j

**REPUBLIQUE ISLAMIQUE MAURITANIE**

**MINISTERE DES MINES ET DE L’INDUSTRIE**



**DOCUMENT OFFICIEL DE CADRAGE ET DE CONCEPTION.**

**Développement d’une Plateforme Numérique pour la Gestion des Demandes d’Autorisation Industrielle**

**Réf : DGI TDR 002/2025**



PARTNERS F 0R DE VEL0PMENT

1. **INTRODUCTION**

La Direction Générale de l’Industrie (DGI) du Ministère des Mines et de l’Industrie souhaite moderniser et numériser la gestion des demandes d’autorisations industrielles à travers une plateforme centralisée, intuitive et sécurisée. Ce document présente le cadrage, la conception fonctionnelle et technique, ainsi que la modélisation des processus, conformément aux exigences des TDR.

**2. CONTEXTE ET OBJECTIFS**

**2.1 Contexte réglementaire**

Décret n°2009-189 (enregistrement des entreprises industrielles)

Décret n°2012-037 (eaux minérales)

Arrêtés n°0185 et n°0333 (boulangeries-pâtisseries)

**2.2 Objectifs du projet**

Dématérialiser le dépôt, l’analyse, la validation et le suivi des demandes.

Accélérer les délais de traitement et renforcer la transparence.

Garantir un archivage numérique sécurisé et un pilotage efficace via des tableaux de bord.

**CHAPITRE I : PORTEE DU PROJET**

**Introduction**

La définition de la portée du projet constitue une étape essentielle pour cadrer le développement de la plateforme numérique centralisée destinée à la gestion des autorisations industrielles par la DGI. Elle permet de clarifier les objectifs visés, de délimiter le périmètre fonctionnel et technique et d’identifier les exclusions, contraintes, ressources et livrables attendus. Ce travail de cadrage facilite l’alignement des parties prenantes, la planification des différentes phases du projet et la mise en place d’un suivi rigoureux, garantissant ainsi la réussite du déploiement dans les délais impartis.

**1- Objectifs de la du Projet**

La portée du projet consiste en la création d'une plateforme numérique centralisée pour la DGI permettant de :

* Dématérialiser le traitement des autorisations industrielles (boulangerie-pâtisserie, eau minérale, usines)
* Optimiser les délais de traitement des demandes
* Assurer la traçabilité complète du processus décisionnel
* Fournir un archivage numérique sécurisé
* Générer des tableaux de bord statistiques et analytiques
* Renforcer la transparence administrative.

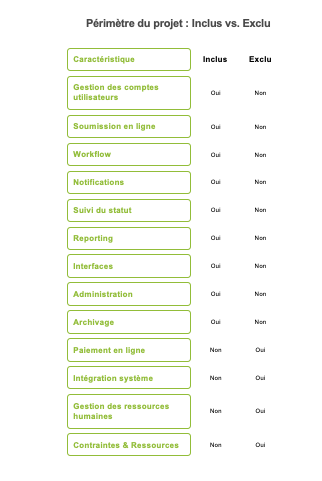
**2. Périmètre Fonctionnel**

**. Inclus dans la portée :**

* Gestion des comptes utilisateurs multi-profils (DEMANDEUR, SG, DGI, DDPI, etc.)
* Dépôt de demandes en ligne avec télé versement des pièces justificatives
* Workflow paramétrable selon le type de demande et conformément aux réglementations
* Notifications et alertes automatiques
* Suivi en temps réel du statut des demandes
* Tableaux de bord et rapports statistiques
* Interfaces web et mobile sécurisées
* Module d'administration et de gestion des droits
* Archivage électronique sécurisé, avec sauvegardes automatiques

**. Exclu du projet**

* Paiement en ligne des frais de dossier
* Intégration avec d’autres systèmes externes non mentionnés
* Gestion des ressources humaines internes du ministère
* Identifier les contraintes et ressources

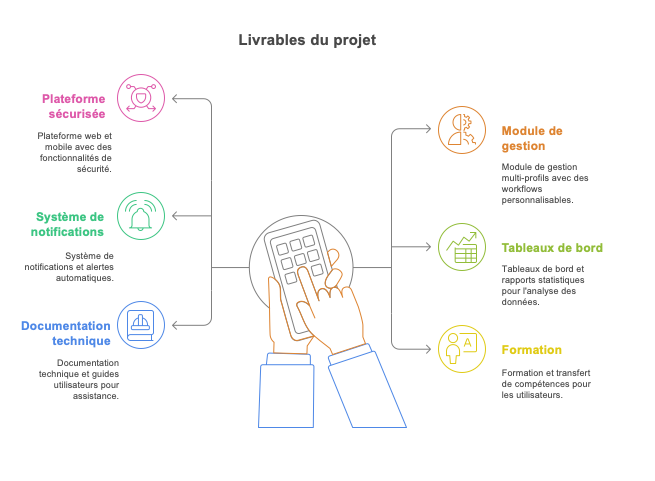


**3. Respect des textes réglementaires (Décrets et Arrêtés cités)**

* Délai de réalisation : 5 mois maximum
* Budget et ressources humaines définis par Richat-Partners
* Disponibilité des parties prenantes pour validation et tests

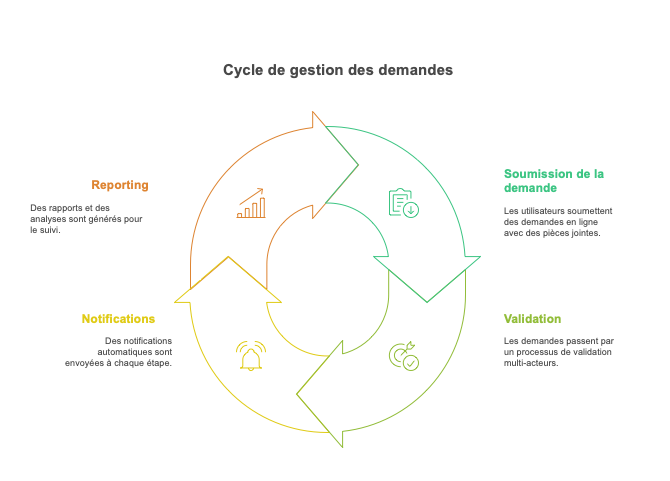
**Décrire les livrables attendus**

* Plateforme web et mobile sécurisée
* Module de gestion multi-profils et workflows paramétrables
* Système de notifications et alertes automatiques
* Tableaux de bord et rapports statistiques
* Documentation technique et guides utilisateurs
* Formation et transfert de compétences



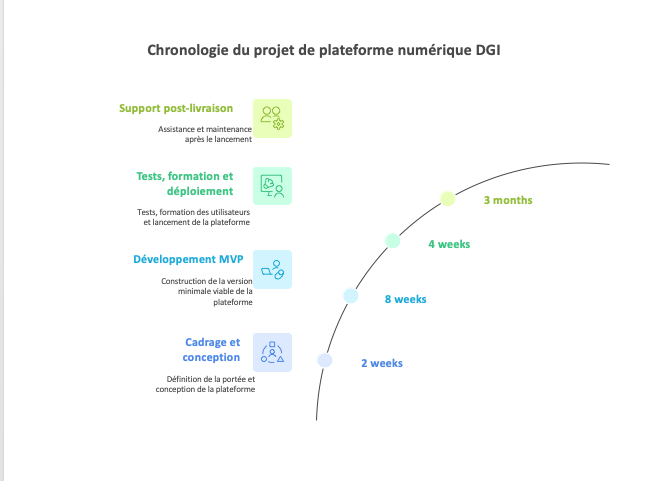
**4. Délimiter les fonctionnalités et travaux inclus**

* Gestion des utilisateurs et rôles (demandeurs, agents, administrateurs)
* Dépôt et suivi des demandes en ligne avec pièces jointes
* Circuit de validation multi-acteurs selon type de demande
* Notifications automatiques à chaque étape
* Reporting et pilotage



**5. Établir le calendrier et jalons clés**

* Phase de cadrage et conception : 2 semaines
* Développement MVP : 8 semaines
* Tests, formation et déploiement : 4 semaines
* Support post-livraison : 3 mois



**6. Valider et communiquer la portée**

* Soumettre la portée formalisée à la DGI et aux parties prenantes pour validation
* Diffuser un document clair et partagé servant de référence tout au long du projet
* Mettre en place un processus de gestion des changements pour maîtriser les
* évolutions

**CHAPITRE II : ANALYSE DES BESOINS.**

**1. Objectif du projet**

Développer une plateforme numérique centralisée, intuitive et sécurisée, permettant la gestion dématérialisée des demandes d’autorisations industrielles auprès de la Direction Générale de l’Industrie (DGI).  
   
 Cette plateforme vise à :  
 • Dématérialiser le dépôt, l’analyse, la validation et le suivi des demandes ;  
 • Améliorer les délais de traitement et la transparence administrative ;  
 • Assurer un archivage numérique sécurisé ;  
 • Fournir des tableaux de bord pour le pilotage statistique et stratégique.

**2. Utilisateurs du système**

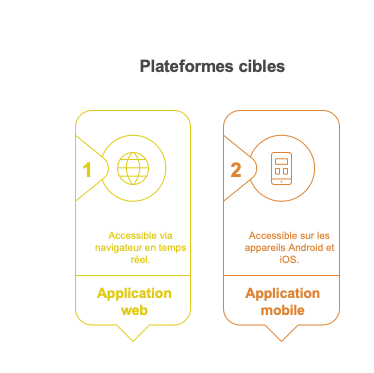
|  |  |
| --- | --- |
| **Type d&apos;utilisateur** | **Rôle principal** |
| **Demandeur** | Crée un compte, soumet une demande, suit son avancement |
| **Administrateurs DGI** | Reçoivent, vérifient et traitent les demandes selon leur service |
| **Superviseurs / décideurs** | Suivi, validation, génération de rapports |

**3. Fonctionnalités principales**

|  |  |
| --- | --- |
| **c** | **Description** |
| **Création de comptes sécurisés** | Chaque entreprise industrielle a un compte personnel |
| **Soumission de demande en ligne** | Formulaire numérique avec pièces jointes |
| **Suivi de l’état d’avancement** | Consultation en temps réel de l’avancement du dossier |
| **Traitement multi-niveaux** | Répartition entre services (SG, DGI, DDPI, etc.) |
| **Workflow dynamique** | Enchaînement automatique des étapes selon le type de demande |
| **Alertes et notifications** | Emails ou messages automatiques selon l’évolution du dossier |
| **Tableau de bord** | Statistiques, suivi des délais, types de demandes |
| **Archivage numérique** | Historique sécurisé de toutes les actions et documents |
| **Gestion des accès** | Permissions selon les profils utilisateurs |
| **Reporting** | Génération de rapports complets, exportables |

**4. Plateformes cibles**

• Application web (navigateur)  
 • Application mobile (Android/iOS)  
   
 Accessible en temps réel via Internet, avec authentification forte.

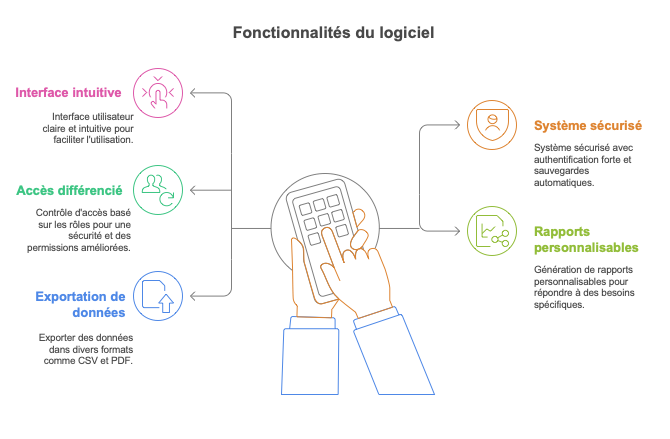


**5. Contraintes et exigences spécifiques**

• Respect des textes réglementaires en vigueur (décrets et arrêtés de 2009, 2012, 2023)  
 • Traçabilité complète de toutes les actions  
 • Conservation des données et historique  
 • Mise à disposition du code source complet au ministère à la fin  
 • Pas de dépendance à des logiciels fermés ou propriétaires  
 • Formation et accompagnement du personnel  
 • Assistance post-livraison (au moins 3 mois)

**6. Éléments techniques attendus**

• Interface utilisateur claire et intuitive  
 • Système sécurisé (authentification forte, sauvegardes automatiques)  
 • Accès différencié selon les rôles  
 • Génération de rapports personnalisables  
 • Export de données (CSV, PDF…)



**7. Technologies choisies**

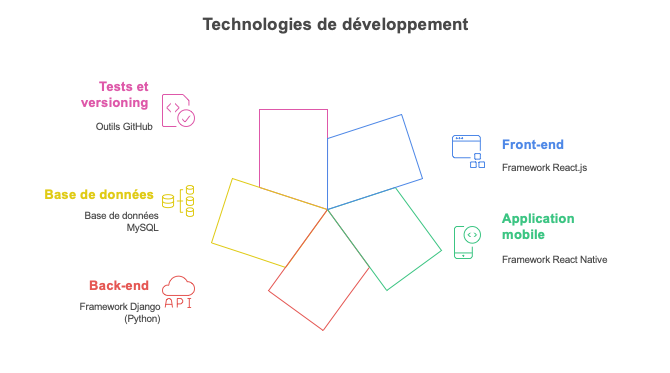
• Front-end web : React.js

• Application mobile : React Native

• Back-end (API & logique métier) : Django (Python)

• Base de données : MySQL

• Outils de test et de versioning : GitHub (tests, CI/CD, gestion de versions)



 Profils fonctionnels (côté utilisateur plateforme)

|  |  |
| --- | --- |
| Profil | Description & Accès |
| Demandeur | Entreprises ou personnes physiques déposant une demande. Accès à la création, suivi, téléchargement. |
| Secrétariat Central | Agents d’accueil, accusent réception et transmettent. |
| Secrétariat Général | Décident du circuit, orientent les dossiers. |
| DGI | Supervise, transmet aux directions, valide et archive. |
| DDPI | Instruisent techniquement certains types de dossiers. |
| Chef Service | Vérifient, préparent, étudient les dossiers. |
| Comité Technique | Interviennent pour validation technique (ex : distance entre boulangeries). |
| Commission Multisectorielle | Valident les dossiers nécessitant une instruction multisectorielle. |
| DRMNE | Traitent spécifiquement les demandes d’adhésion PMNE. |
| Ministre / MMI | Décident, signent les autorisations. |
| Administrateur | Supervise, gère les droits, paramètre le système. |

**Profils techniques (pour la réalisation du projet)**

|  |  |
| --- | --- |
| Profil | Missions principales |
| Chef de projet | Coordination, suivi, reporting, communication client |
| Architecte logiciel | Conception de l’architecture technique, choix des technologies |
| Développeur backend | Implémentation des API, logique métier, sécurité, gestion des workflows |
| Développeur frontend | Développement de l’interface web (Angular/React) et mobile (Flutter/React Native) |
| Expert DevOps | Déploiement, CI/CD, sécurité, supervision, sauvegardes |
| Expert sécurité | Audit, sécurisation des accès, conformité RGPD |
| Testeur QA | Tests fonctionnels, de charge, automatisés |
| Formateur | Formation des utilisateurs et administrateurs |
| Expert documentaire | Rédaction des guides utilisateurs et techniques |

**CHAPITRE III : STRUCTURE DE DONNEES**

**Support de Base de Données – MySQL**

**1. Introduction**

Ce document décrit en détail la structure et les choix techniques de la base de données MySQL pour la plateforme d’automatisation, afin d’assurer robustesse, performance et évolutivité.

**2. Objectifs de la base**

* Stocker en toute sécurité les informations métier : utilisateurs, demandes, pièces jointes, notifications, etc.
* Garantir la traçabilité complète des opérations (audit, historique).
* Optimiser les temps de réponse et les recherches grâce aux index.
* Faciliter l’évolution du schéma et l’ajout de nouvelles fonctionnalités.

**3. Modèle relationnel simplifié**

**Principales tables**

|  |  |
| --- | --- |
| **Table** | **Description rapide** |
| **Utilisateur** | Informations de connexion et profil des utilisateurs |
| **Demande** | Dossiers déposés par les utilisateurs |
| **PieceJointe** | Documents associés à chaque demande |
| **Notification** | Messages système envoyés aux utilisateurs |
| **LogSysteme** | Journalisation des actions pour audit et restauration |
| **TypeDemande** | Catégories et contraintes des demandes |
| **StatutDemande** | États possibles d’une demande |

**4. Détail de chaque table et de chaque colonne**

**Table Utilisateur**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description & Utilisation** |
| id | CHAR(36) | UUID généré côté application pour identifier un utilisateur. Immutable, utilisé en clé primaire. |
| login | VARCHAR(50) | Nom de connexion unique. Sert à l’authentification. |
| motDePasseHash | VARCHAR(255) | Stocke le hash sécurisé du mot de passe. Jamais en clair. |
| role\_id | INT | Clé étrangère vers Role.id. Détermine les permissions. |
| nom | VARCHAR(100) | Nom de famille, affiché dans l’interface utilisateur. |
| prenom | VARCHAR(100) | Prénom, pour l’affichage personnalisé. |
| email | VARCHAR(100) | Adresse email unique, utilisée pour notifications et identifiant secondaire. |
| telephone | VARCHAR(20) | Numéro de contact, facultatif, format libre (international). |
| dateCreation | TIMESTAMP | Date de création du compte, par défaut CURRENT\_TIMESTAMP. |
| derniereConnexion | TIMESTAMP | Date et heure de la dernière connexion. Mise à jour à chaque login. |
| actif | BOOLEAN | Indique si le compte est actif (1) ou suspendu (0). |

**Explications** :

* Le choix de CHAR(36) pour id assure la compatibilité avec les UUID v4 standard.
* Le champ dateCreation trace automatiquement l’enregistrement initial.
* actif permet de désactiver un compte sans le supprimer, pour conserver l’historique.

**Table Role**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | INT | Clé primaire, identifiant numérique. |
| libelle | VARCHAR(30) | Nom du rôle (ex : DEMANDEUR, ADMINISTRATEUR). Unique. |
| description | TEXT | Description détaillée du rôle et de ses droits associés. |

**Explications :**

* Permet de gérer finement les accès et les fonctionnalités disponibles pour chaque profil.

**Table TypeDemande**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | INT | Clé primaire numérique. |
| libelle | VARCHAR(50) | Code unique du type (ex : UNITE\_INDUSTRIELLE). |
| description | TEXT | Exposé du circuit et des conditions associées. |
| piecesRequises | JSON | Tableau JSON listant les pièces obligatoires. |
| delaiTraitement | INT | Délai moyen de traitement en jours. |
| fraisDossier | DECIMAL(10,2) | Montant des frais en monnaie locale. |

**Explications** :

* JSON utilisé pour simplifier l’ajout ou la modification des pièces sans altérer le schéma.

**Table StatutDemande**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | INT | Clé primaire. |
| libelle | VARCHAR(50) | État du dossier (ex : DEPOSEE, EN\_ATTENTE\_VALIDATION). |
| ordre | INT | Position dans le workflow pour faciliter les tris. |
| couleur | VARCHAR(7) | Code hexadécimal pour afficher l’état en couleur. |
| description | TEXT | Détail sur la signification de chaque statut. |

**Explications** :

* Le champ ordre permet de gérer le passage séquentiel d’un état à un autre.

**Table Demande**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | CHAR(36) | UUID unique. |
| utilisateurId | CHAR(36) | FK vers Utilisateur.id. |
| type\_id | INT | FK vers TypeDemande.id. |
| statut\_id | INT | FK vers StatutDemande.id. |
| dateDepot | TIMESTAMP | Horodatage du dépôt initial. |
| dateDerniereModification | TIMESTAMP | Mise à jour automatique à chaque modification (via trigger). |
| reference | VARCHAR(50) | Code généré pour référencer la demande (doit être unique). |
| objet | VARCHAR(255) | Brève description du projet ou dossier. |
| description | TEXT | Présentation détaillée du contenu de la demande. |
| metadonnees | JSON | Stocke les informations spécifiques (ex : superficie, etc.). |

**Explications** :

* Le trigger update\_date\_derniere\_modification met à jour dateDerniereModification.
* metadonnees offre une flexibilité maximale pour les champs particuliers.

**Table PieceJointe**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | CHAR(36) | UUID unique. |
| demandeId | CHAR(36) | FK vers Demande.id. |
| nomFichier | VARCHAR(255) | Nom original du fichier. |
| typePiece | VARCHAR(100) | Nature du document (ex : carte d&apos;identité). |
| chemin | VARCHAR(255) | Chemin d&apos;accès sur le serveur. |
| dateUpload | TIMESTAMP | Horodatage de l&apos;envoi. |
| statut | VARCHAR(20) | Indique la validité de la pièce (VALIDE, EN\_ATTENTE...). |
| taille | INT | Poids du fichier en octets. |
| format | VARCHAR(10) | Extension du fichier (PDF, JPG, PNG...). |
| commentaire | TEXT | Observations éventuelles lors de la vérification. |

**Explications** :

* Permet de tracer l’adresse exacte et le statut de chaque document.

**Table Notification**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | CHAR(36) | UUID unique. |
| destinataireId | CHAR(36) | FK vers Utilisateur.id. |
| contenu | TEXT | Texte intégral de la notification. |
| dateEnvoi | TIMESTAMP | Horodatage de la notification. |
| lu | BOOLEAN | Indique si la notification a été consultée. |
| dateLecture | TIMESTAMP | Si lu, date de la consultation. |
| typeNotification | VARCHAR(50) | Catégorie (INFO, ALERTE, ...) |
| lien | VARCHAR(255) | URL relative ou absolue pointant vers la ressource. |

**Explications** :

* Facilite la redirection contextuelle (un clic sur la notification ouvre la page cible).

**Table LogSysteme**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Type** | **Description** |
| id | SERIAL | Identifiant incrémental automatique. |
| utilisateurId | CHAR(36) | FK vers Utilisateur.id, pour savoir qui a agi. |
| action | VARCHAR(100) | Opération détectée (INSERT, UPDATE, DELETE). |
| entite | VARCHAR(50) | Table cible de l’action. |
| entiteId | VARCHAR(50) | ID de l’élément impacté. |
| dateHeure | TIMESTAMP | Horodatage de l’action. |
| ancienneValeur | TEXT | État avant modification (JSON). |
| nouvelleValeur | TEXT | État après modification (JSON). |
| adresseIP | VARCHAR(45) | Adresse IP client initiant l’action. |

**Explications** :

* Les triggers trig\_log\_update\_demande, trig\_log\_insert\_demande et trig\_log\_delete\_demande alimentent automatiquement cette table.

**5. Contraintes et intégrité référentielle**

* **Clés primaires** (PK) sur tous les id.
* **Clés étrangères** (FK) pour garantir l’existence des références.
* **Contraintes UNIQUE** sur les colonnes nécessitant une unicité (ex : login, email, reference).
* **Contraintes CHECK** pour valider certaines valeurs (ex : statut dans PieceJointe).

**6. Optimisation (Index et Vues)**

L’optimisation de la base de données repose sur deux piliers majeurs : les **index** (pour accélérer les requêtes) et les **vues** (pour simplifier et réutiliser les requêtes complexes).

**Index**

Un index est une structure de données associée à une ou plusieurs colonnes d’une table, permettant d’accélérer la recherche et le filtrage des lignes.

* **Principe :** sans index, MySQL doit effectuer un full table scan (parcours de toutes les lignes) pour répondre à une requête SELECT ... WHERE. Avec un index, il utilise un arbre B-tree pour localiser rapidement les lignes correspondant au critère.
* **Coût :**
* ↗️ Lecture plus rapide (O(log N) au lieu de O(N)).
* ↘️ Légère surcharge à l’insertion et à la mise à jour (car l’index doit être maintenu).

**Index créés dans ce schéma :**

|  |  |  |
| --- | --- | --- |
| Nom d’index | Colonne(s) ciblée(s) | Usage principal |
| idx\_demande\_utilisateur | Demande(utilisateurId) | Accélère les requêtes filtrant les demandes par utilisateur (ex. historique personnel). |
| idx\_demande\_type | Demande(type\_id) | Rapide sélection des demandes selon leur type. |
| idx\_demande\_statut | Demande(statut\_id) | Filtrage des demandes par statut (ex. en attente). |
| idx\_notification\_destinataire | Notification(destinataireId) | Recherche des notifications pour un utilisateur rapide. |
| idx\_piecejointe\_statut | PieceJointe(statut) | Tri et comptage des pièces selon leur état de validation. |

**Vues**

Une **vue** est une table virtuelle résultant d’une requête SQL stockée. Elle ne contient pas de données en propre (sauf si matérialisée) mais facilite la réutilisation et la maintenance.

* **Avantages** :
* **Lisibilité** : encapsule des JOIN et des calculs complexes.
* **Réutilisation :** la même logique est partagée partout, évitant la duplication.
* **Sécurité** : on peut restreindre l’accès à la vue plutôt qu’à la table de base.

**Vues définies** :

|  |  |
| --- | --- |
| **Vue** | **Description** |
| vue\_demandes\_completes | Agrège pour chaque demande : |

* Infos principales (id, reference, dateDepot)
* Nom complet du demandeur
* Type et statut en clair
* Nombre de pièces jointes
* Nombre d’actions de suivi | | vue\_statistiques | Fournit des indicateurs globaux :
* Nombre de demandes par combinaison type/statut
* Durée moyenne de traitement en jours (calculée par TIMESTAMPDIFF) |

**Utilisation pratique** :

* Dans l’interface admin, interroger vue\_demandes\_completes pour afficher la liste des dossiers sans écrire plusieurs JOIN.
* Dans le dashboard, appeler vue\_statistiques pour générer des graphiques et **KPI**.

**Qu’est-ce qu’un KPI ? Un KPI (Key Performance Indicator), ou indicateur clé de performance,** est une mesure quantifiable permettant d’évaluer l’efficacité d’un processus ou d’une activité. Dans le contexte de la plateforme :

* Nombre de demandes traitées en un laps de temps donné
* Durée moyenne de traitement des dossiers
* Taux de demandes en attente versus traitées Ces KPI aident les responsables à prendre des décisions éclairées, à ajuster les ressources et à optimiser les workflows en identifiant rapidement les points de blocage.

**7. Évolutivité**

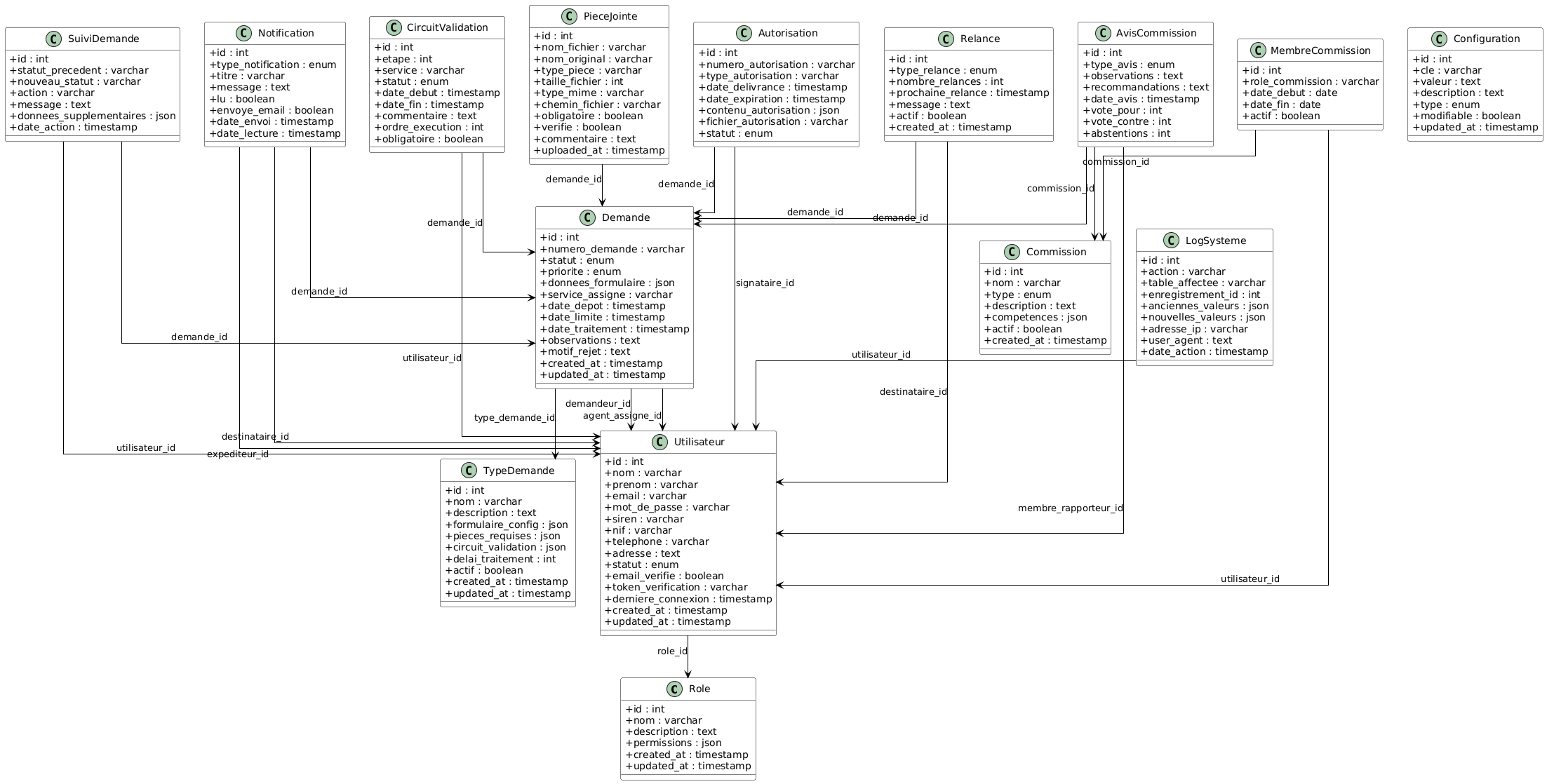
* Ajout de tables pour modules additionnels (paiement, signature numérique...).
* Extensions JSON pour métadonnées et configuration dynamique.
* Migration simple vers d’autres SGBD (PostgreSQL, MariaDB).

**Diagramme de classe de la gestion des autorisations**



**Structure détaillée :**

**Analyse détaillée du diagramme de classes - Système de gestion de demandes d'autorisation**



**Vue d'ensemble du système**

Ce diagramme représente un système complet de gestion de demandes d'autorisation, destiné aux administrations publiques ou organismes de régulation. Le système gère l'intégralité du processus d'octroi d'autorisations : depuis la soumission des demandes par les citoyens/entreprises jusqu'à la délivrance des autorisations officielles, en passant par l'instruction, la validation par les services compétents et les commissions spécialisées.

**Entités principales et leurs fonctionnalités**

**1. Role - Gestion des rôles et permissions**

**Fonctionnalités :**

* Définition des rôles système (administrateur, agent, citoyen, etc.)
* Gestion granulaire des permissions via un champ JSON
* Contrôle d'accès basé sur les rôles (RBAC)

**Champs clés :**

* permissions (JSON) : stocke les permissions spécifiques pour chaque rôle
* description : documentation du rôle et de ses responsabilités

**2. Utilisateur - Cœur du système d'authentification**

**Fonctionnalités principales :**

* Authentification et autorisation des utilisateurs
* Gestion des profils citoyens et agents
* Vérification d'identité via email
* Traçabilité des connexions

**Particularités métier :**

* siren/nif : identifiants entreprise/particulier (contexte français)
* email\_verifie + token\_verification : processus de validation email
* statut : gestion des comptes (actif, suspendu, en attente)
* derniere\_connexion : audit et sécurité

**3. TypeDemande - Configuration des types d'autorisation**

**Fonctionnalités avancées :**

* Configuration dynamique des formulaires selon le type d'autorisation
* Définition des pièces justificatives spécifiques à chaque autorisation
* Paramétrage des circuits d'instruction et de validation
* Gestion des délais réglementaires de traitement

**Flexibilité du système :**

* formulaire\_config (JSON) : formulaires adaptés (permis, licence, agrément, etc.)
* pieces\_requises (JSON) : documents obligatoires selon la réglementation
* circuit\_validation (JSON) : processus d'instruction réglementaire

**4. Demande - Entité centrale du processus d'autorisation**

**Cycle de vie complet d'une demande d'autorisation :**

* Création et numérotation automatique des dossiers
* Gestion des statuts (en instruction, avis favorable, autorisé, refusé)
* Assignation aux services instructeurs compétents
* Gestion des priorités et respect des délais réglementaires

**Fonctionnalités métier spécifiques :**

* donnees\_formulaire (JSON) : informations spécifiques à l'autorisation demandée
* service\_assigne : service instructeur selon la compétence
* date\_limite : respect des délais légaux d'instruction
* motif\_rejet : motivation obligatoire en cas de refus

**5. PieceJointe - Gestion documentaire**

**Fonctionnalités de GED (Gestion Électronique de Documents) :**

* Upload sécurisé avec validation du type MIME
* Gestion des tailles de fichiers
* Marquage obligatoire/optionnel
* Processus de vérification des documents

**Sécurité et conformité :**

* chemin\_fichier : stockage sécurisé
* verifie : validation par un agent
* commentaire : feedback sur la conformité

**6. SuiviDemande - Traçabilité complète**

**Audit trail complet :**

* Historique de tous les changements de statut
* Traçabilité des actions utilisateur
* Données supplémentaires en JSON pour flexibilité

**Fonctionnalités :**

* Permet de reconstituer l'historique complet d'une demande
* Aide au débogage et à l'analyse des processus
* Base pour les rapports et statistiques

**7. Notification - Système de communication**

**Communication multi-canal :**

* Notifications dans l'application
* Envoi d'emails automatiques
* Gestion du statut de lecture
* Communication bidirectionnelle

**Fonctionnalités avancées :**

* type\_notification : catégorisation (alerte, info, rappel)
* envoye\_email : traçabilité des envois
* Liens avec expéditeur et destinataire

**8. CircuitValidation - Workflow de validation**

**Gestion des processus métier :**

* Définition d'étapes de validation séquentielles
* Assignation par service ou utilisateur
* Gestion de l'ordre d'exécution
* Étapes obligatoires vs optionnelles

**Flexibilité organisationnelle :**

* Adaptation aux différents types de demandes
* Possibilité de validation parallèle ou séquentielle
* Commentaires à chaque étape

**9. Autorisation - Délivrance des autorisations officielles**

**Gestion des actes administratifs d'autorisation :**

* Génération d'autorisations, permis, licences, agréments officiels
* Numérotation unique et traçable des actes
* Gestion des périodes de validité et dates d'expiration
* Stockage du contenu juridique et du document officiel

**Fonctionnalités légales et réglementaires :**

* signataire\_id : validation par l'autorité compétente
* contenu\_autorisation (JSON) : conditions, prescriptions, obligations
* fichier\_autorisation : document officiel signé (PDF sécurisé)
* date\_expiration : gestion des renouvellements

**10. Relance - Gestion proactive**

**Automatisation des relances :**

* Programmation automatique des rappels
* Escalade progressive
* Personnalisation des messages

**Intelligence du système :**

* nombre\_relances : éviter le spam
* prochaine\_relance : planification automatique
* actif : possibilité de désactiver

**11. Commission - Instances décisionnaires pour les autorisations**

**Gestion des commissions d'autorisation :**

* Commissions spécialisées par type d'autorisation (urbanisme, environnement, sécurité)
* Définition des compétences et domaines d'intervention
* Gestion des types (consultative, décisionnaire)
* Configuration des seuils de compétence

**12. MembreCommission - Composition des commissions**

**Gestion des mandats :**

* Historisation des membres
* Gestion des rôles au sein de la commission
* Périodes de mandat avec dates de début/fin

**13. AvisCommission - Délibérations sur les autorisations**

**Processus décisionnel collégial :**

* Enregistrement des avis (favorable, défavorable, favorable avec réserves)
* Gestion des votes et des délibérations
* Désignation du rapporteur pour chaque dossier
* Prescriptions et conditions d'autorisation

**Traçabilité institutionnelle :**

* Lien avec la demande d'autorisation concernée
* Date et contexte de l'avis rendu
* Recommandations pour la décision finale

**14. Configuration - Paramétrage système**

**Administration flexible :**

* Paramétrage global de l'application
* Gestion des valeurs métier
* Contrôle des modifications

**Types de configuration :**

* Délais par défaut
* Messages standards
* Paramètres d'intégration

**15. LogSysteme - Audit et sécurité**

**Conformité et sécurité :**

* Traçabilité complète des actions
* Stockage des anciennes et nouvelles valeurs
* Information de contexte (IP, navigateur)

**Fonctionnalités légales :**

* Preuve en cas de litige
* Audit de conformité RGPD
* Analyse forensique

**Relations et cardinalités importantes**

**Relations principales :**

1. **Utilisateur ↔ Role** (n:1) : Système de permissions
2. **Demande ↔ Utilisateur** (n:1) : Propriété et assignation
3. **Demande ↔ TypeDemande** (n:1) : Typage des demandes
4. **PieceJointe ↔ Demande** (n:1) : Documents attachés
5. **CircuitValidation ↔ Demande** (n:1) : Processus de validation

**Relations complexes :**

* **Commission ↔ Utilisateur** via **MembreCommission** : Relation many-to-many avec historique
* **Notification** : Relations multiples (expéditeur, destinataire, demande)

**Fonctionnalités système globales**

**1. Workflow Management**

* Définition flexible des processus
* Assignation automatique ou manuelle
* Gestion des délais et escalades

**2. Document Management**

* Upload sécurisé
* Validation des formats
* Archivage organisé

**3. Communication**

* Notifications temps réel
* Emails automatiques
* Système de relances

**4. Audit et Conformité**

* Traçabilité complète
* Logs système
* Historique des modifications

**5. Gestion des droits**

* RBAC (Role-Based Access Control)
* Permissions granulaires
* Sécurité multi-niveaux

**6. Reporting et Analytics**

* Suivi des demandes
* Statistiques de performance
* Analyse des processus

**Architecture technique**

Le système utilise une architecture orientée données avec :

* **Flexibilité** via les champs JSON
* **Extensibilité** par configuration
* **Traçabilité** complète
* **Sécurité** par design
* **Évolutivité** des processus métier

Cette conception permet d'adapter le système à différents types d'administrations tout en maintenant une cohérence fonctionnelle.

**8. Conclusion**

Ce support très détaillé présente chaque colonne et ses choix techniques, justifiant la structure et garantissant l’efficacité, la sécurité et la maintenabilité de la base MySQL pour la plateforme d’automatisation.

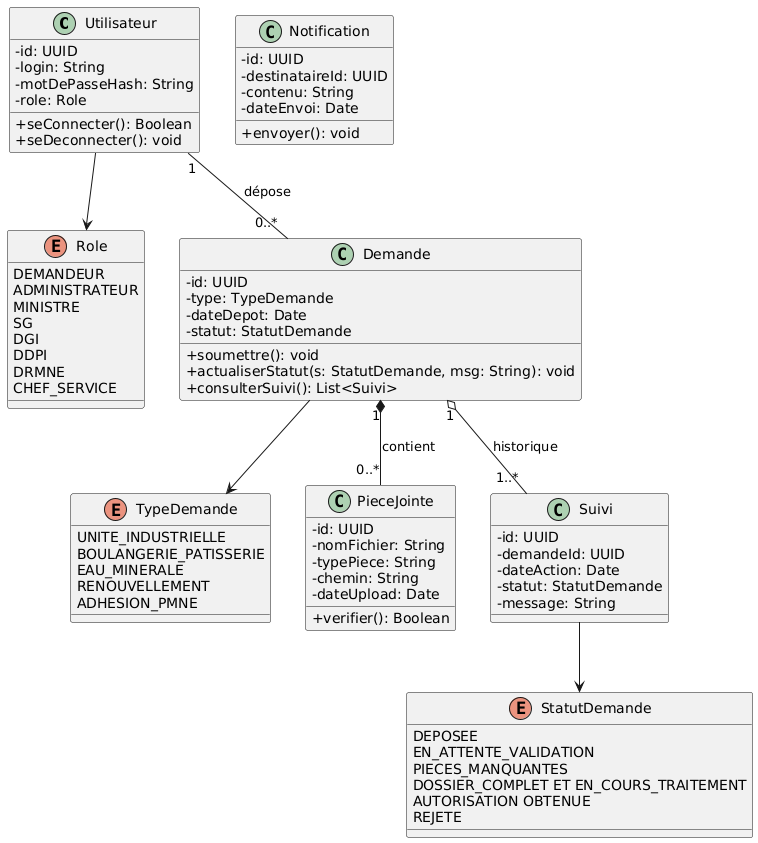
**CHAPITRE V : CONCEPTION**

**5.1 Modélisation UML**

# Scénarios

**Référence à la scénérios détaillées**

**a) Diagramme de classes**



Entités principales : Utilisateur, Demande, PièceJointe, Notification, Suivi, TypeDemande, StatutDemande, Role

**Relations :**

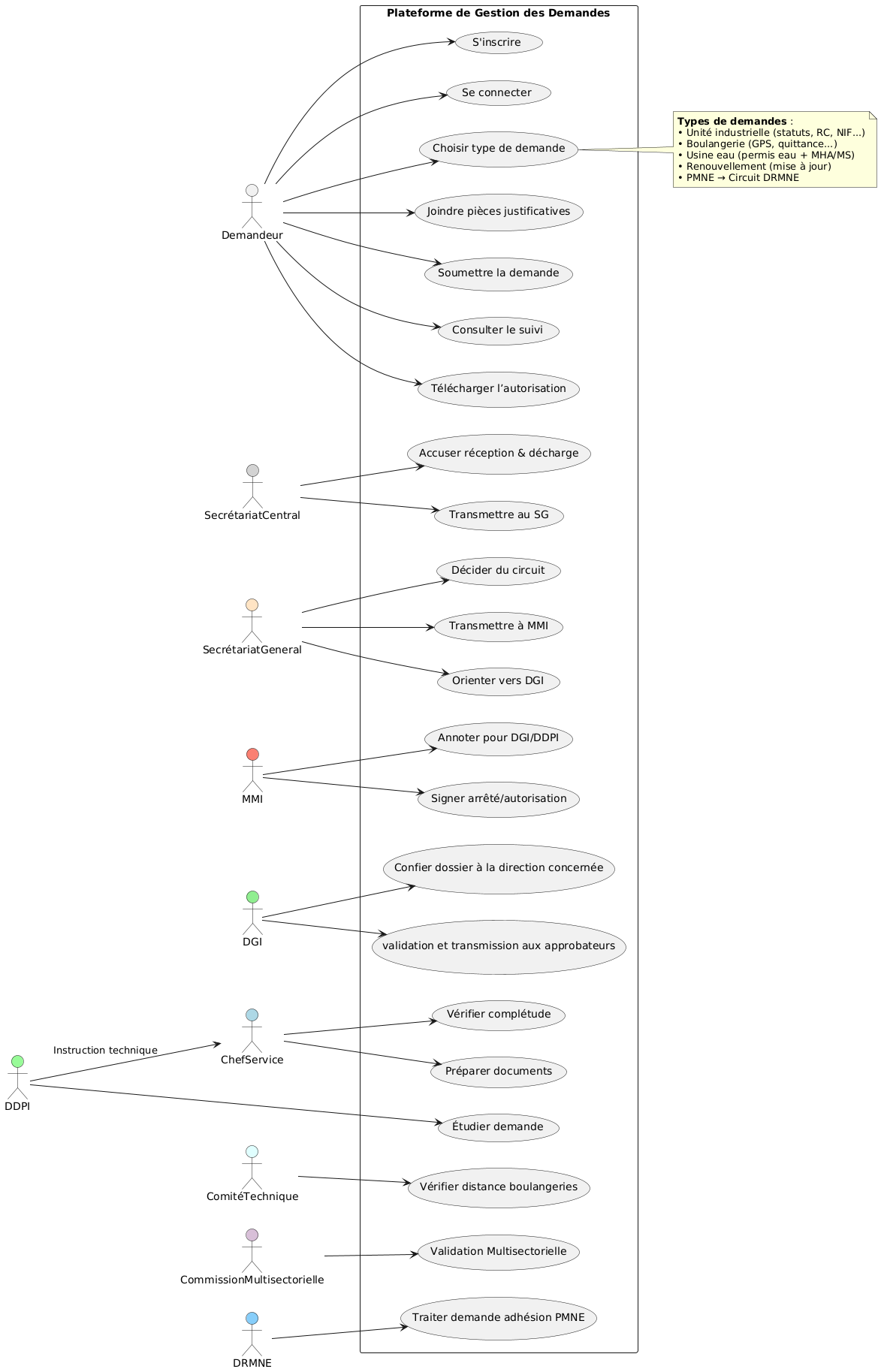
Un utilisateur dépose plusieurs demandes

Une demande contient plusieurs pièces jointes et un historique de suivi

Les notifications sont liées aux utilisateurs

**b) Diagramme de cas d’utilisation**

Voir Image 2 (ci-dessus)

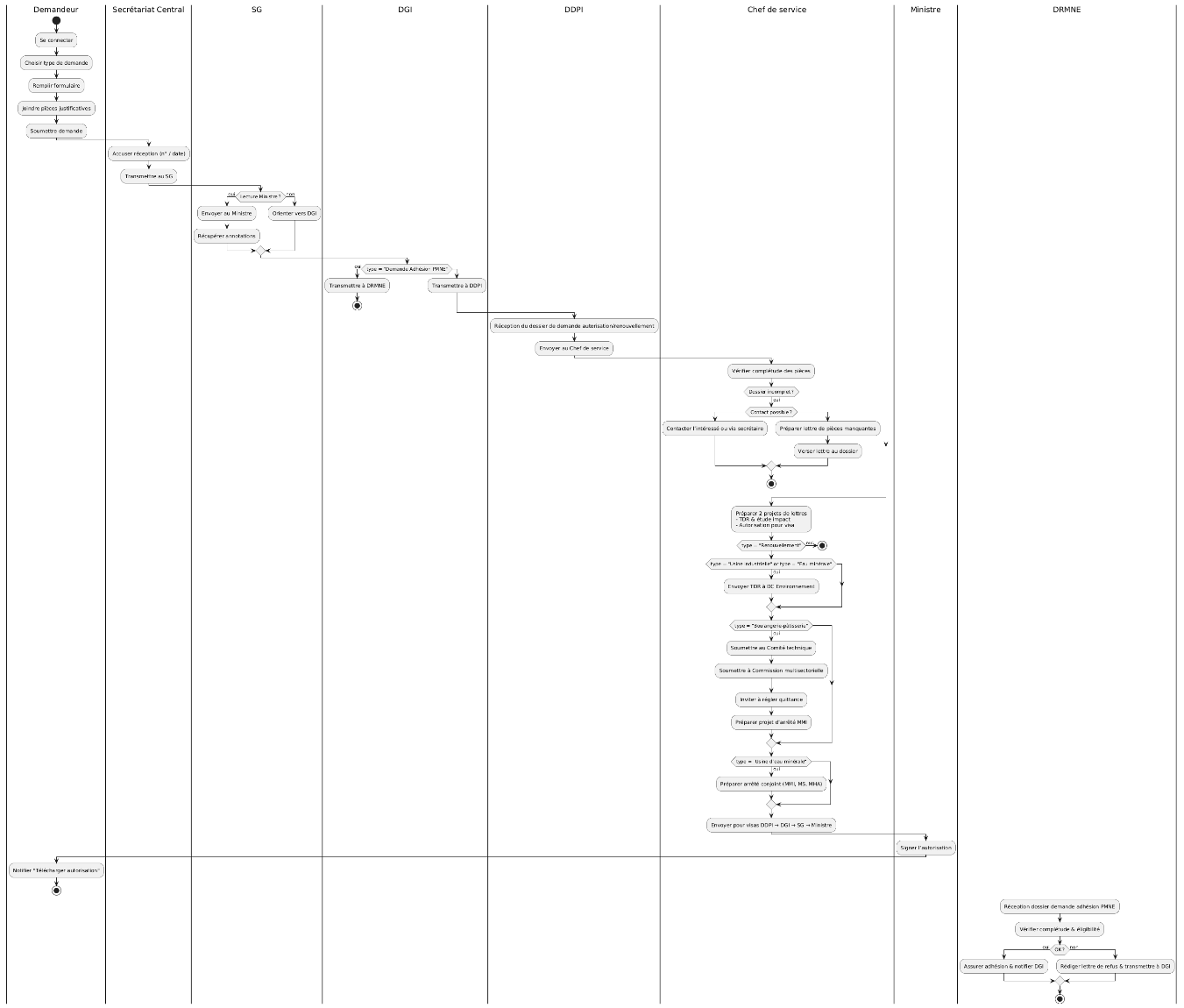


Parcours du demandeur : inscription, connexion, choix du type de demande, dépôt, suivi, téléchargement de l’autorisation

Parcours des agents : réception, validation, transmission, annotation, signature, instruction technique, validation multisectorielle

**c) Diagramme d’activité**

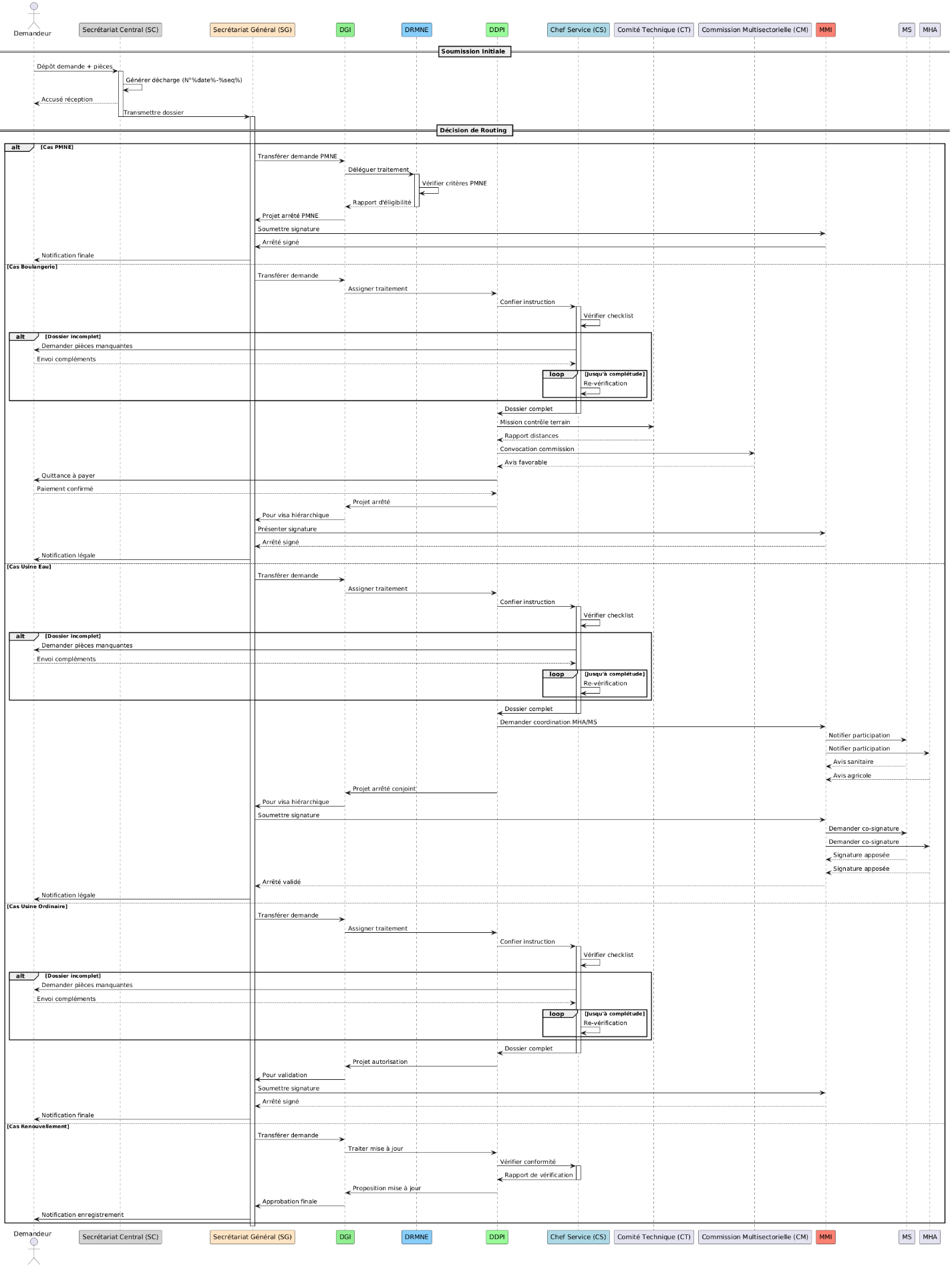
Voir Image 3 (ci-dessus)



Workflow détaillé : dépôt, réception, transmission, instruction, validation, signature, notification, archivage

**d) Diagramme de séquence**

Voir Image 4 (ci-dessus)



Orchestration des échanges entre acteurs à chaque étape du processus

**5.2 Fonctionnalités principales**

**Espace Demandeur**

Création de compte sécurisé

Dépôt de demande selon le type

Téléversement des pièces justificatives

Suivi du statut et téléchargement de l’autorisation

**Espace Administration**

Réception et vérification des demandes

Répartition et traitement par les services compétents

Historisation et archivage des décisions

**Moteur de workflow**

Enchaînement automatisé des étapes selon le type de demande

Alertes et notifications automatiques

Suivi en temps réel et traçabilité complète

**Reporting et tableaux de bord**

Statistiques sur les demandes par type, statut, délai

Export des données pour analyse

**6. CONCEPTION TECHNIQUE**

**6.1 Architecture technique**

**Frontend** : application web responsive et mobile

**Backend** : API RESTful, moteur de workflow, gestion documentaire

**Base de données :** relationnelle (PostgreSQL/MySQL), stockage sécurisé des documents

**Sécurité :** authentification forte, gestion fine des droits, chiffrement, journalisation

**Déploiement :** conteneurisation (Docker), CI/CD, sauvegardes automatiques

**6.2 Schéma de la base de données (extrait du diagramme de classes) voir la partie structure de données chapitre III**

**Utilisateur** (UUID, login, motDePasseHash, role)

**Role** (libellé, description)

**Demande** (UUID, type, dateDepot, statut)

**TypeDemande** (libellé, description, pièces requises)

**PieceJointe** (UUID, nomFichier, typePiece, chemin, dateUpload, statut)

**Suivi** (UUID, demandeId, utilisateurId, dateAction, statut, message)

**StatutDemande** (libellé, description)

**Notification** (UUID, destinataireId, contenu, dateEnvoi, lu, typeNotification)

**CHAPITRE V : Wireframes**

**Introduction**

**1 – Page Index**

**Page d’accueil de la plateforme MMIAPP** : cette interface présente la plateforme de gestion des demandes du Ministère des Mines et de l’Industrie, avec accès aux sections principales (Admin, User, Menu, etc.) et un en-tête trilingue (français, anglais, arabe).



**2 . Page Demandeur**

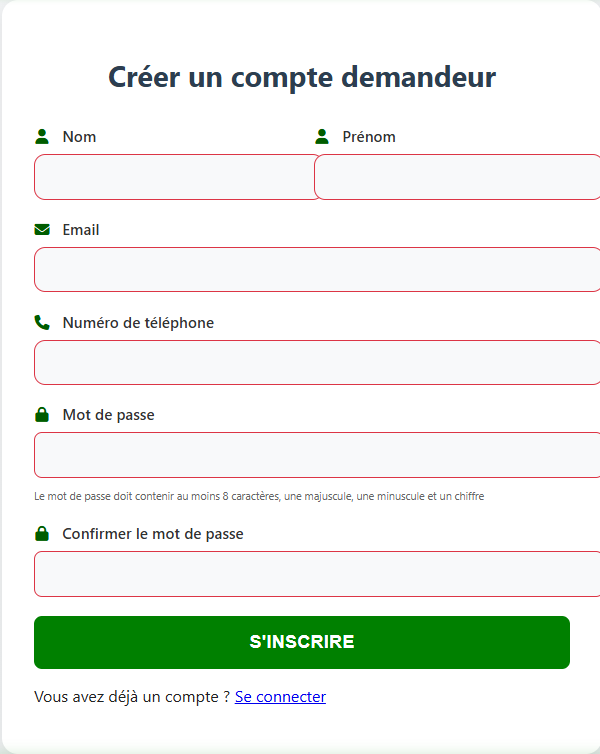
**A – Accés à la page Demandeur**

Accès à l’espace Demandeur : cette section permet aux utilisateurs de rejoindre leur espace personnel afin de gérer et suivre l’état de leurs demandes.



**B- Inscription**

Pour l’utilisateur puisse acceder a l’espace demandeur, il est obligatoire de rensigner le formulaire d’inscription. une fois ce dernier remplis un boutton s’inscrire lui permettra de valider la procedure d’inscription.



**C- Mot de passe oublié**

Cette interface permet à l'utilisateur de réinitialiser son mot de passe en saisissant son adresse e-mail. Une fois l’adresse saisie, il peut cliquer sur le bouton « Envoyer le lien » pour recevoir un lien de réinitialisation. Un lien « Retour à la connexion » permet également de revenir à la page de connexion.



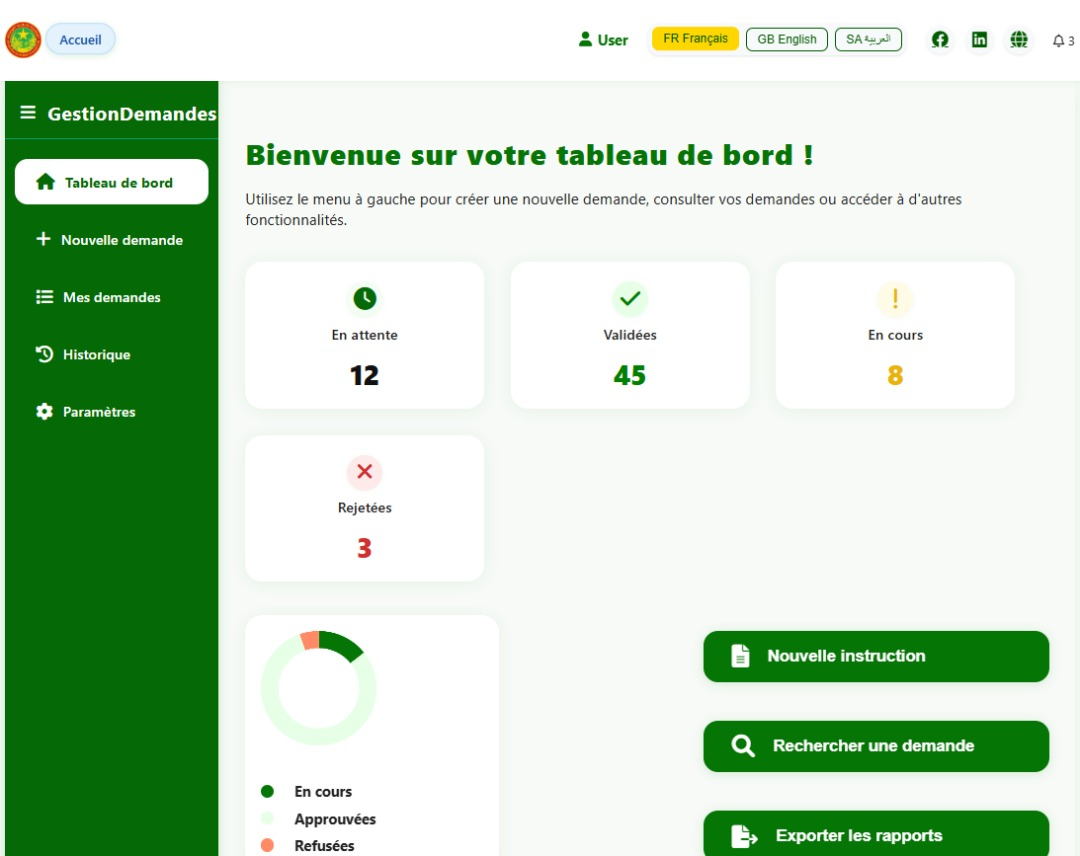
**D- Login**

Interface de connexion du demandeur : cette section permet à l'utilisateur demandeur de se connecter à son compte en saisissant son email et son mot de passe.



E-Dashboard Demandeur

Tableau de bord du demandeur : cette interface affiche un résumé des demandes (en attente, validées, en cours, rejetées) et permet d'accéder rapidement aux actions comme créer une demande, consulter l’historique ou exporter des rapports.



F- Choix De Type de demand

Sur le tableau de bord, il clique sur ‘Nouvelle demande’

Il sélectionne le type de demande : unité industrielle, boulangerie-pâtisserie, usine d’eau minérale, renouvellement, adhésion PMNE



**G- Remplissage Du Formulaire**

Le formulaire dynamique affiche les champs spécifiques au type choisi.

Le demandeur remplit les informations obligatoires (ex : statuts, permis, GPS, etc.).Téléversement des pièces justificatives

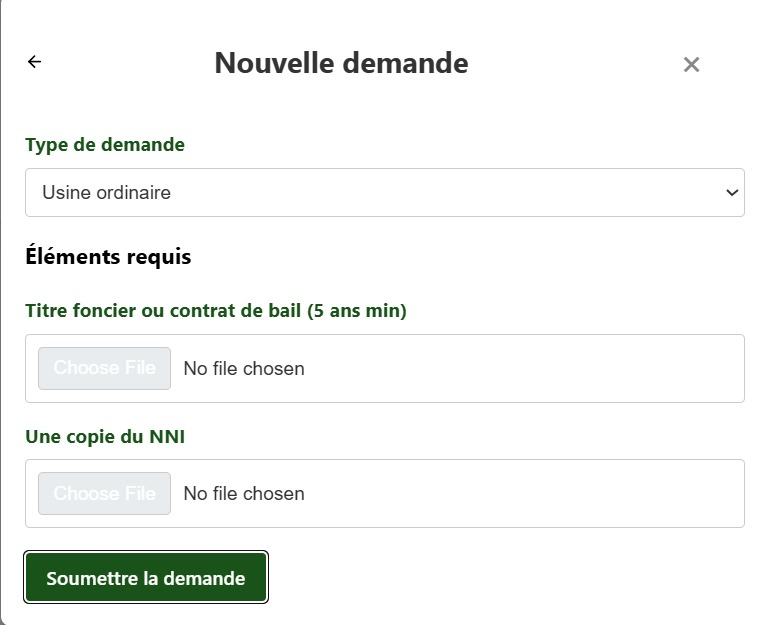
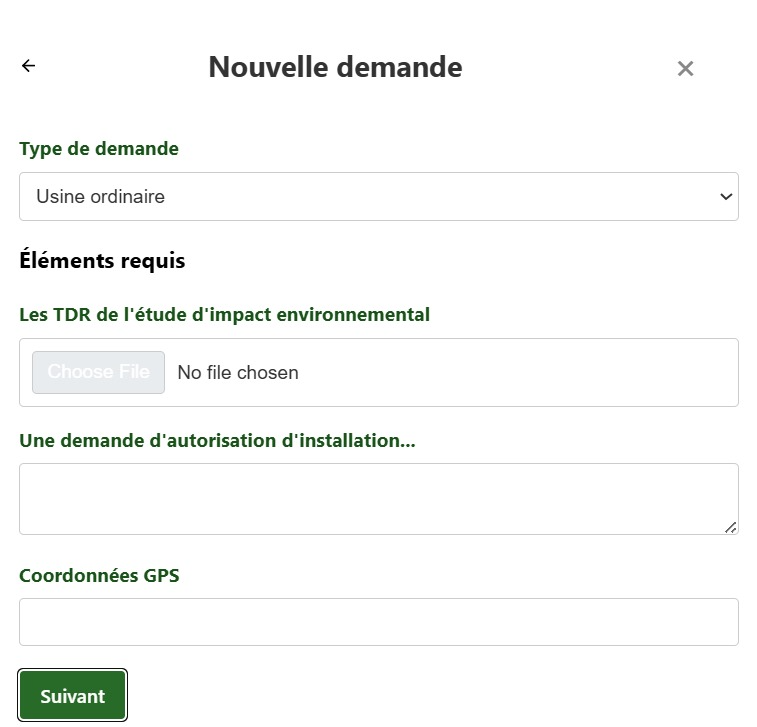
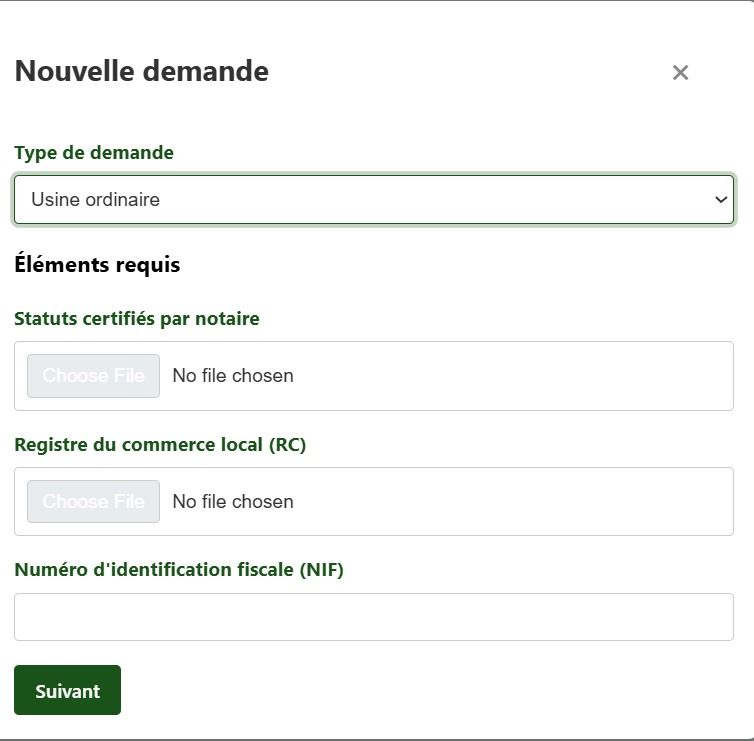
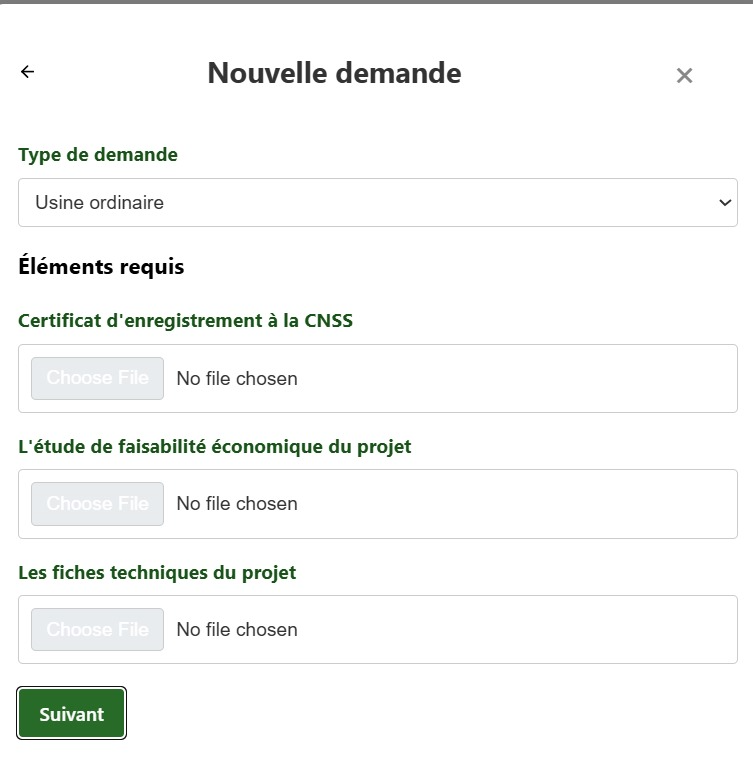
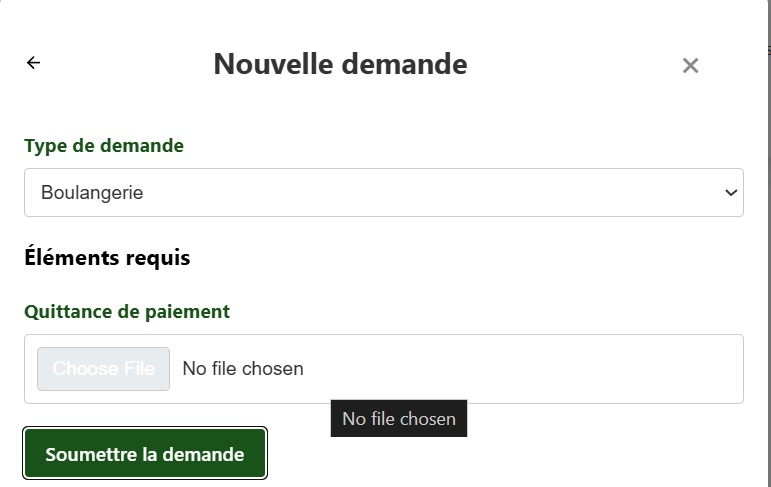
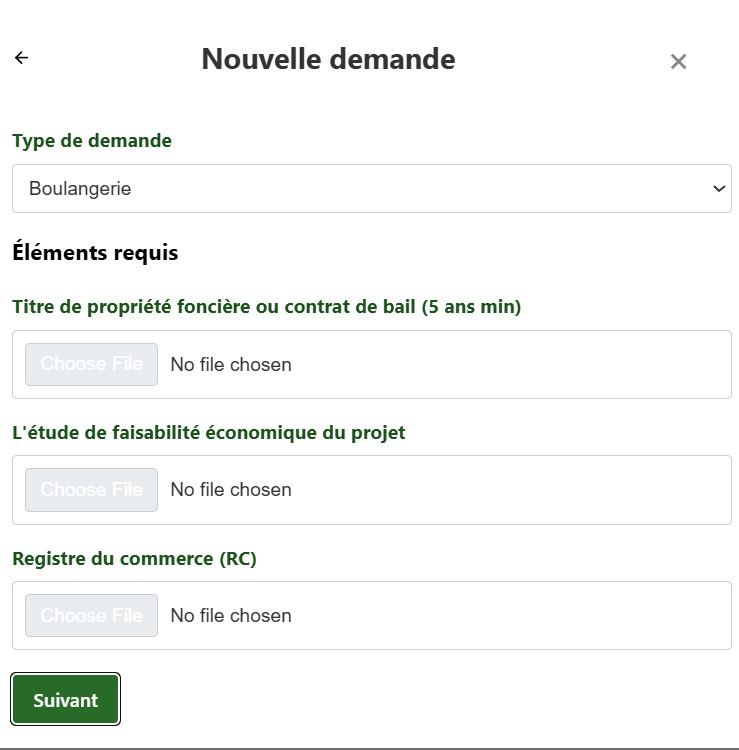
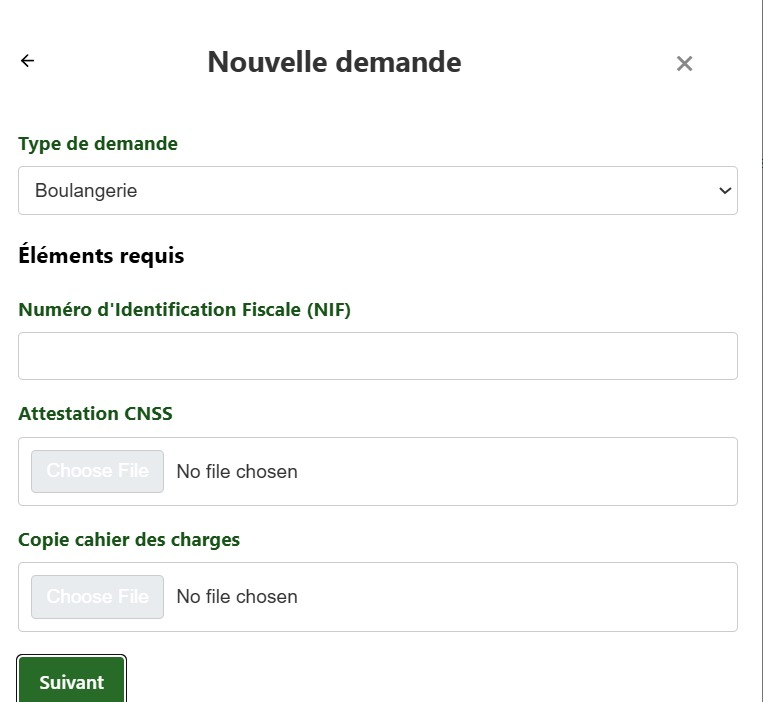
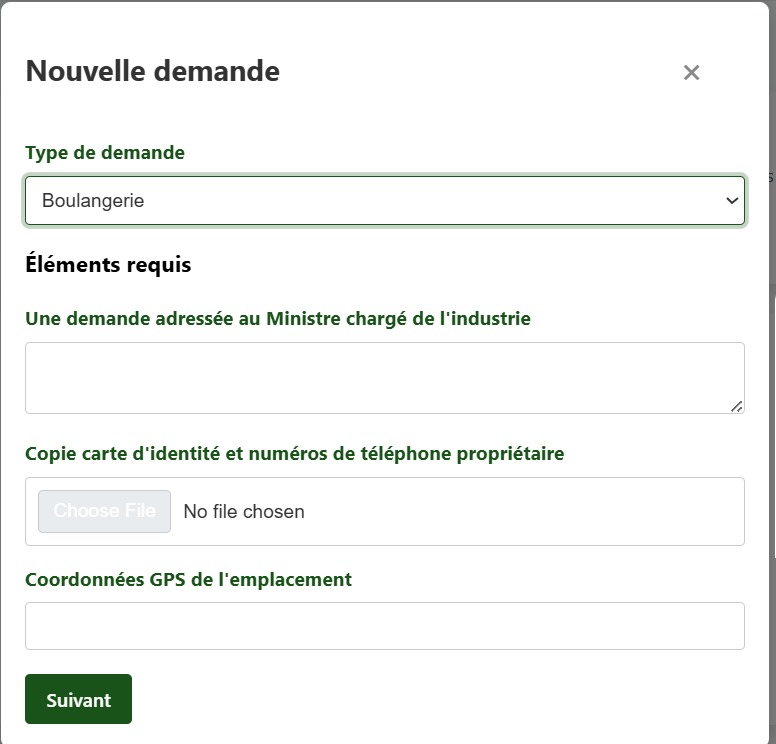
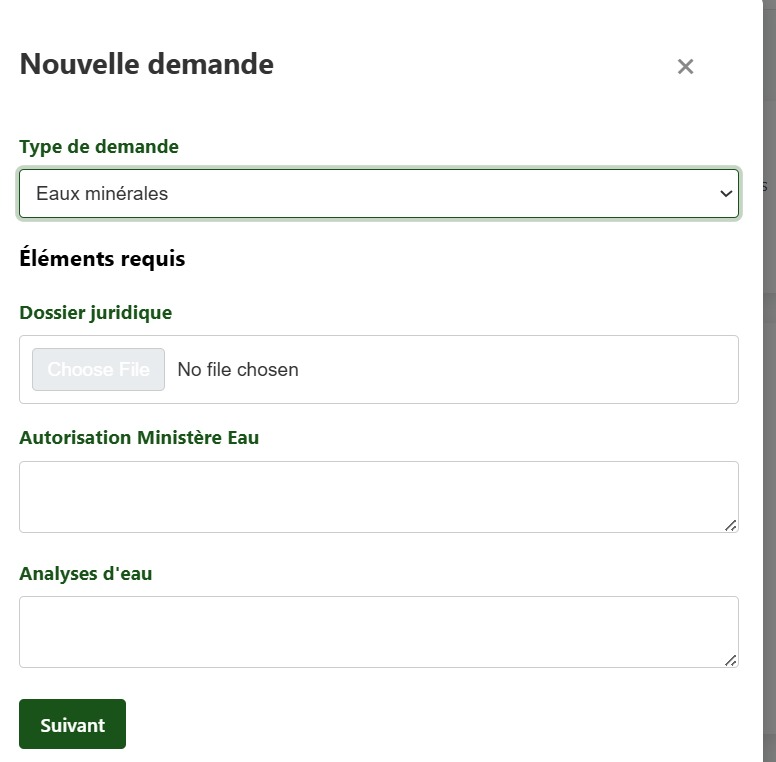
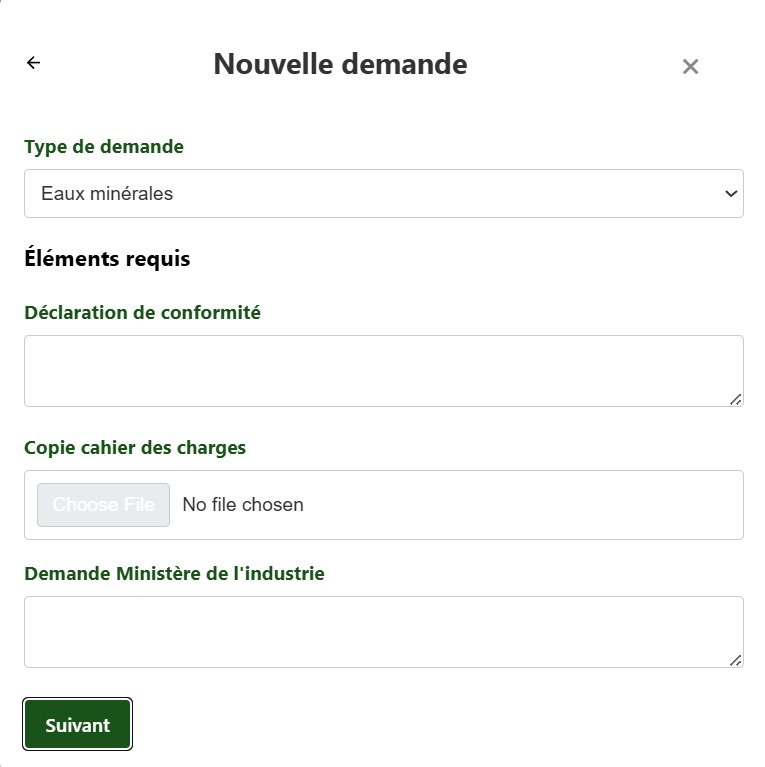
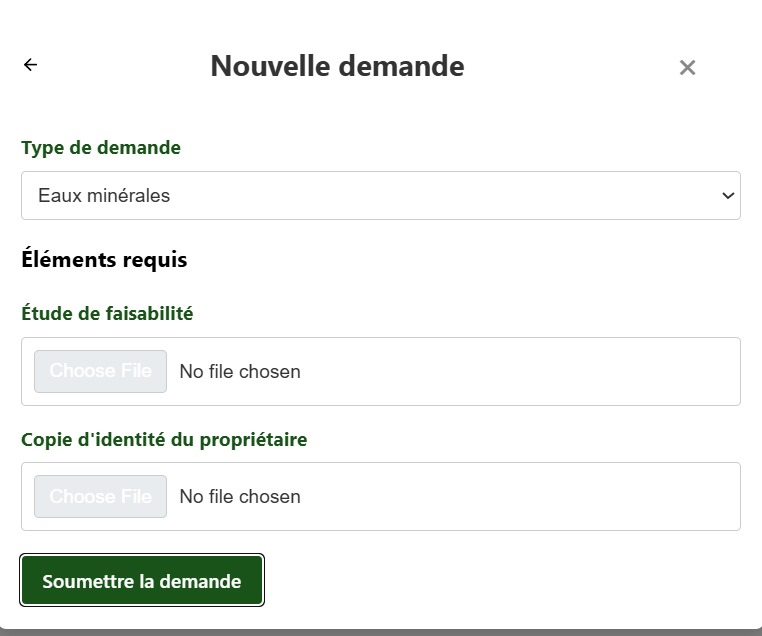
Il joint les documents demandés (PDF, images…).

Le système vérifie le format et la complétude des pièces.

Soumission de la demande

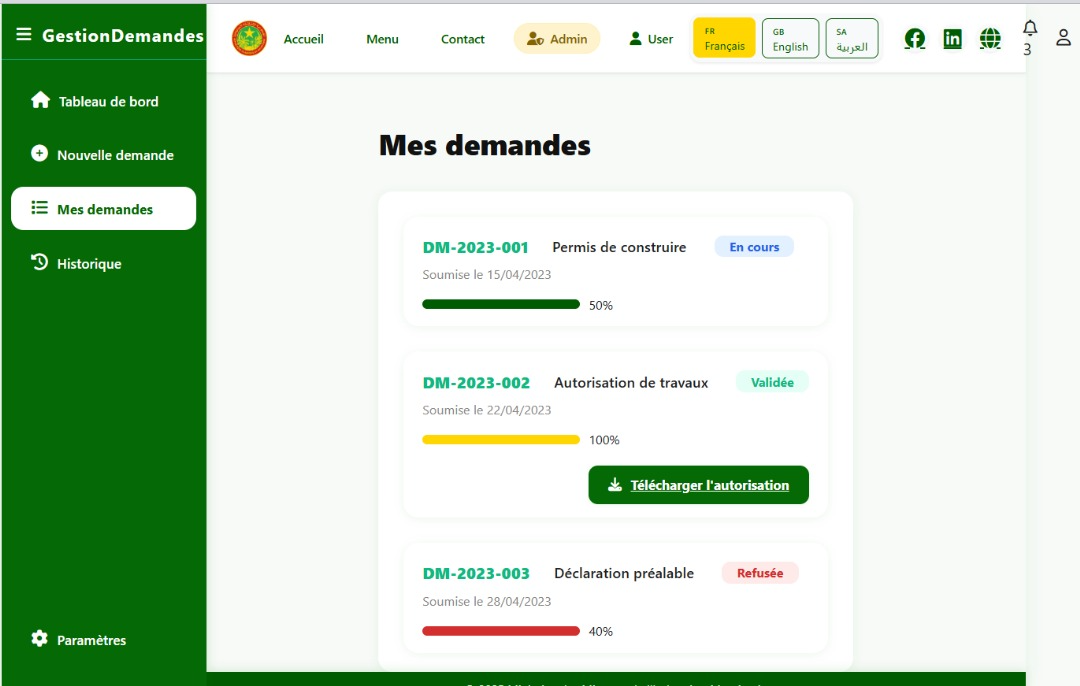
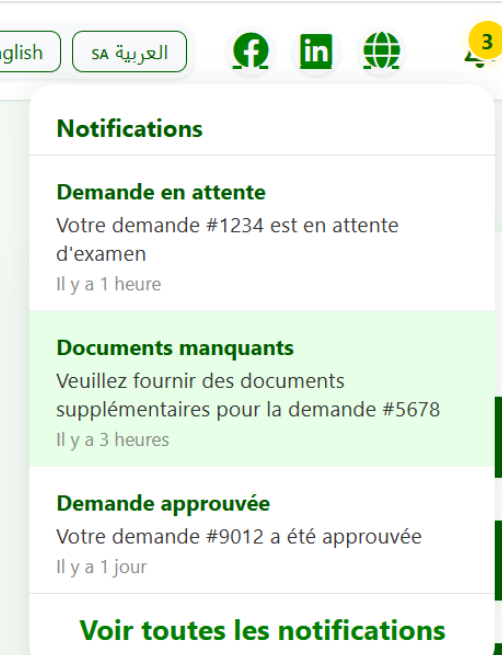
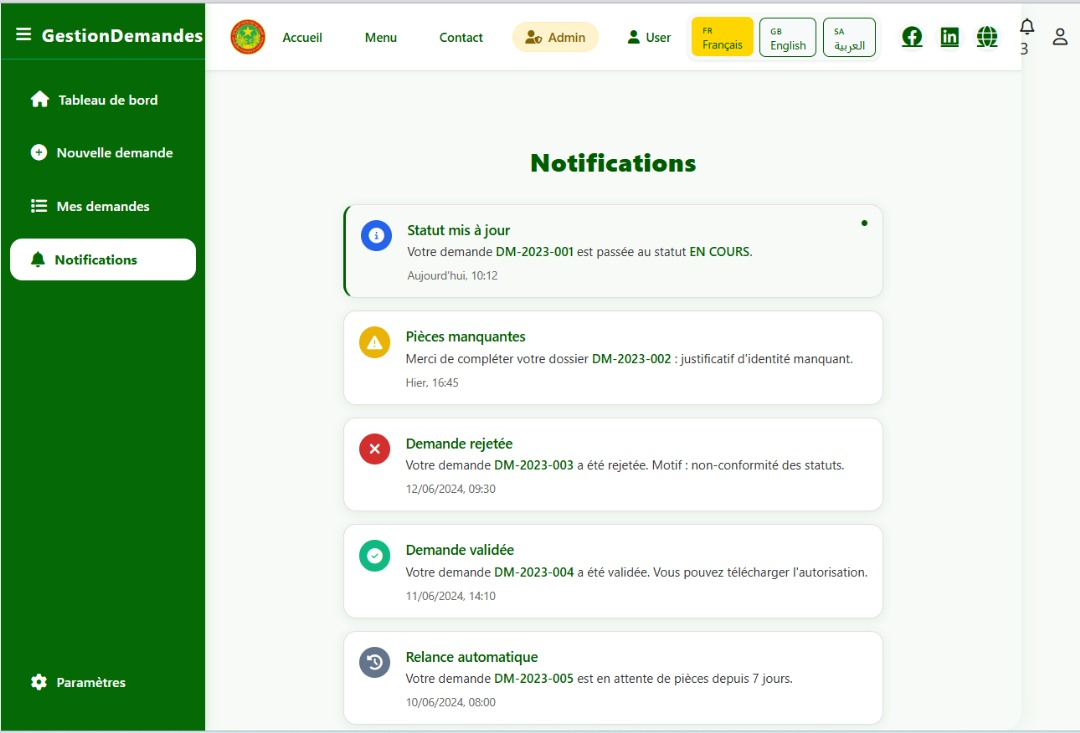
Il valide et soumet la demande.

Les Differentes Types Des Formulaire De Chaque Demandes:



H- Les Demandes Et Les Notification

**Le système enregistre la demande avec le statut « DEPOSEE » et notifie le demandeur avec la réponse de**



**3-Page Adminstrateur/**

**A-Accés à La Page Adminstrateur**

Cette carte d’accès permet aux membres autorisés de l’administration de se connecter à un espace sécurisé.



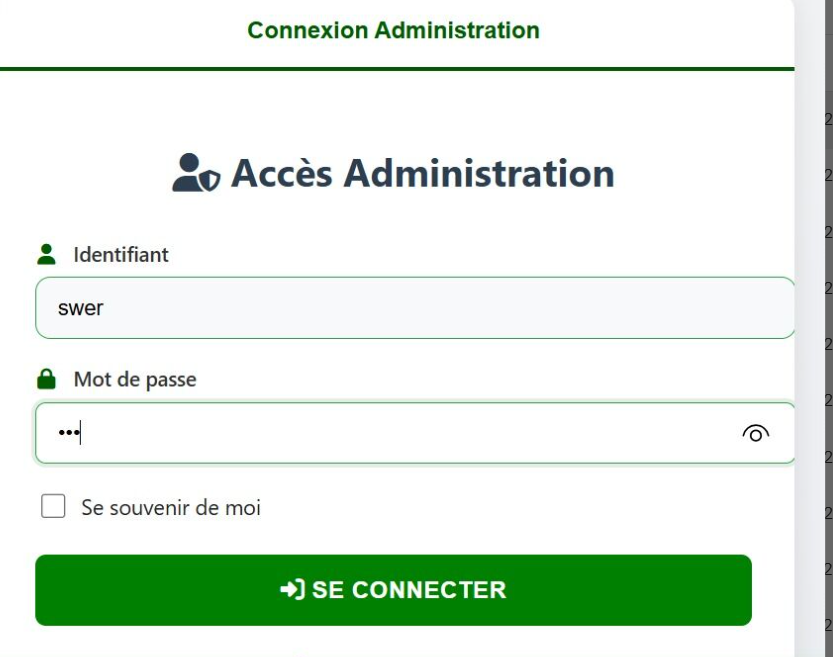
**B-Page De L’Espace Adminstrateur**

Cette page affiche les différents accès sécurisés réservés aux membres de l'administration, répartis par service (ministère, commissions, secrétariats, instructeurs, etc.).



**C-LogIn**

Cette interface permet aux membres de l'administration de se connecter en saisissant leur identifiant et mot de passe. Une option « Se souvenir de moi » est disponible, ainsi qu’un bouton vert « Se connecter ».



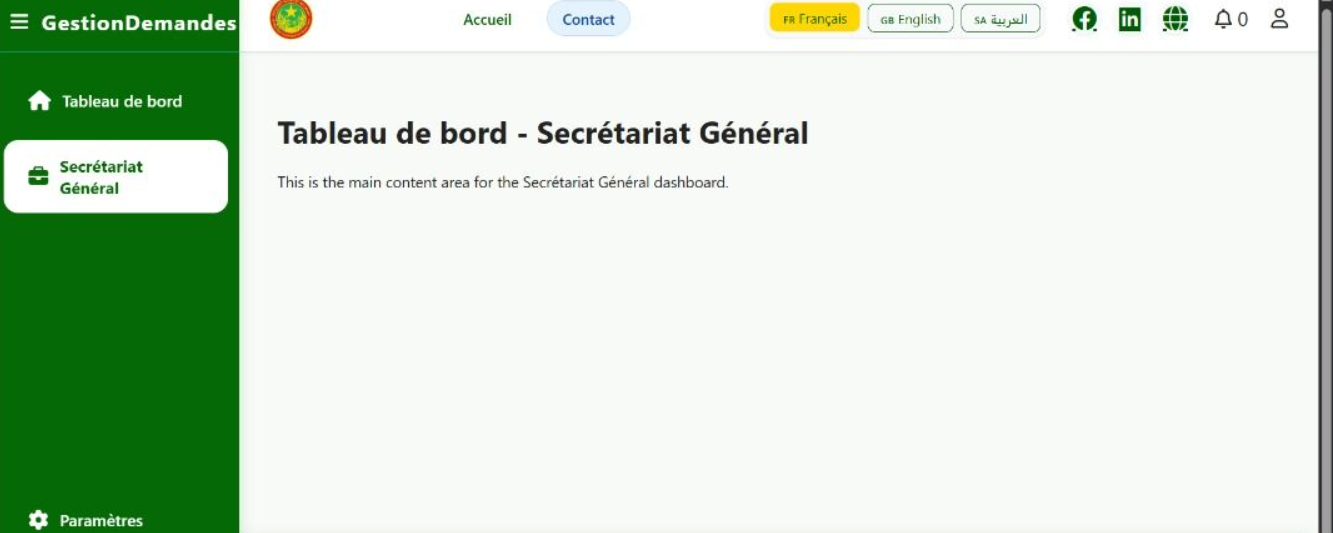
**D-Mot De Passe Oublié**

Permet à l’adminstrateur de recevoir un lien pour réinitialiser son mot de passe



**E-DashBoard Secrétariat General**

Cette page permet au Secrétariat Général de consulter et suivre l’état des demandes reçues et transmises, avec les détails de chaque dossier (référence, demandeur, date, statut, action).



F-DashBoard Secrétariat Centrale

Cette page permet au Secrétariat Central de visualiser les nouvelles demandes déposées, avec leurs détails (référence, demandeur, date, statut) et d’y accéder via le bouton « Consulter ».

