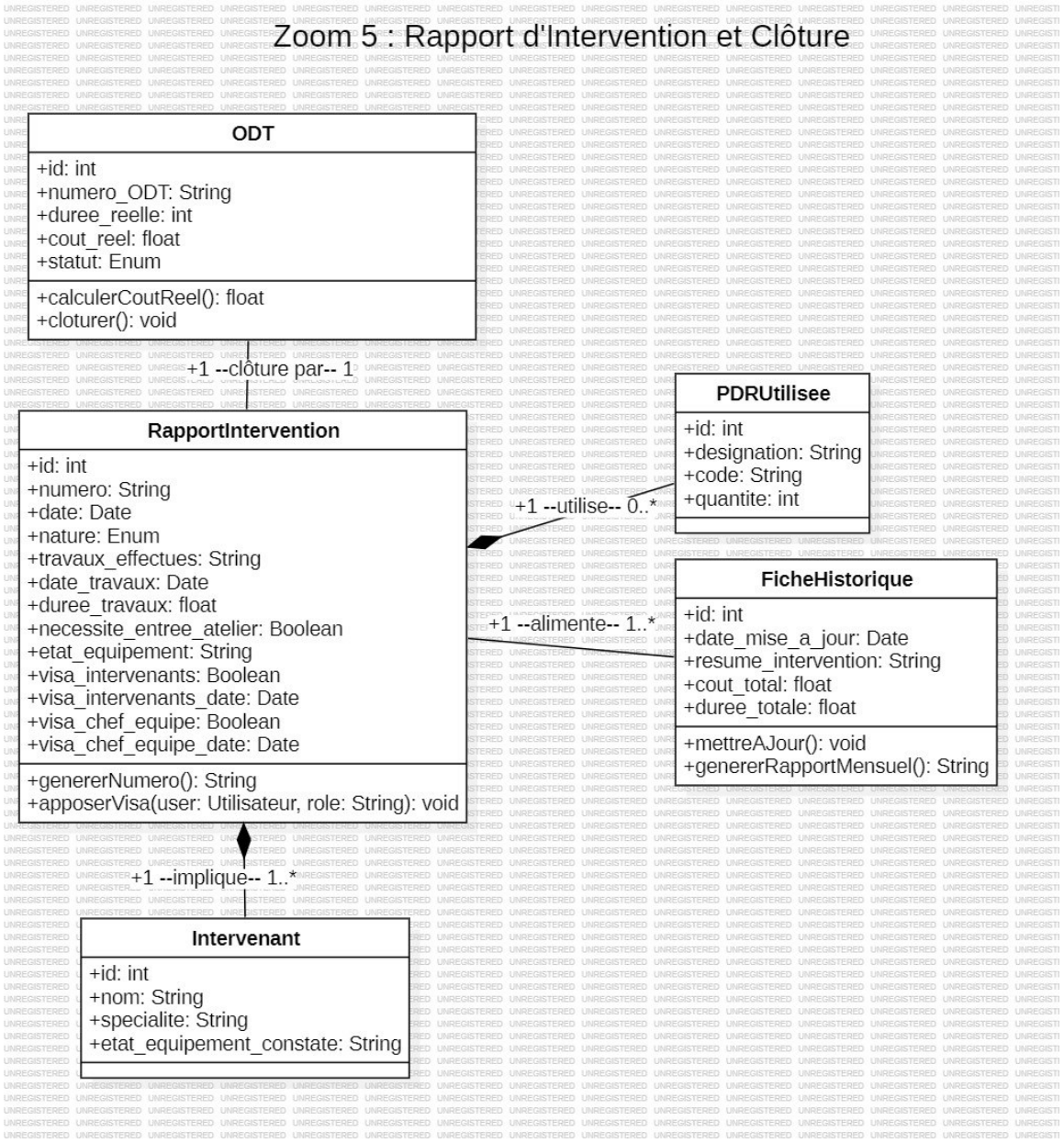


FIGURE 2.20: Zoom 5 : Rapport d'Intervention et Clôture

Le système comporte quatorze classes principales :

TABLE 2.3: Classes principales du système

Classe	**Rôle**
Utilisateur	Utilisateur du système avec son rôle et ses droits
Equipement	Engin du parc matériel SARPI
DemandeIntervention	Signalement de panne depuis le chantier
DDR	Demande De Réparation émise par la SDGMM
FicheDiagnostic	Résultats de l'examen technique
PieceARemplacer	PDR identifiées lors du diagnostic
Devis	Estimation du coût et des délais
PropositionReforme	Mise hors service en cas de devis refusé
ODT	Ordre De Travail par phase d'intervention
ChargePrevue	Ressources humaines planifiées par ODT
Executant	Relevés journaliers des intervenants
DemandeSubTraitance	Demande de sous-traitance
RapportIntervention	Bilan complet des travaux
FicheHistorique	Historique de toutes les interventions par équipement

Conclusion

Ce deuxième chapitre a défini ce que le système doit faire, qui l'utilise et comment il est structuré. Les diagrammes UML – cas d'utilisation, séquences et classes – constituent la base conceptuelle sur laquelle repose le développement présenté au chapitre suivant.

Chapitre 3

Réalisation et Mise en œuvre

Introduction

Ce chapitre présente les outils et les technologies utilisés pour développer l'application, ainsi que la structure technique du projet.

3.1 Environnement de Développement

3.1.1 Visual Studio Code

VS Code a été notre éditeur principal tout au long du projet. Son terminal intégré permet d'exécuter les commandes Artisan et Composer directement, et ses extensions PHP facilitent la navigation dans le code Laravel.

3.1.2 XAMPP

XAMPP fournit l'environnement local complet : Apache comme serveur HTTP, MySQL pour la base de données (`sarpi_maintenance`), PHP 8.2.12 comme langage serveur, et phpMyAdmin pour inspecter les tables.

3.1.3 PowerDesigner

PowerDesigner a servi à concevoir tous les diagrammes UML du chapitre précédent : cas d'utilisation, séquences et classes.

3.2 Technologies Utilisées

3.2.1 Backend

Laravel 12

Laravel est un framework PHP open source suivant le patron MVC. Nous avons utilisé la version 12.56.0 avec PHP 8.2.12, gérés via Composer 2.9.5. Il nous a apporté : Eloquent ORM (seize modèles PHP), Artisan (génération et exécution des vingt-huit migrations), Blade (templates HTML), et un système de routage centralisé couvrant 87 routes.

Spatie Laravel Permission

La gestion des rôles est assurée par **spatie/laravel-permission v6.25**. Six rôles ont été définis :

TABLE 3.1: Rôles définis dans le système

Rôle	**Acteur**
gestionnaire_chantier	Gestionnaire Matériel (Chantier)
sdgmm	Sous-Direction SDGMM
bureau_methodes	Bureau des Méthodes (DR)
chef_atelier	Chef d'Atelier (DR)
ingenieur_ordonnancement	Ingénieur d'Ordonnancement (DR)
gestionnaire_materiel_dr	Gestionnaire Matériel (DR)

L'authentification est gérée par **Laravel Breeze v2.4** (connexion, inscription, réinitialisation de mot de passe).

3.2.2 Frontend

— **Tailwind CSS v3.1 :** framework utilitaire pour le style des interfaces.
— **Alpine.js v3.4.2 :** gestion des comportements dynamiques côté client (menus, confirmations, affichages conditionnels).
— **Vite v7.0.7 :** compilation et optimisation des ressources JS et CSS, avec le plugin `laravel-vite-plugin`.

3.3 Structure de la Base de Données

La base `sarpi_maintenance` est décrite par vingt-huit migrations Laravel. Les tables principales :

TABLE 3.2: Principales tables de la base de données

Table	**Contenu**
users	Comptes utilisateurs avec rôles et DR
equipements	Parc matériel SARPI
demande_interventions	Signalements de pannes depuis les chantiers
ddrs	Demandes De Réparation
fiche_diagnostics	Résultats des examens techniques
piece_a_remplacers	PDR identifiées
devis	Devis avec visas électroniques
proposition_reformes	Propositions de mise hors service
odts	Ordres De Travail avec visas
charge_prevues	Charges planifiées par qualification
executants	Relevés journaliers des intervenants
demande_sub_traitances	Demandes de sous-traitance
rapport_interventions	Bilans finaux des travaux
fiche_historiques	Historique des interventions par équipement

Conclusion

L'application a été développée avec Laravel 12, Tailwind CSS et Alpine.js, hébergée sous XAMPP et modélisée avec PowerDesigner. Elle couvre l'intégralité du processus de maintenance corrective de SARPI, du signalement de la panne à la mise à jour de la Fiche Historique, avec des droits d'accès stricts pour chaque acteur.

Bibliographie

[1] SARPI – Société Algérienne de Réalisation de Pipelines et d'Infrastructures. *Site officiel de SARPI*. Disponible sur : `https://sarpi.dz/` [Consulté en 2026].

[2] SARPI. *Procédure interne PRO-SMI-PS2-03 – Maintenance*, révision 05. Document interne, mars 2025.