

Penguins Platform

PORTFOLIO PROJECT — 2026

Une plateforme web de découverte, recommandation et organisation collaborative d'outils d'intelligence artificielle.



TECHNOLOGIES :

React.js • PHP • MySQL • OpenRouter AI



AUTHOR: Yahya Elmaoui

CONTRIBUTORS: Aram A., Zeryouel A., El-Assali S.

Table des Matières

1. Introduction Générale	3
1.1 Contexte du Projet	3
1.2 Problématique	3
1.3 Objectifs	3
2. Spécification des Besoins	4
2.1 Identification des Acteurs	4
2.2 Diagrammes de Cas d'Utilisation	4
3. Modélisation et Conception du Système	11
3.1 Diagrammes de Classes	11
3.2 Diagrammes de Séquence	15
3.3 Diagrammes d'Activité	41
3.4 Diagrammes d'États	50
3.5 Architecture de Déploiement et Base de Données	55
4. Réalisation et Implémentation	57
4.1 Architecture et Technologies Frontend	57
4.2 Présentation Détaillée des Écrans Utilisateurs	58
4.3 Interface d'Administration	80
4.4 Architecture et Technologies Backend	91
5. Conclusion Générale et Perspectives	94

Rapport de Projet de Fin d'Études : Plateforme "Penguins"

1. Introduction Générale

1.1 Contexte du Projet

Dans un écosystème technologique en constante évolution, l'émergence rapide de nouveaux outils basés sur l'Intelligence Artificielle (IA) rend difficile la veille technologique pour les professionnels et les passionnés. La plateforme **Penguins** a été conçue pour centraliser, catégoriser et recommander ces outils de manière intelligente et communautaire.

1.2 Problématique

Comment offrir aux utilisateurs une interface intuitive permettant de découvrir, d'évaluer et de sauvegarder les meilleurs outils IA tout en garantissant un contenu modéré et de qualité via un système de suggestions collaboratif ?

1.3 Objectifs

- Développer une application web complète (Frontend interactif et Backend robuste).
- Implémenter un moteur de recherche et de recommandation d'IA.
- Mettre en place un système de bibliothèque personnelle (favoris, collections/shelves).
- Offrir une interface d'administration avancée pour la gestion des données, des utilisateurs et la modération.

2. Spécification des Besoins

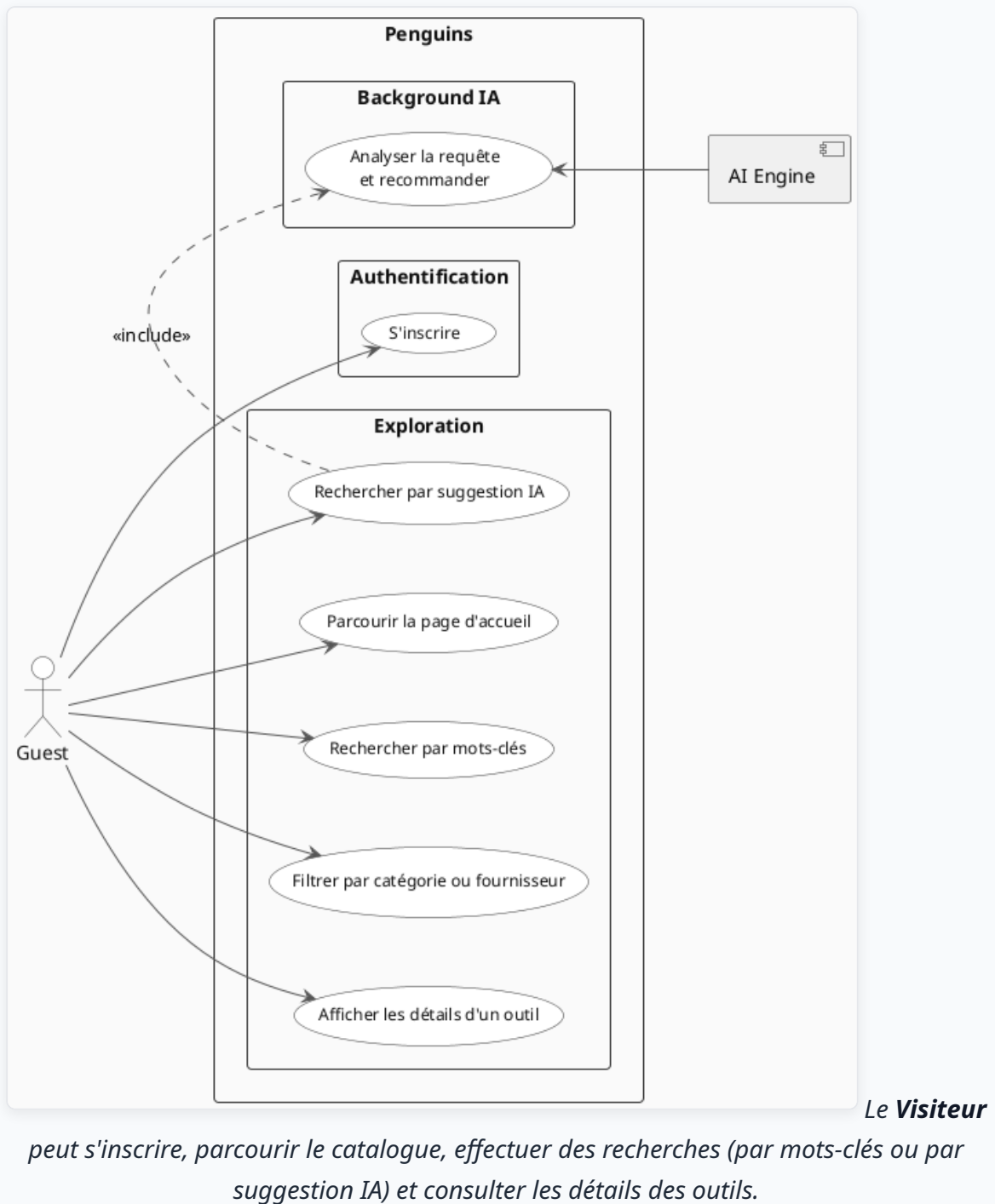
2.1 Identification des Acteurs

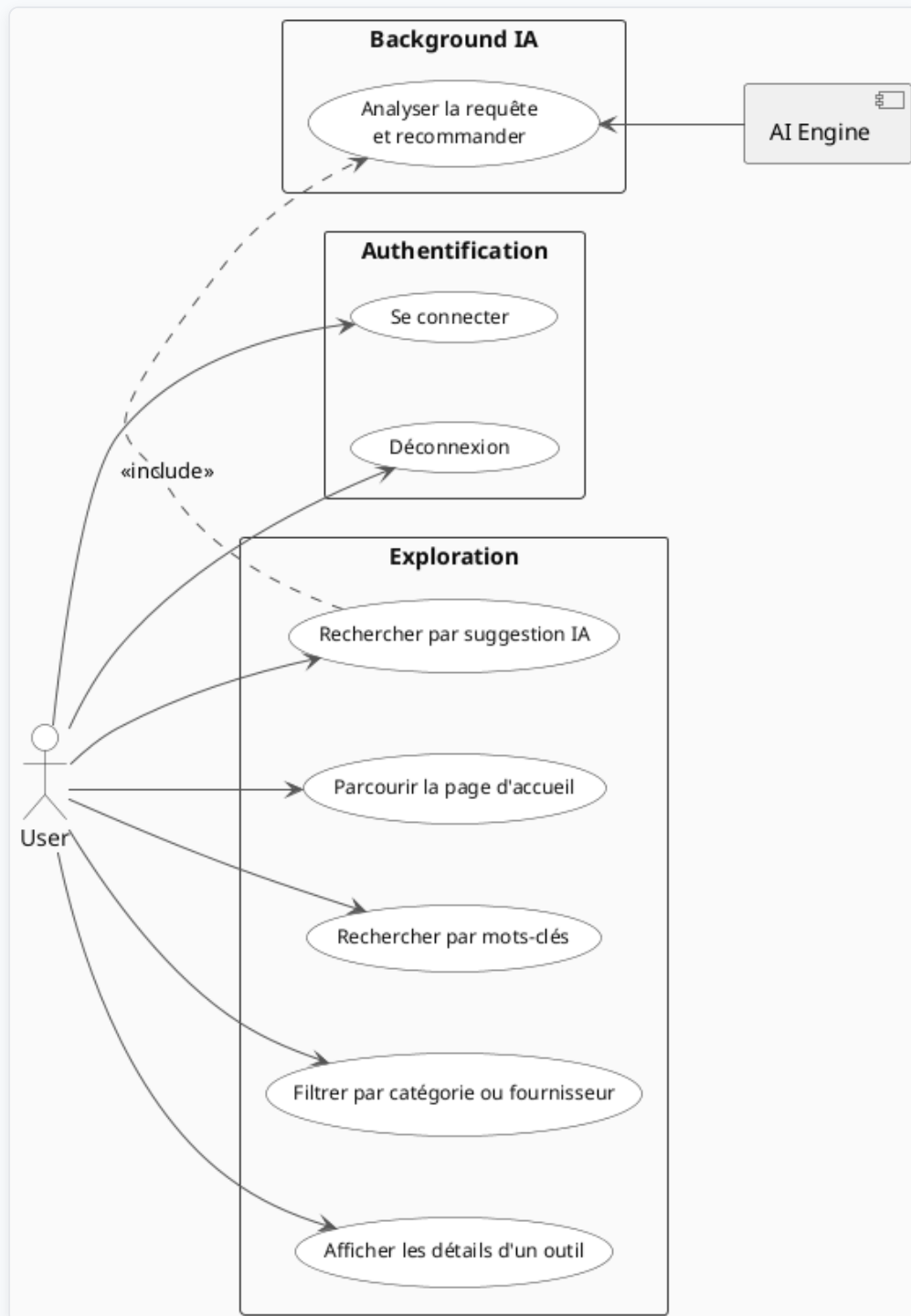
Le système interagit principalement avec quatre types d'acteurs :

1. **Visiteur** : Peut consulter la page d'accueil, rechercher des outils et voir leurs détails.
2. **Utilisateur Authentifié** : Dispose d'un profil, peut ajouter des outils à sa bibliothèque (favoris/shelves), laisser des avis et soumettre des suggestions d'outils.
3. **Manager** : Responsable de la validation des suggestions soumises par les utilisateurs.
4. **Administrateur** : Possède tous les droits, y compris la gestion des utilisateurs (bannissement), des données de référence (entités) et la modération des avis.

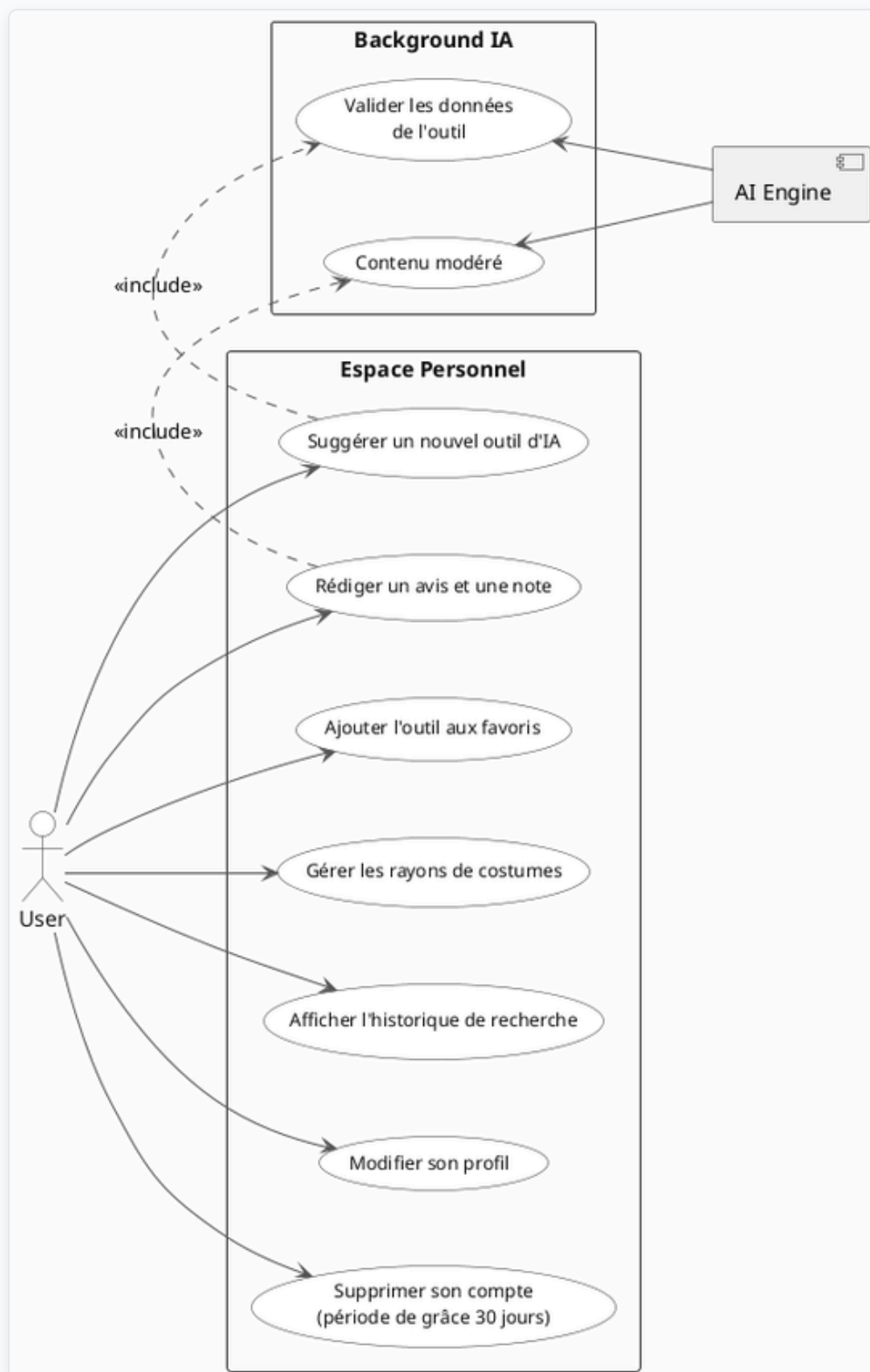
2.2 Diagrammes de Cas d'Utilisation

Les diagrammes de cas d'utilisation ci-dessous représentent les fonctionnalités offertes par la plateforme Penguins pour chaque acteur du système. On distingue les quatre profils — le **Visiteur (Guest)**, l'**Utilisateur connecté (User)**, le **Manager** et l'**Administrateur** — ainsi que leurs interactions avec le moteur d'intelligence artificielle intégré. Les relations `<<include>>` précisent les dépendances entre certains cas d'utilisation.

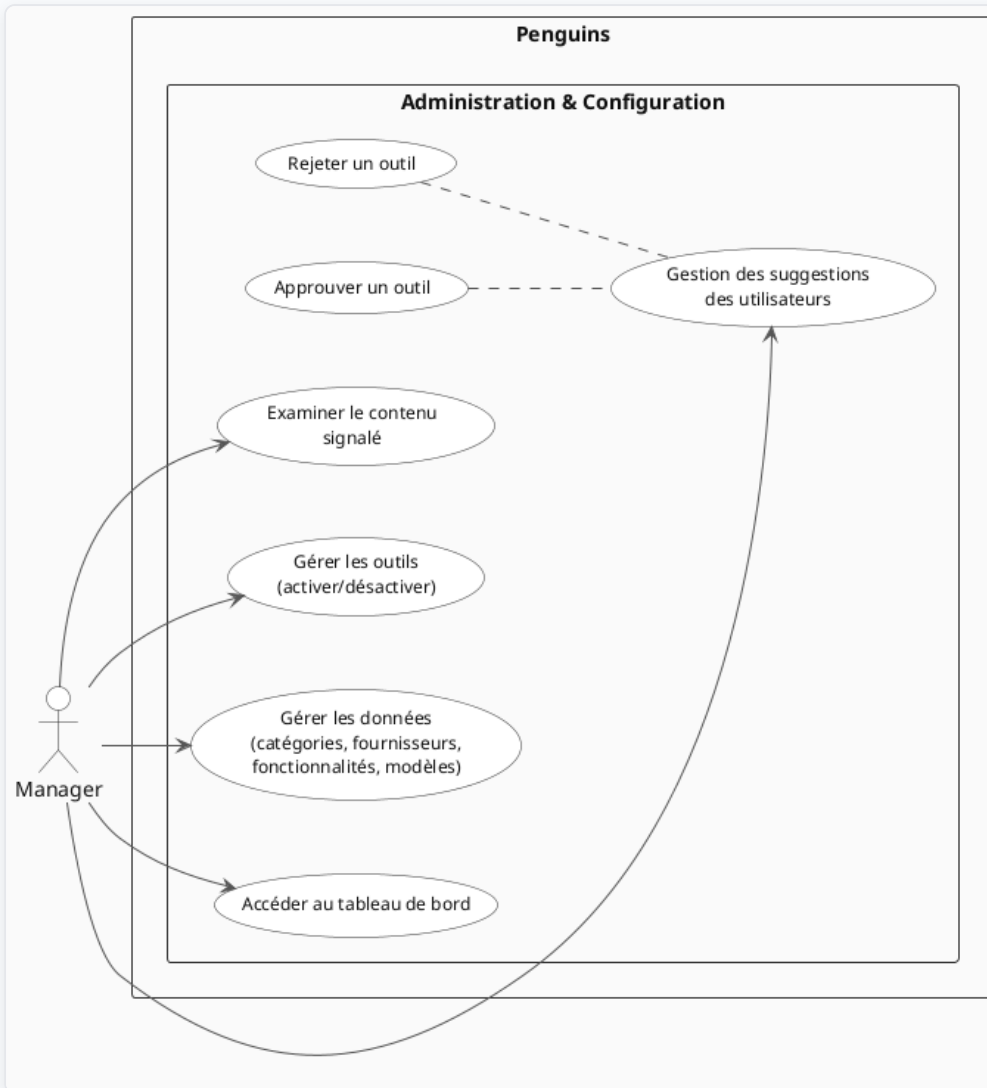




*L'**Utilisateur connecté** peut s'authentifier (connexion, déconnexion) et explorer le catalogue : parcourir la page d'accueil, rechercher par mots-clés ou par suggestion IA, filtrer et consulter les détails d'un outil. La recherche par suggestion IA s'appuie sur le moteur de recommandation.*

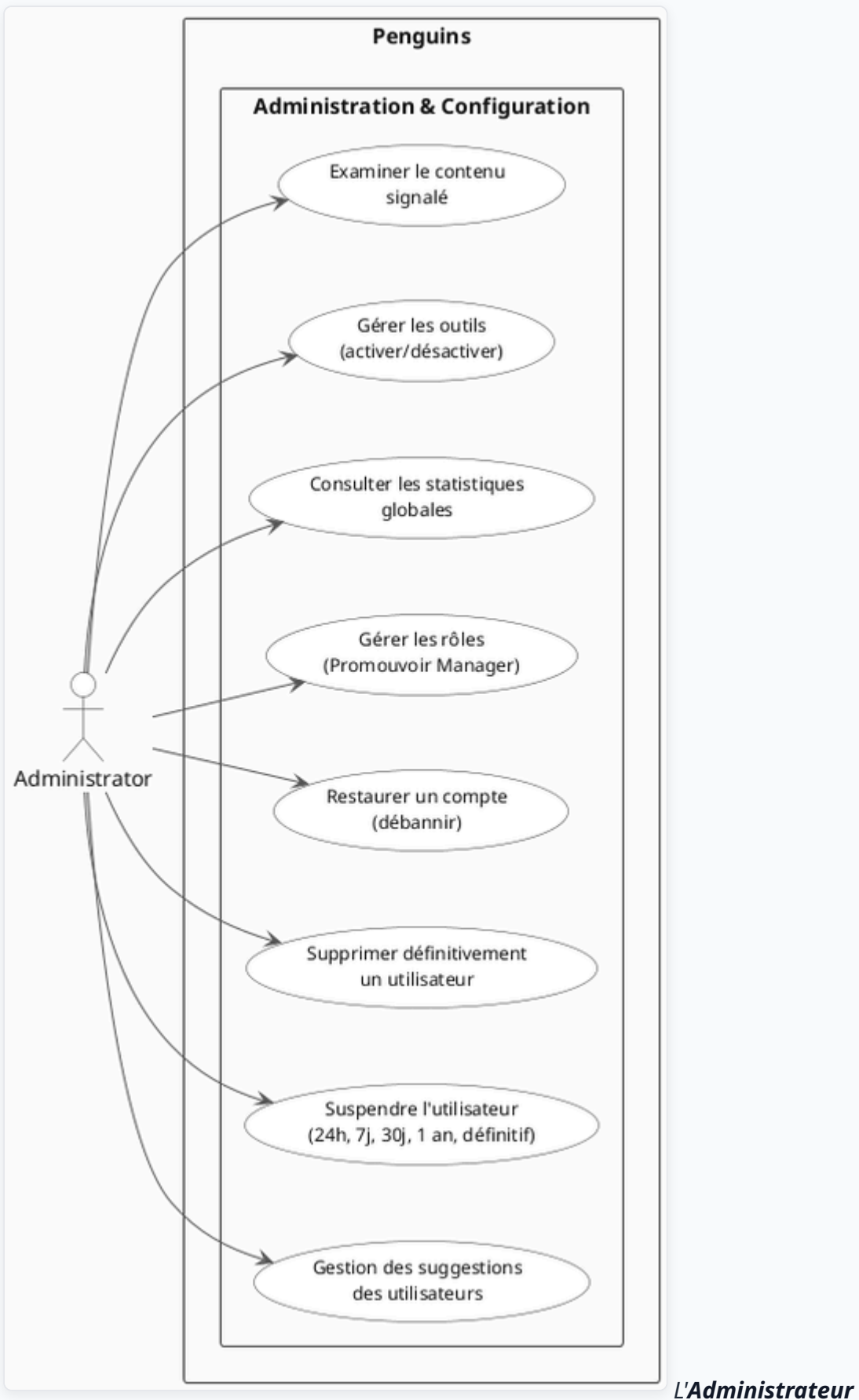


*L'**Utilisateur connecté** dispose d'un espace personnel complet : favoris, étagères, historique de recherche, avis/notes, suggestions d'outils, modification du profil et auto-suppression avec période de grâce. La soumission d'avis et de suggestions sollicite l'IA pour la modération et la validation.*



Le Manager

accède au tableau de bord, gère les suggestions des utilisateurs (approuver/rejeter), administre les données de référence, active/désactive les outils et examine les contenus signalés.



dispose de tous les privilèges : suspension et suppression d'utilisateurs, gestion des rôles, consultation des statistiques globales, et toutes les fonctions de gestion du Manager.

3. Modélisation et Conception du Système

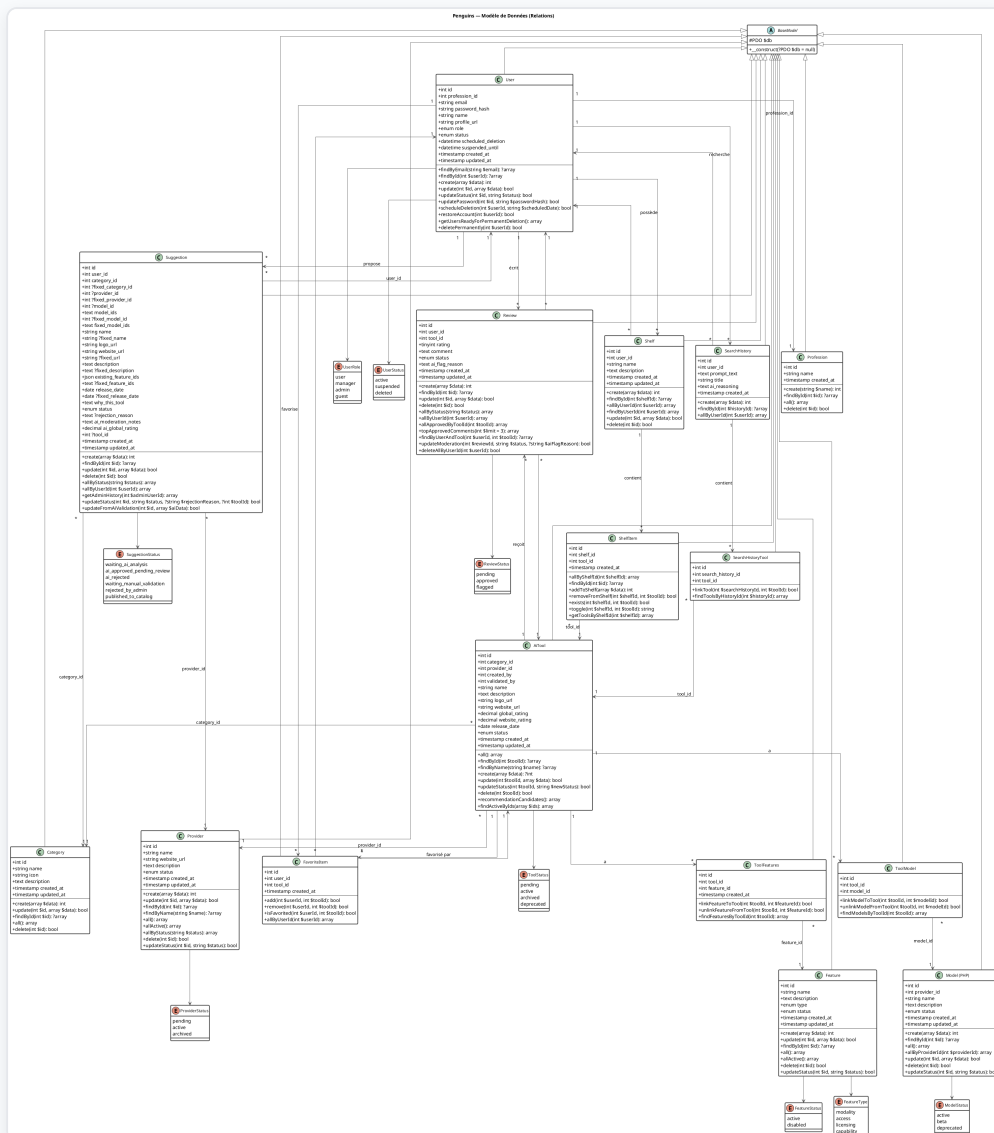
Cette section présente l'ensemble des diagrammes UML qui modélisent la plateforme Penguins. Chaque diagramme est accompagné d'une explication détaillée de son rôle et de ce qu'il illustre.

3.1 Diagrammes de Classes

Les diagrammes de classes décrivent la structure statique du backend en séparant les responsabilités selon le patron MVC. Quatre diagrammes distincts couvrent respectivement les modèles (entités et relations), les contrôleurs (points d'entrée API), les services (logique métier) et les classes de support (middleware, utilitaires, configuration).

3.1.1 Classes — Modèles (Entités et Relations)

Ce diagramme présente l'ensemble des classes du dossier `Models/` qui héritent toutes de la classe abstraite `BaseModel`. Celle-ci fournit la connexion PDO à la base de données MySQL. On distingue les entités principales (`User` , `AiTool` , `Review` , `Suggestion`), les tables de référence (`Category` , `Provider` , `Feature` , `Model` , `Profession`), les données utilisateur (`FavoriteItem` , `Shelf` , `ShelfItem` , `SearchHistory` , `SearchHistoryTool`) et les tables de jonction (`ToolFeatures` , `ToolModel`). Les cardinalités (`1 → *` , `* → 1`) précisent les relations entre ces entités.

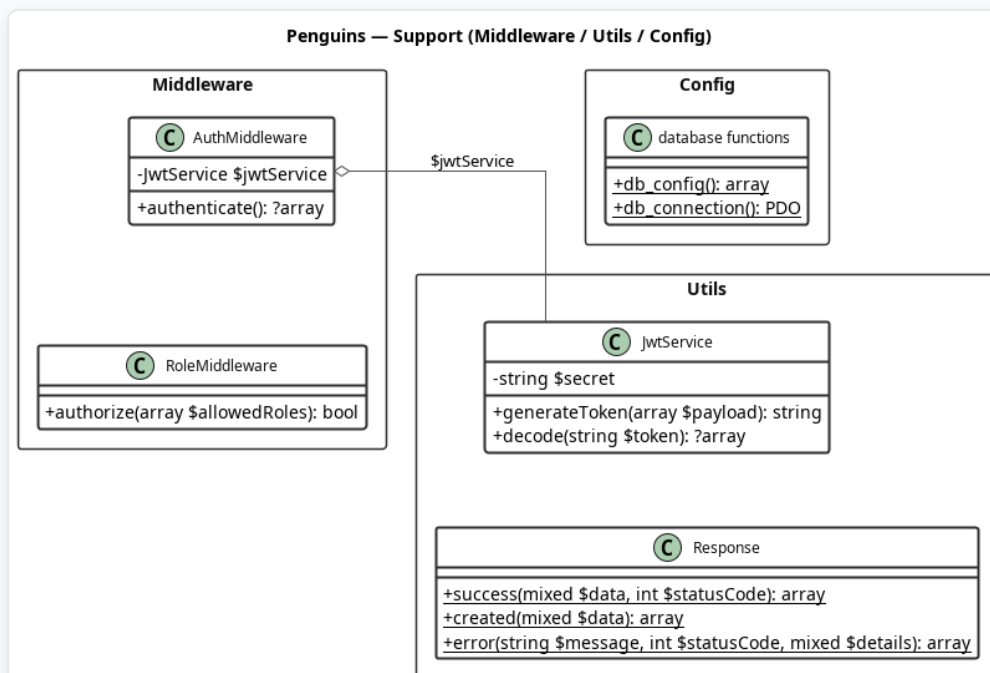


3.1.2 Classes — Contrôleurs (Points d'Entrée API)

Ce diagramme illustre les dix contrôleurs qui constituent l'interface RESTful du backend. Chaque contrôleur est responsable d'un domaine fonctionnel : **AuthController** (authentification), **UserController** (profil), **AdminController** (administration complète), **ExploreController** (recherche et exploration), **ToolController** (outils et avis), **FavoriteController** (favoris), **ReviewController** (avis utilisateurs), **ShelfController** (étagères personnelles), **SearchHistoryController** (historique de recherche) et **SuggestionController** (suggestions d'outils). Les dépendances vers les services et modèles sont représentées par des relations d'agrégation.



Page 13 / 94

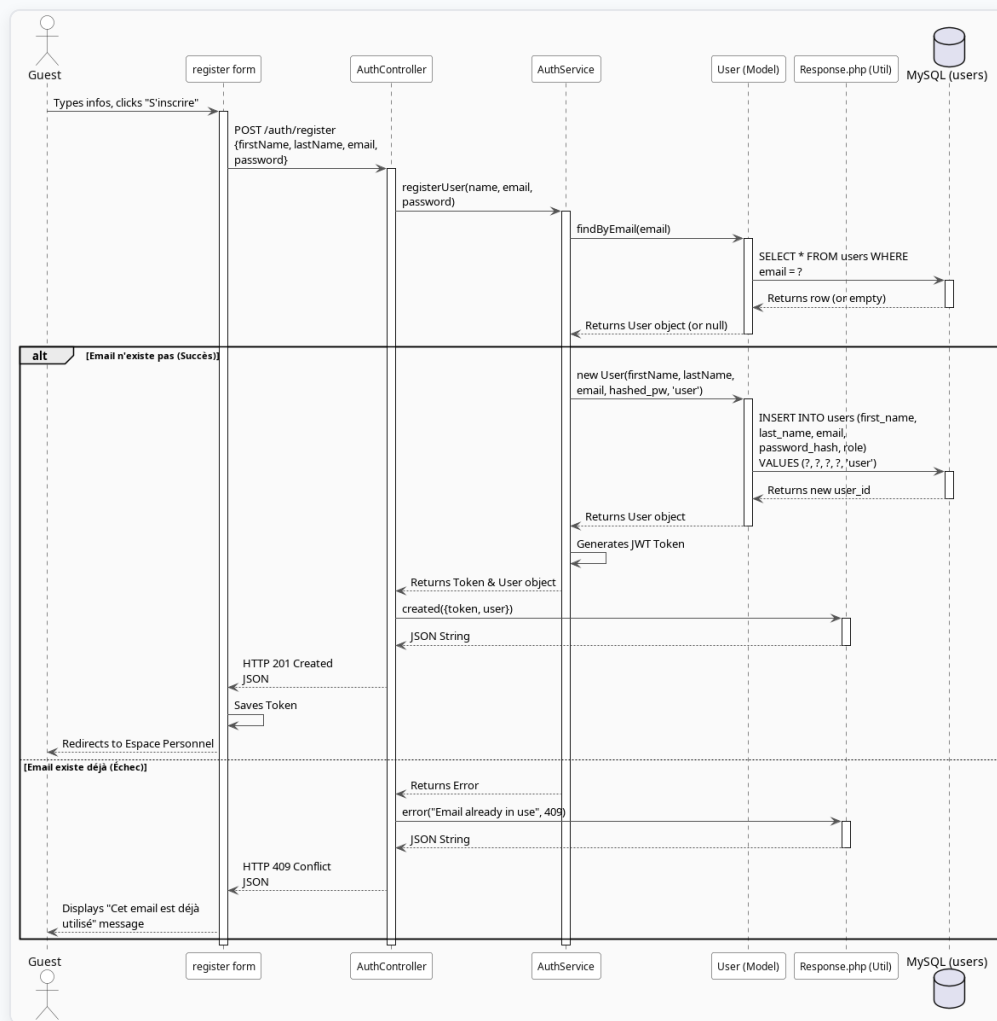


3.2 Diagrammes de Séquence

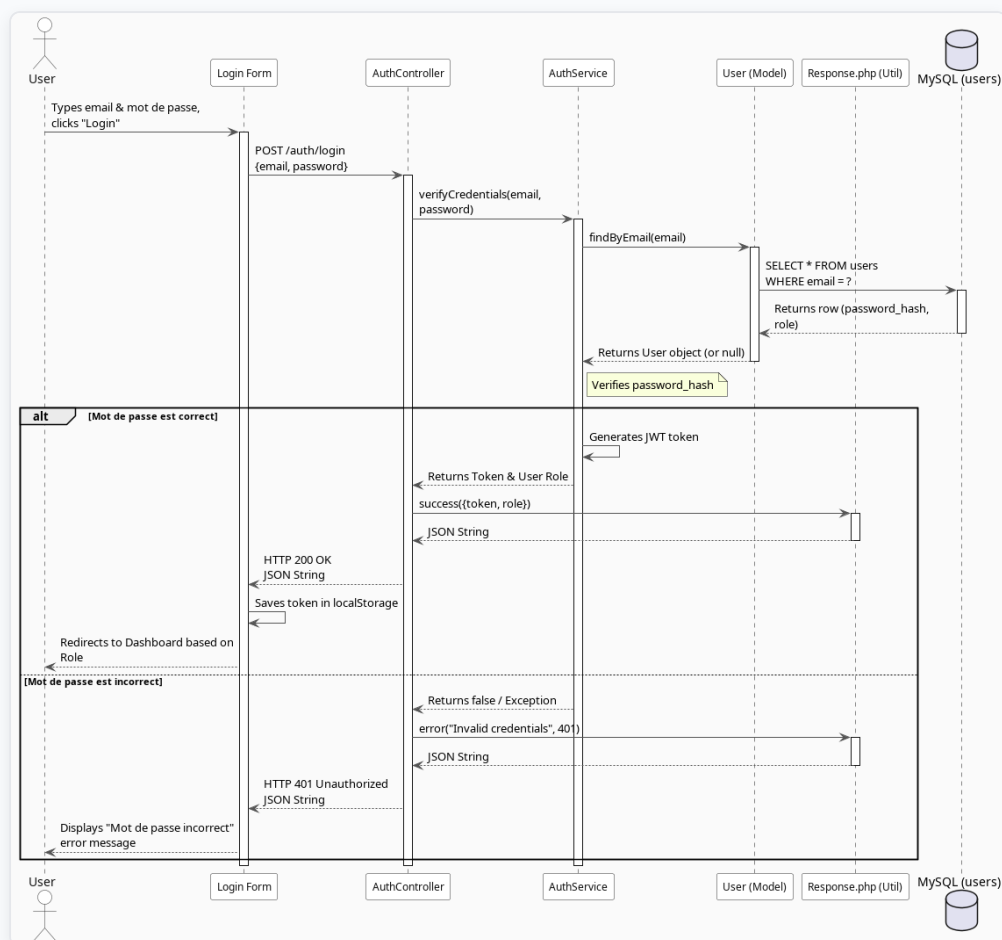
Les diagrammes de séquence illustrent l'ordre chronologique des échanges entre les différents composants du système (Frontend React, Contrôleurs PHP, Services, Modèles et Base de données) pour réaliser une fonctionnalité spécifique.

3.2.1 Authentification et Gestion de Compte

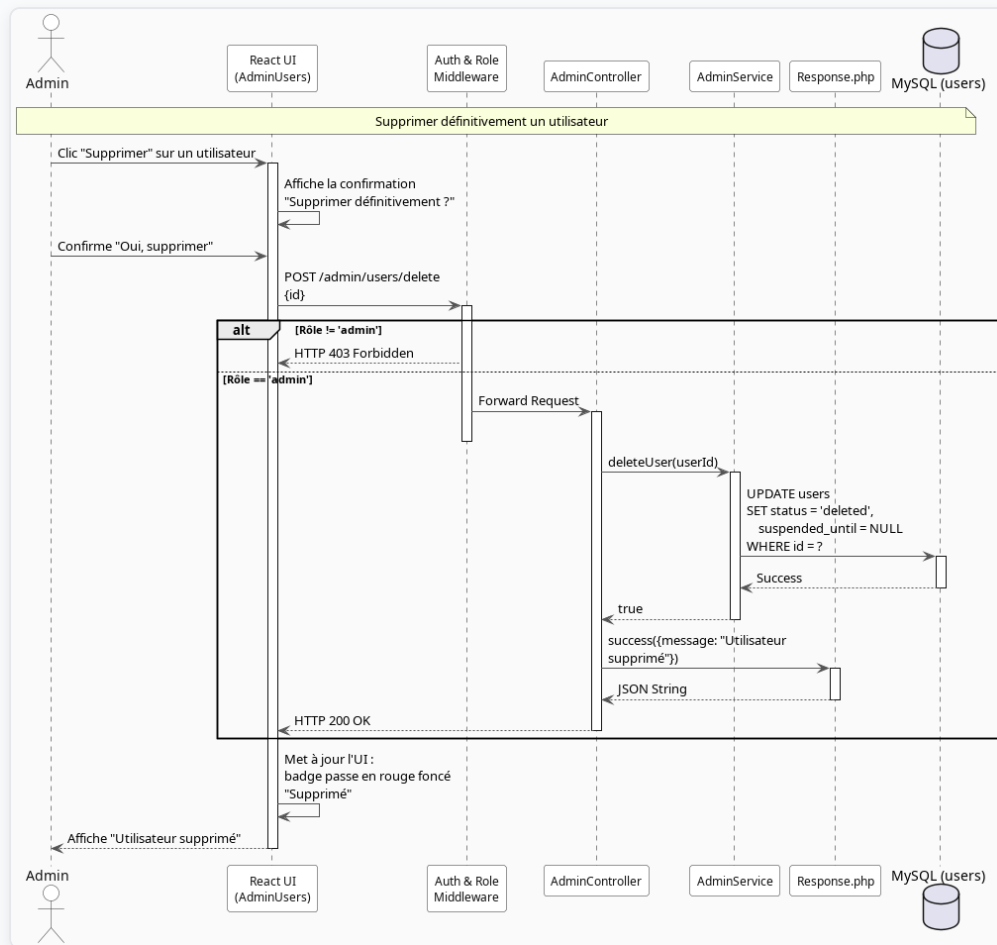
Séquence — Inscription (S'inscrire) : Ce diagramme détaille le processus d'inscription. L'utilisateur remplit le formulaire d'inscription (nom, email, mot de passe). Le `AuthController` reçoit la requête POST et délègue à `AuthService::registerUser()`. Le service vérifie d'abord si l'email n'est pas déjà utilisé via `User::findByEmail()`. Si l'email est libre, il hache le mot de passe avec BCrypt, crée l'utilisateur via `User::create()`, génère un token JWT et retourne une réponse HTTP 201 avec le token. Si l'email existe déjà, une erreur HTTP 409 est renvoyée.



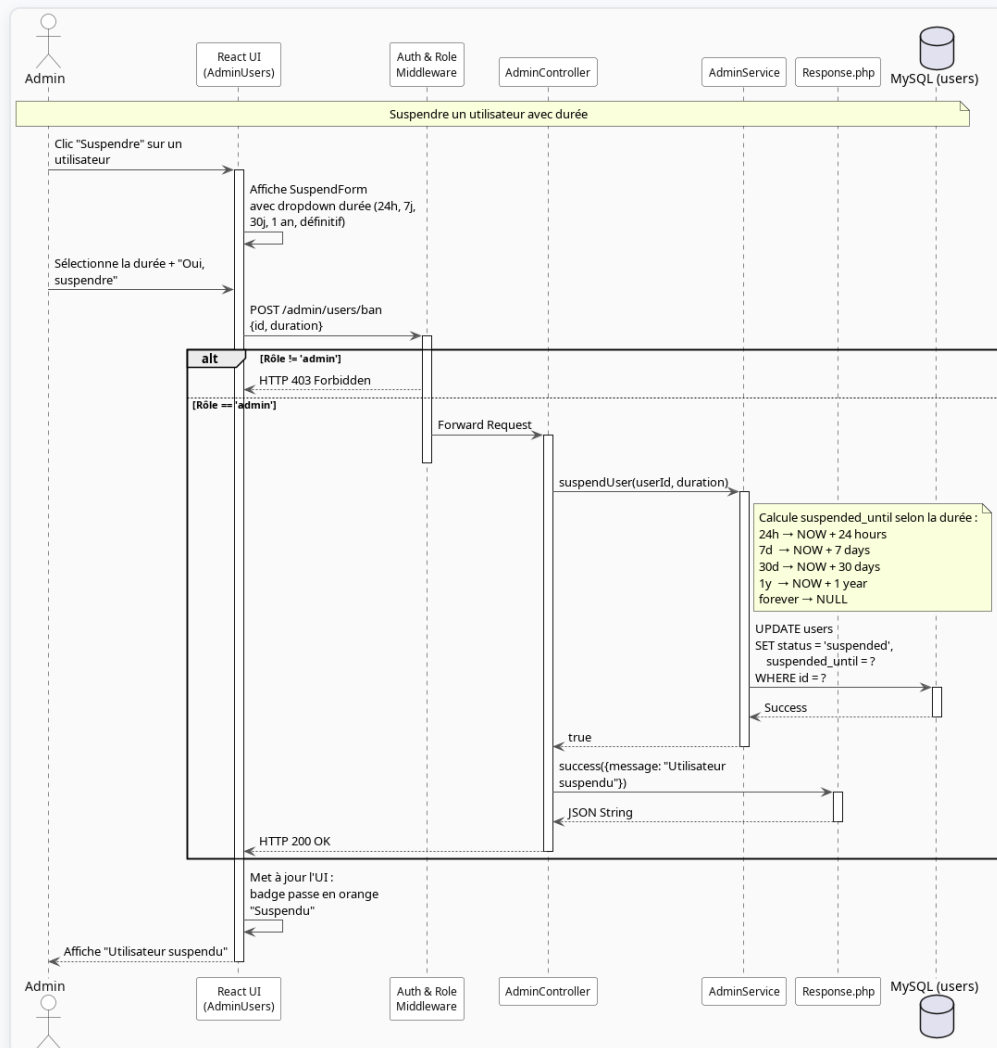
Séquence — Connexion (Se connecter) : Ce diagramme montre le flux de connexion. L'utilisateur saisit son email et mot de passe. `AuthService::verifyCredentials()` recherche l'utilisateur par email, puis vérifie le mot de passe avec `password_verify()`. Si les identifiants sont valides, un token JWT est généré et retourné (HTTP 200). Sinon, une erreur HTTP 401 est renvoyée. Le frontend stocke ensuite le token dans le `localStorage` via `AuthContext.jsx`.



Séquence — Suppression définitive par l'Admin : Ce diagramme illustre la suppression permanente d'un utilisateur par l'administrateur. Après vérification du rôle admin via `RoleMiddleware`, le `AdminController` appelle `AdminService::deleteUser()` qui met à jour le statut de l'utilisateur en 'deleted' et efface `suspended_until`. L'interface React met alors à jour l'affichage du badge en rouge foncé "Supprimé".

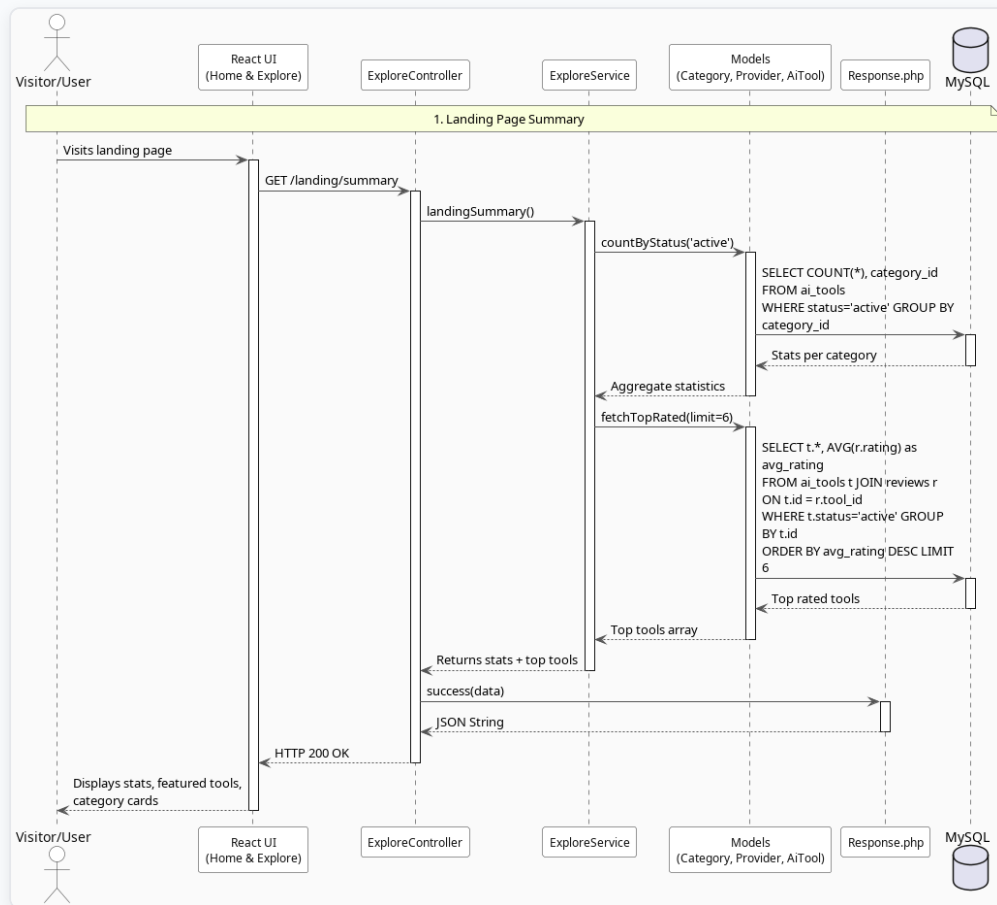


Séquence — Suspension d'un utilisateur avec durée : L'administrateur sélectionne une durée de suspension (24h, 7j, 30j, 1 an ou définitif) via un formulaire dédié. `AdminService::suspendUser()` calcule la date `suspended_until` en fonction de la durée choisie et met à jour le statut de l'utilisateur en 'suspended'. Un badge orange "Suspendu" apparaît dans l'interface.

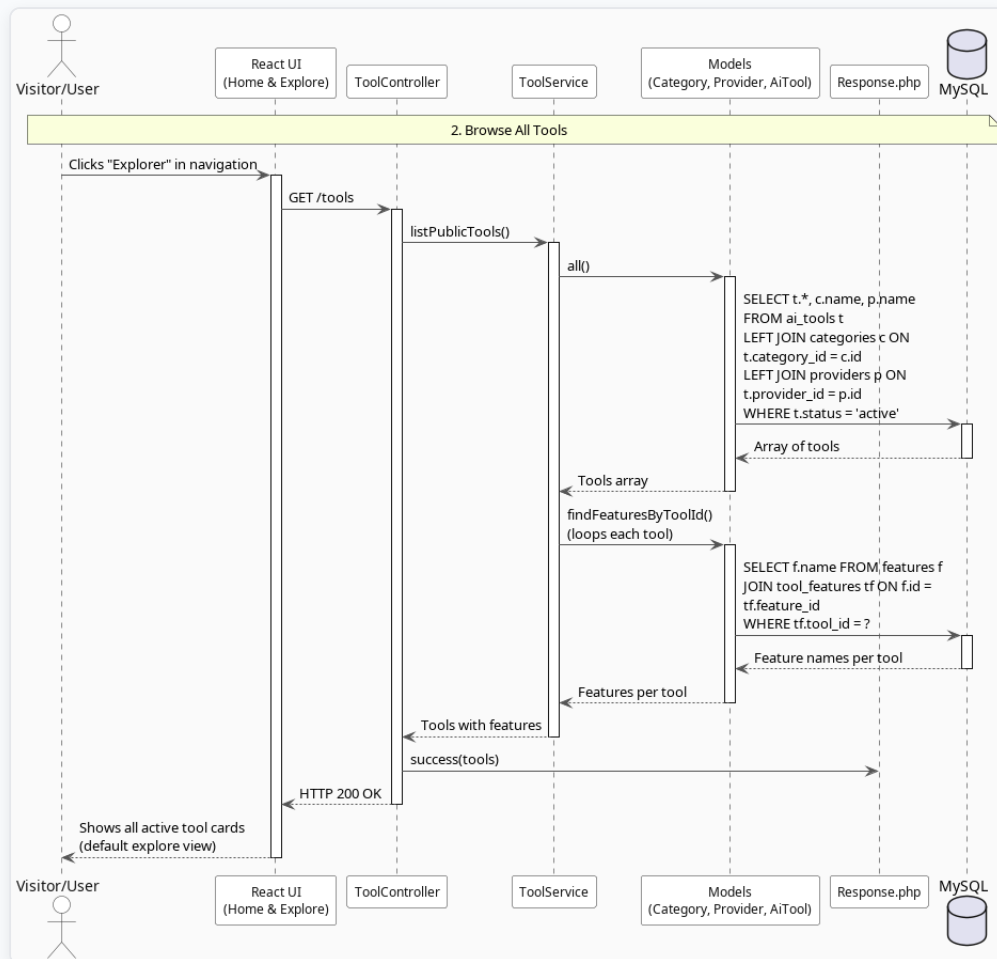


3.2.2 Exploration et Navigation

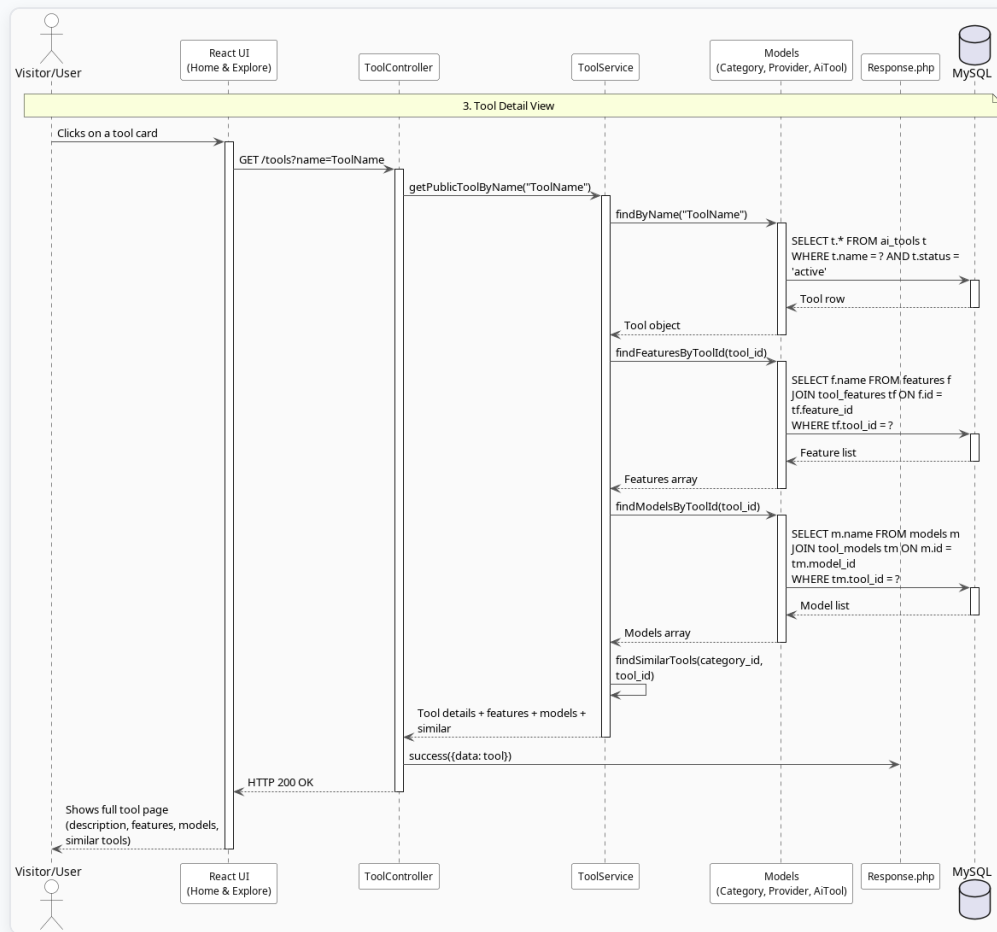
Séquence — Résumé de la Page d'Accueil (Landing Page) : Lorsqu'un visiteur ou utilisateur arrive sur la page d'accueil, le frontend appelle `GET /landing/summary`. L'`ExploreService` interroge la base de données pour compter les outils actifs par catégorie et récupère les 6 outils les mieux notés (via `COUNT` et `AVG(rating)` avec `JOIN` sur les reviews). Ces données sont renvoyées sous forme de statistiques et de cartes d'outils "Top 6" affichées dynamiquement.



Séquence — Parcourir tous les outils : L'utilisateur clique sur "Explorer" dans la navigation. `ToolService::listPublicTools()` exécute une requête qui sélectionne tous les outils actifs avec leurs catégories et fournisseurs (via LEFT JOIN), puis pour chaque outil, récupère ses fonctionnalités associées via `ToolFeatures::findFeaturesByToolId()`. La liste complète est affichée sous forme de cartes d'outils.

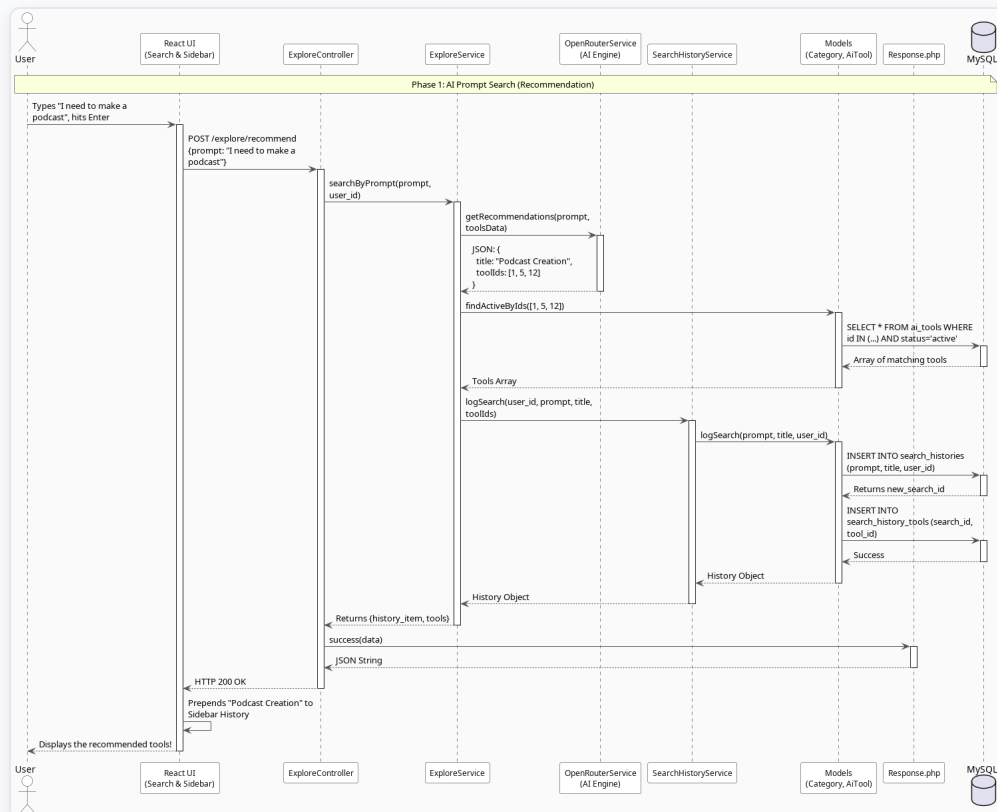


Séquence — Détail d'un outil : L'utilisateur clique sur une carte d'outil. `ToolService::getPublicToolByName()` récupère l'outil avec ses détails complets, ses fonctionnalités (`findFeaturesByToolId()`), ses modèles (`findModelsByToolId()`) et les outils similaires. La page affiche alors l'en-tête, les caractéristiques, la section des avis et les outils recommandés similaires.

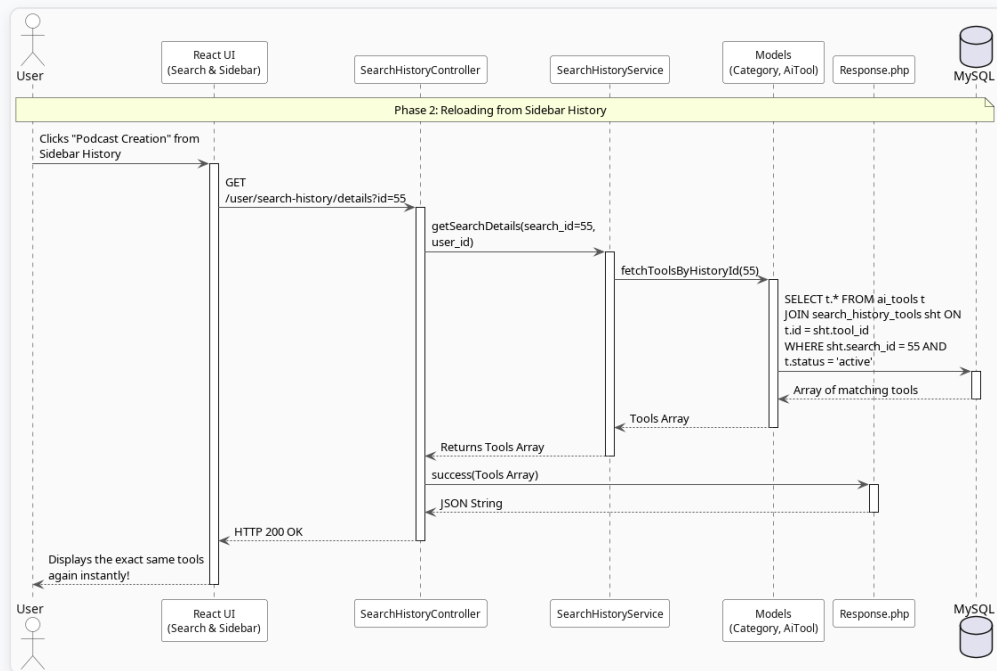


3.2.3 Recherche Intelligente par IA

Phase 1 — Recherche par Prompt et Recommandation IA : L'utilisateur saisit une requête en langage naturel (ex: "I need to make a podcast"). Le frontend envoie `POST /explore/recommend` avec le prompt. `ExploreService::searchByPrompt()` appelle `OpenRouterService::getRecommendations()` qui transmet le prompt et les candidats (outils actifs) à l'API d'IA externe (Claude/GLM). L'IA retourne un titre et une liste d'IDs d'outils pertinents. `SearchHistoryService::logSearch()` enregistre ensuite cette recherche dans l'historique pour l'utilisateur connecté. Les outils recommandés sont affichés avec le titre et le raisonnement de l'IA.

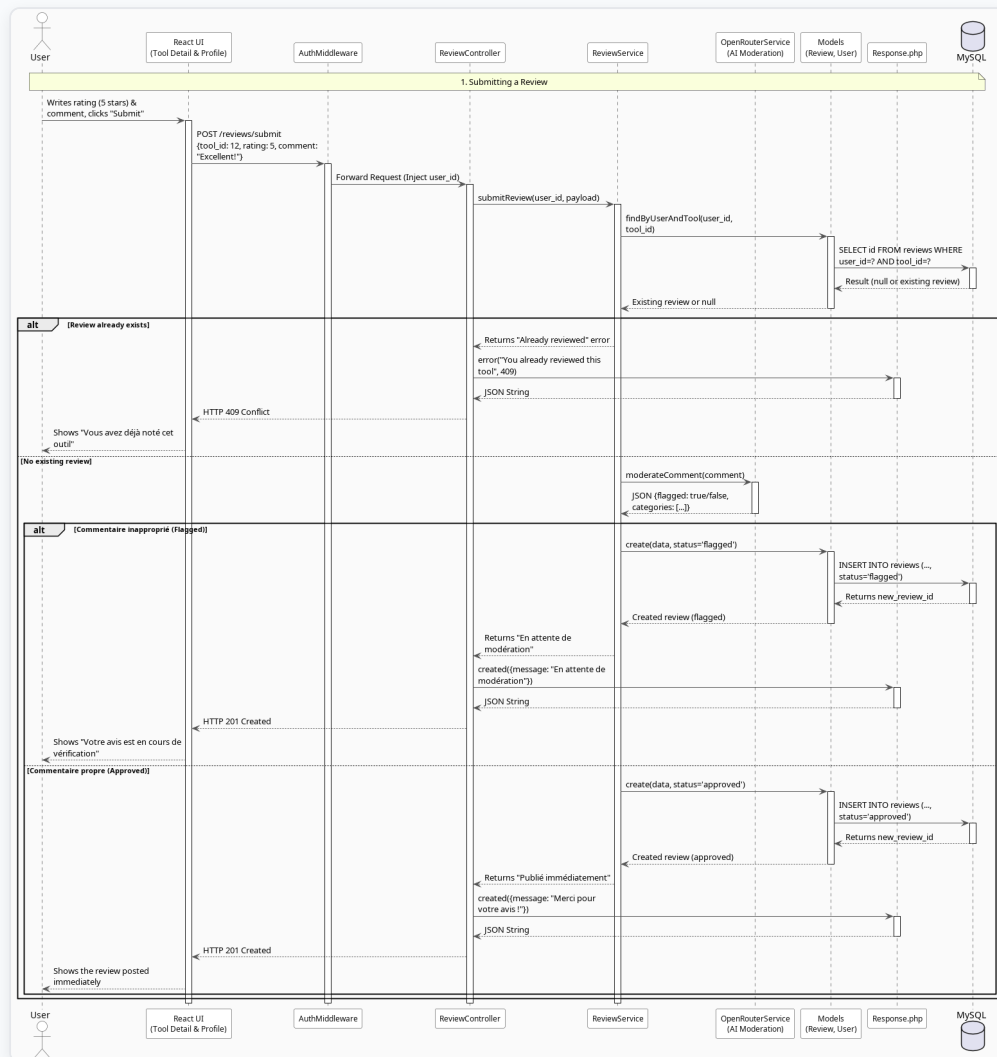


Phase 2 — Rechargement depuis l'Histoire : L'utilisateur clique sur un élément de son historique dans la sidebar. `SearchHistoryController::details()` récupère les outils associés à cette recherche via `SearchHistoryTool::findToolsByHistoryId()`. Si les données sont en cache côté frontend, elles sont affichées instantanément ; sinon, une requête API est effectuée. Cette fonctionnalité permet de retrouver rapidement les résultats d'une recherche antérieure sans repasser par l'IA.

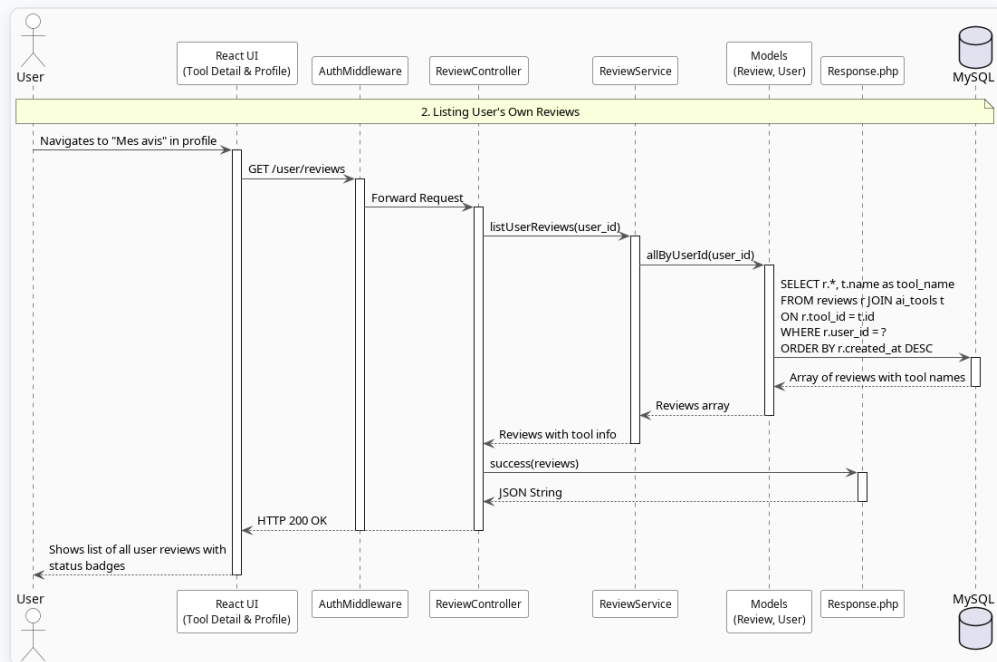


3.2.4 Gestion des Avis (Reviews)

Séquence — Soumission d'un avis : L'utilisateur note un outil (1-5 étoiles) et rédige un commentaire. `ReviewService::submitReview()` vérifie d'abord que l'utilisateur n'a pas déjà noté cet outil (évite les doublons). Ensuite, `ModerationService` (via `OpenRouterService`) analyse le commentaire. Si l'IA détecte un contenu toxique ou du spam, l'avis est marqué 'flagged' et mis en attente de modération. Sinon, il est immédiatement approuvé et visible publiquement.

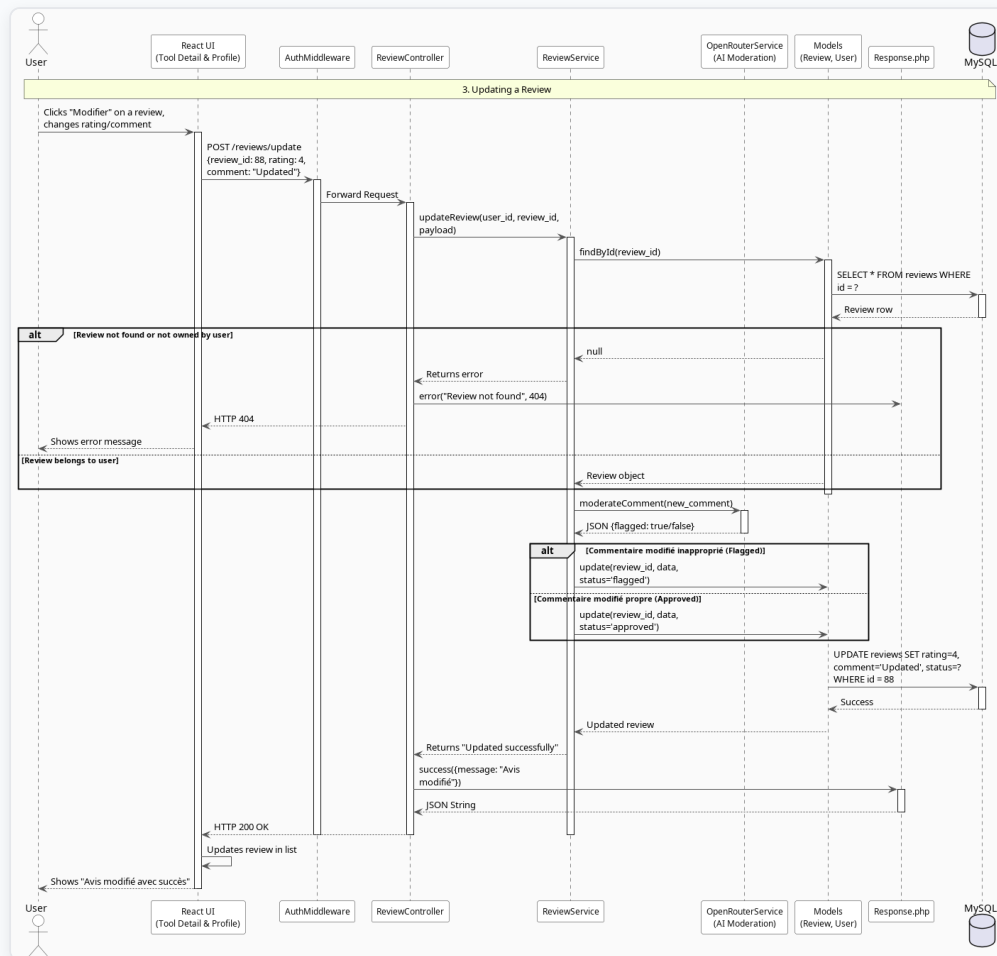


Séquence — Liste des avis de l'utilisateur : L'utilisateur navigue vers "Mes avis" dans son profil. `ReviewService::listUserReviews()` exécute une requête avec JOIN sur la table `ai_tools` pour récupérer tous les avis de l'utilisateur avec le nom de l'outil associé, triés par date de création décroissante. Chaque avis affiche son statut (approuvé, en modération).

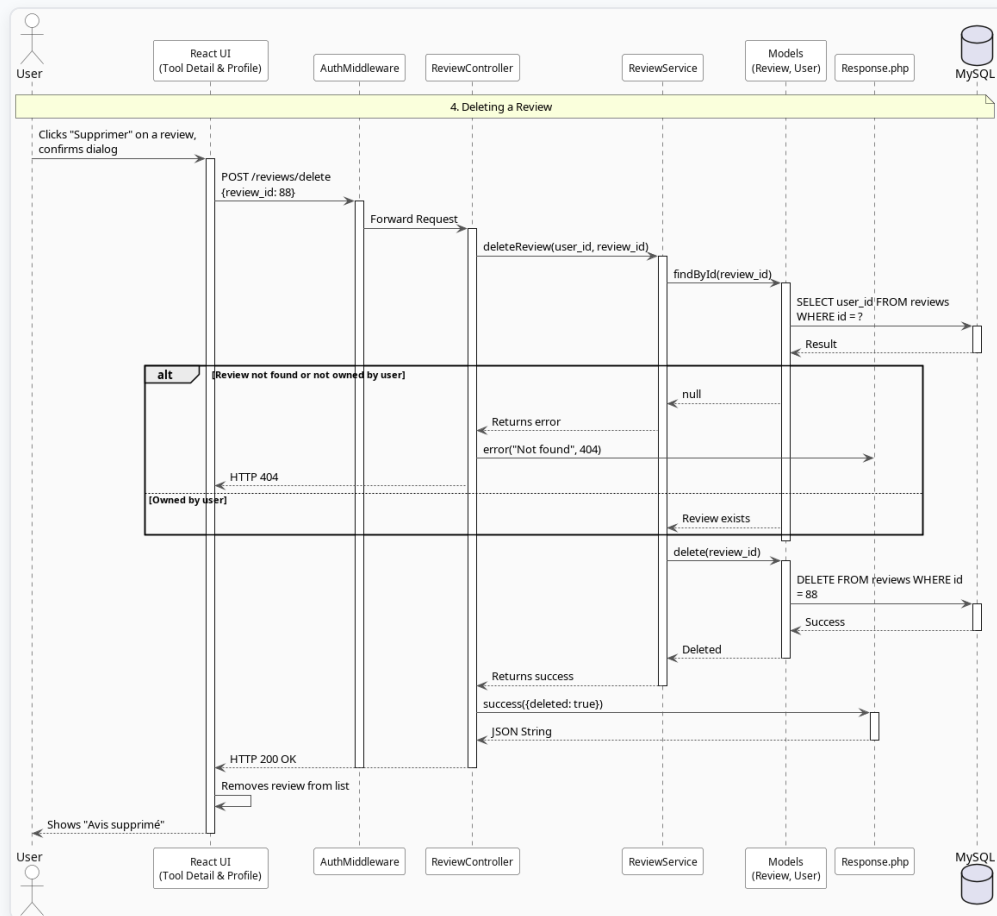


Séquence — Modification d'un avis : L'utilisateur modifie sa note ou son commentaire.

`ReviewService::updateReview()` vérifie d'abord que l'avis appartient bien à l'utilisateur, puis soumet le nouveau commentaire à la modération IA. Si le nouveau commentaire est jugé inapproprié, le statut repasse à 'flagged'. Sinon, il reste 'approved'. L'avis modifié est mis à jour en base de données.

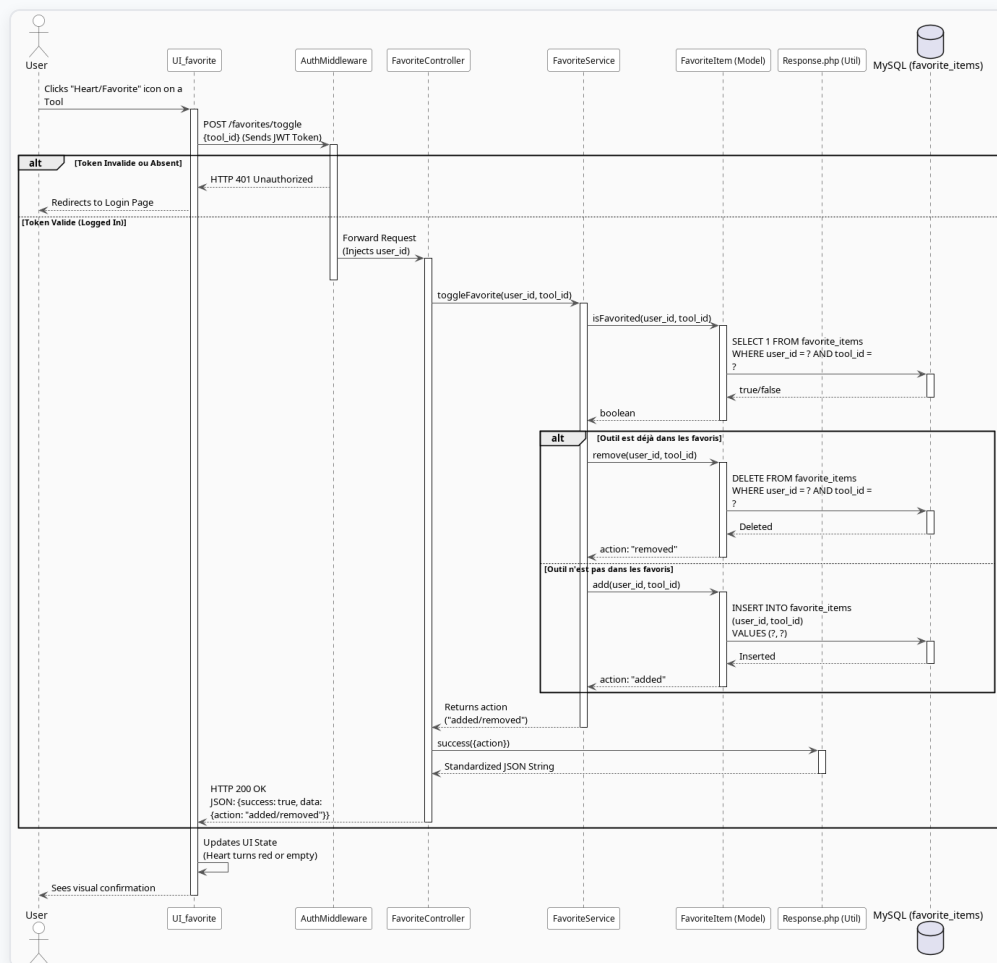


Séquence — Suppression d'un avis : L'utilisateur clique sur "Supprimer" et confirme la suppression. `ReviewService::deleteReview()` vérifie la propriété de l'avis, puis supprime définitivement l'enregistrement de la table `reviews`. L'avis disparaît instantanément de l'interface.

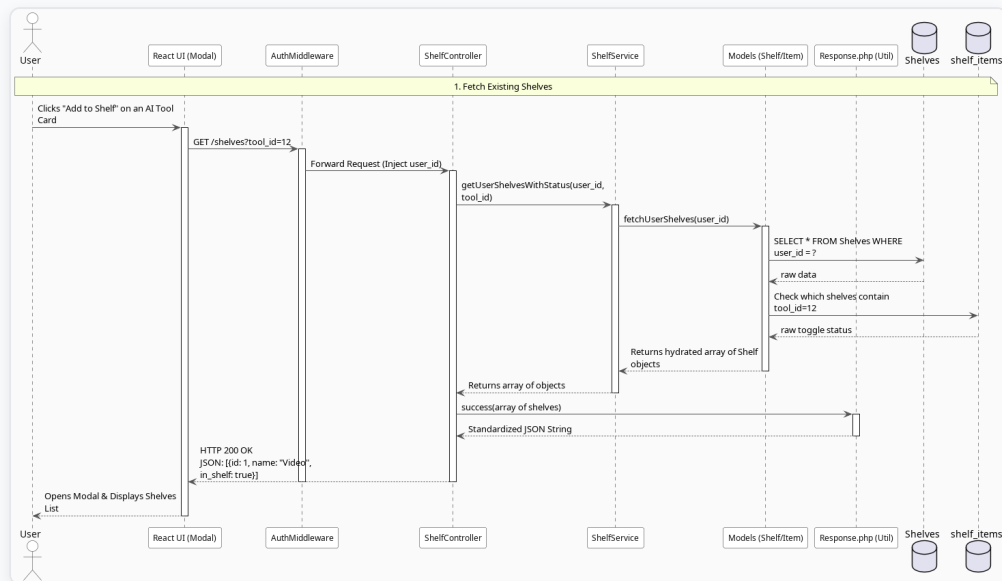


3.2.5 Gestion des Favoris et Étagères (Shelves)

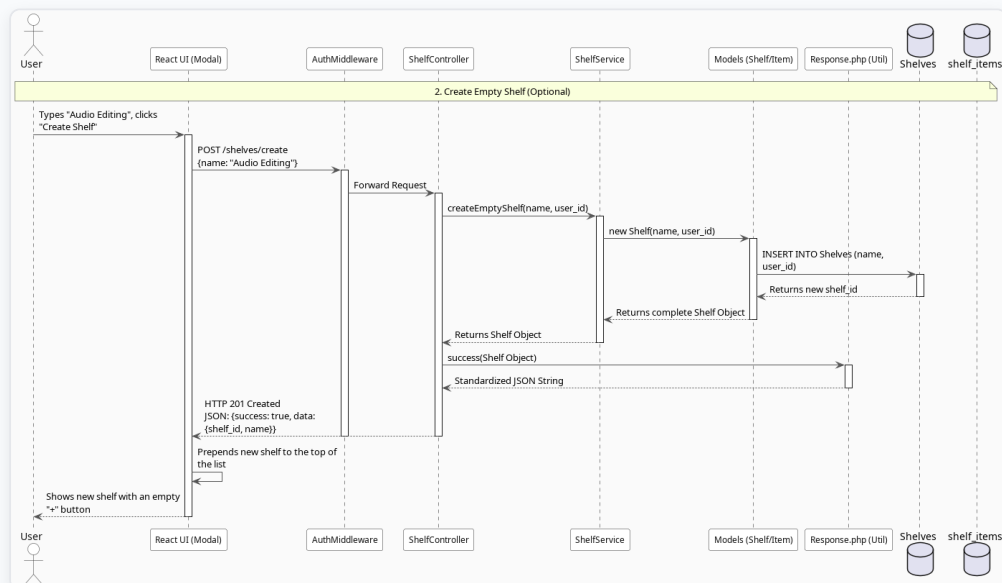
Séquence — Ajout/Retrait des Favoris : L'utilisateur clique sur l'icône "Coeur" d'un outil. `FavoriteService::toggleFavorite()` vérifie si l'outil est déjà dans les favoris via `FavoriteItem::isFavorited()`. Si oui, il le retire (DELETE) ; sinon, il l'ajoute (INSERT). La réponse retourne l'action effectuée ("added"/"removed") et l'interface React met à jour l'icône (cœur rouge ou vide).



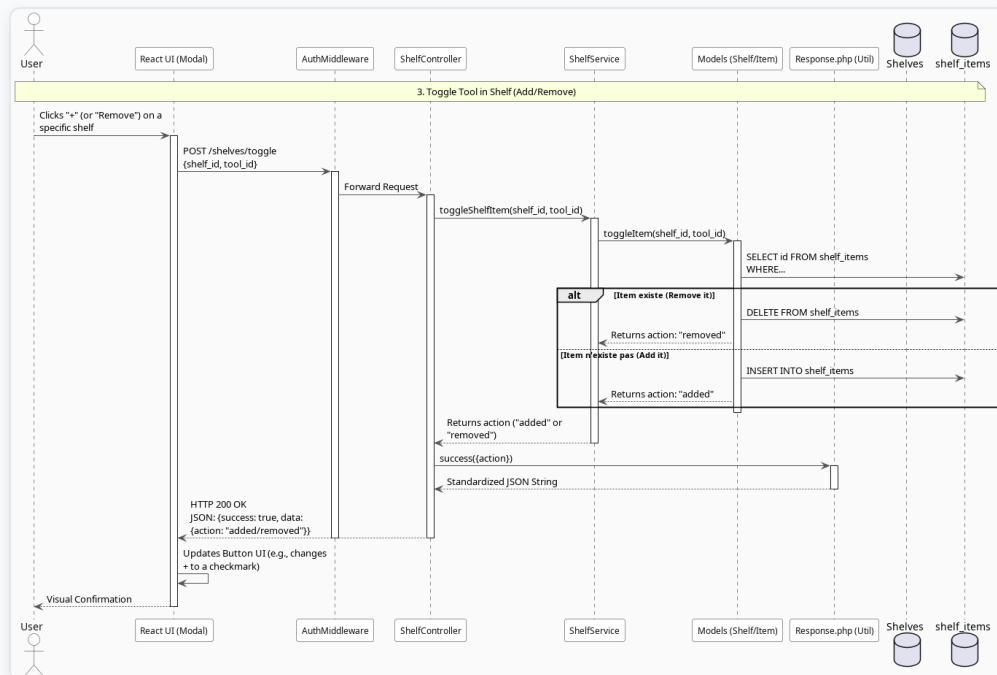
Séquence — Récupération des étagères existantes : L'utilisateur clique sur "Ajouter à une étagère" sur une carte d'outil. `ShelfService::getUserShelvesWithStatus()` récupère toutes les étagères de l'utilisateur et vérifie pour chacune si l'outil sélectionné y est déjà présent. Une modale s'ouvre affichant la liste des étagères avec leur statut (outil présent ou non).



Séquence — Création d'une nouvelle étagère : L'utilisateur saisit un nom (ex: "Audio Editing") et clique sur "Créer l'étagère". `ShelfService::createEmptyShelf()` insère un nouvel enregistrement dans la table `shelves` avec le nom et l'ID utilisateur. La nouvelle étagère apparaît immédiatement en haut de la liste dans la modale.

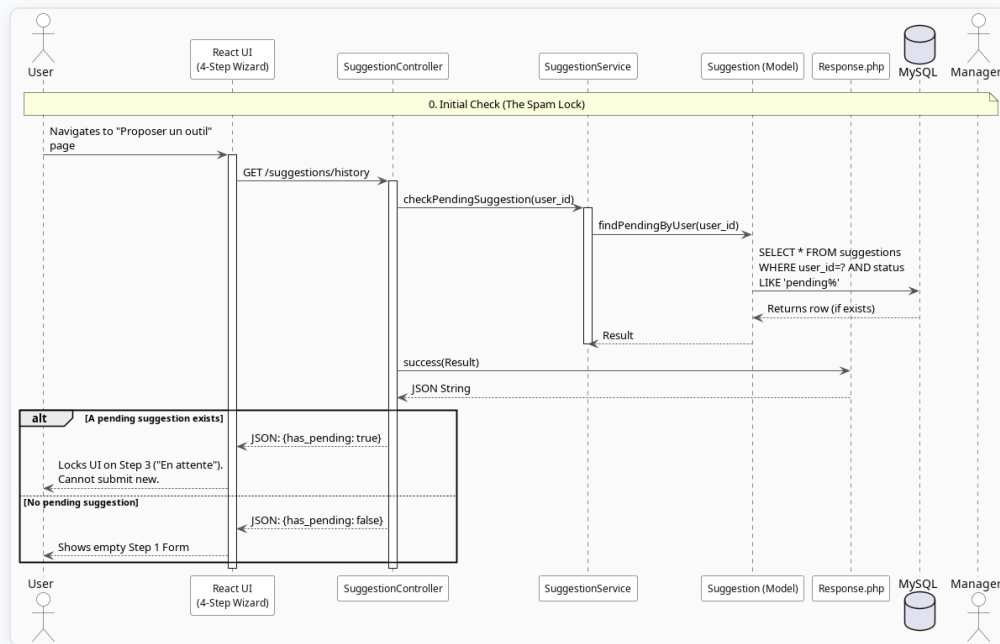


Séquence — Ajout/Retrait d'un outil dans une étagère : L'utilisateur clique sur "+" (ou "Retirer") pour une étagère spécifique. `ShelfService::toggleShelfItem()` vérifie si l'outil est déjà présent dans l'étagère. Si oui, il le supprime (DELETE) ; sinon, il l'ajoute (INSERT). Le bouton dans l'interface se met à jour visuellement (coche verte ou "+").

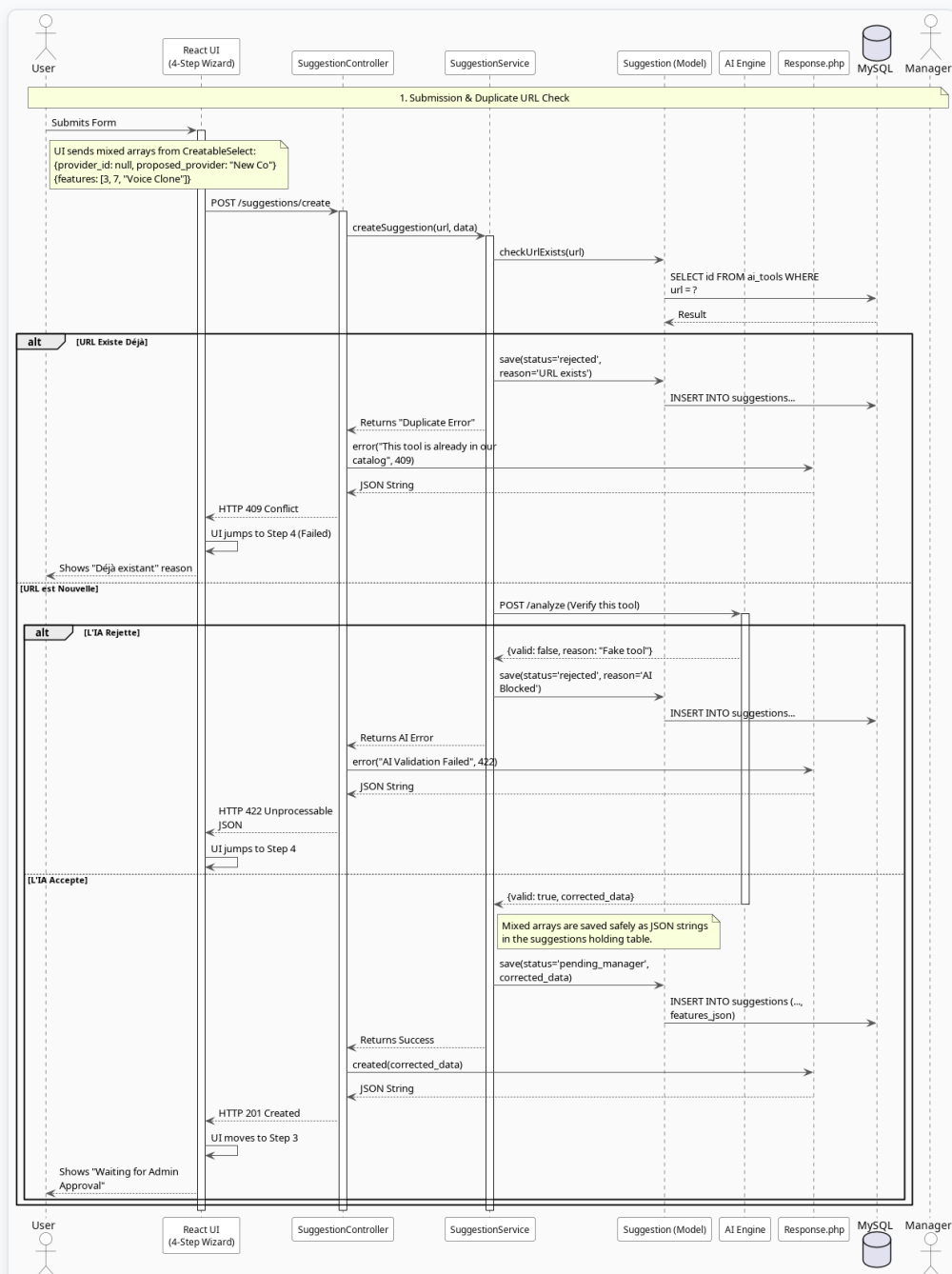


3.2.6 Processus de Suggestion d'Outils

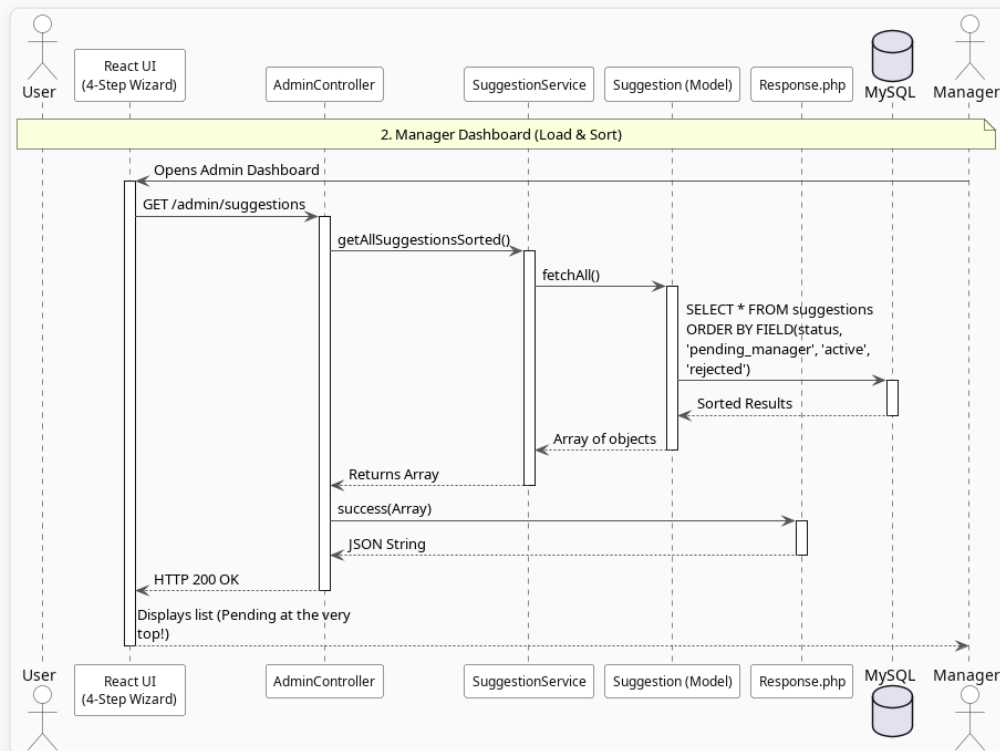
Étape 0 — Vérification initiale (Anti-Spam Lock) : Avant d'afficher le formulaire de suggestion, le frontend appelle `GET /suggestions/history`. `SuggestionService::checkPendingSuggestion()` recherche une suggestion en attente pour cet utilisateur. Si une suggestion est déjà en attente, l'interface est verrouillée sur l'étape 3 ("En attente") et l'utilisateur ne peut pas soumettre une nouvelle suggestion. Sinon, le formulaire vide de l'étape 1 s'affiche.



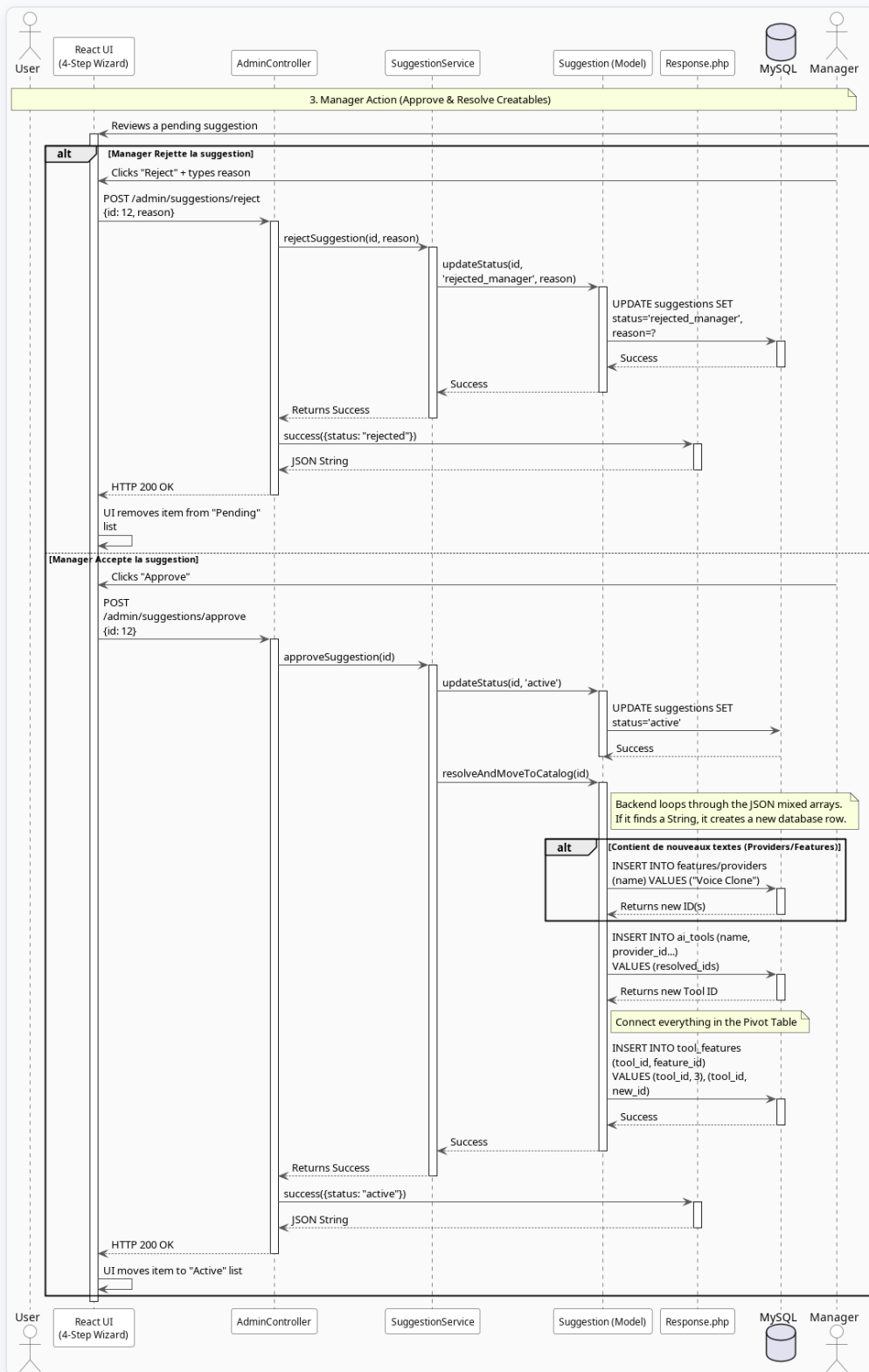
Étape 1 — Soumission et Vérification d'URL en Doublet : L'utilisateur remplit le formulaire (nom, URL, description, catégorie, fournisseur, fonctionnalités, etc.) et le soumet. `SuggestionService::createSuggestion()` vérifie d'abord si l'URL de l'outil existe déjà dans la base de données (via `SELECT id FROM ai_tools WHERE url = ?`). Si l'URL existe, la suggestion est immédiatement rejetée (statut 'rejected') avec la raison "URL exists". Si l'URL est nouvelle, l'IA externe est appelée pour valider l'outil. Si l'IA approuve, la suggestion est sauvegardée avec le statut 'waiting_manual_validation'.



Étape 2 — Chargement du Dashboard Manager : Le Manager ouvre son tableau de bord administratif. `GET /admin/suggestions` déclenche `SuggestionService::getAllSuggestionsSorted()` qui exécute une requête `SELECT` triée par statut (les suggestions en attente en premier). Les résultats sont affichés dans une interface de validation.

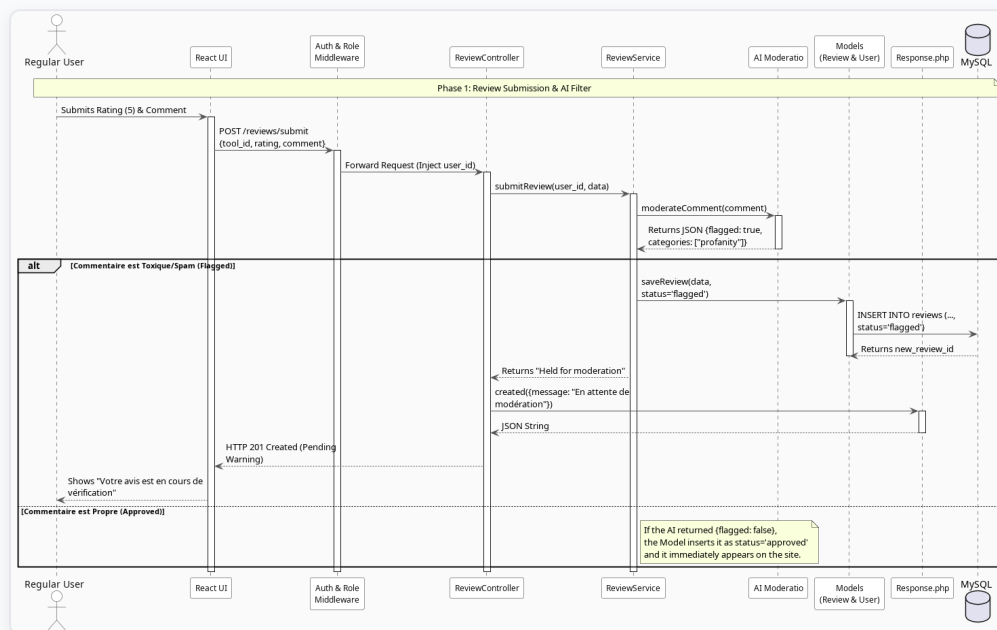


Étape 3 — Action du Manager (Approbation et Résolution) : Le Manager peut rejeter une suggestion (avec motif) ou l'approuver. En cas d'approbation, `SuggestionService::promoteToTool()` effectue les opérations suivantes : (1) Parcourt les tableaux JSON mixtes pour créer les nouvelles entités (fournisseurs, fonctionnalités) si nécessaire ; (2) Insère le nouvel outil dans `ai_tools` ; (3) Insère les relations dans `tool_features` ; (4) Met à jour le statut de la suggestion en 'published_to_catalog'. L'outil devient alors visible dans le catalogue public.

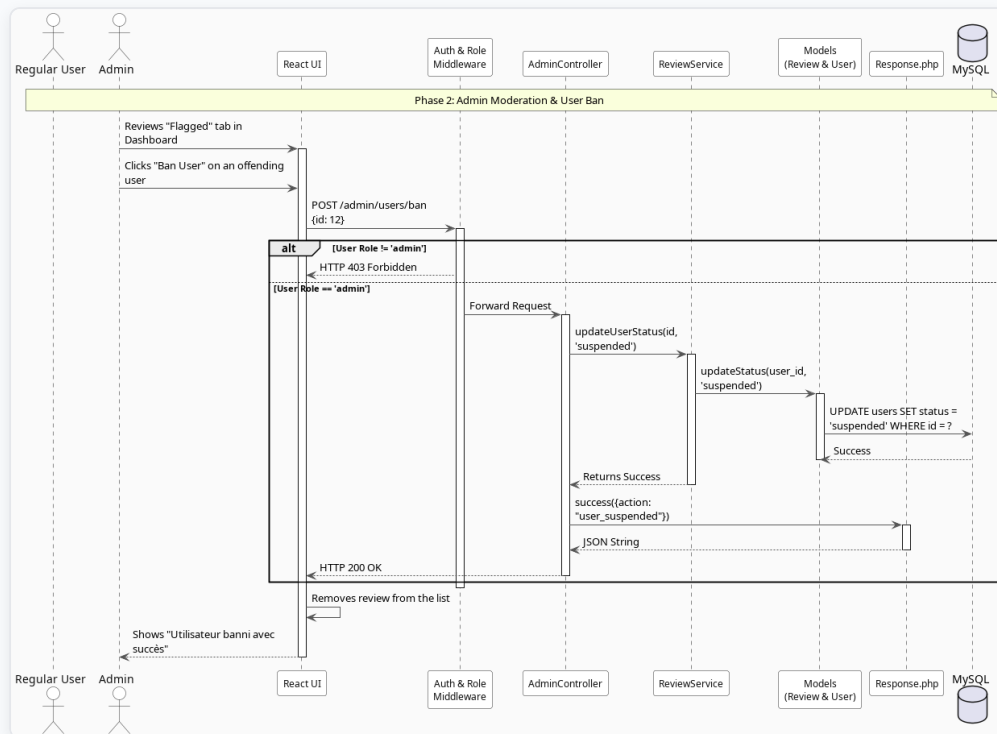


3.2.7 Modération du Contenu

Phase 1 — Filtre IA à la soumission d'un avis : Ce diagramme se concentre sur le pipeline de modération automatique. Lorsqu'un utilisateur soumet un avis, `ReviewService::submitReview()` appelle `OpenRouterService::moderateComment()` qui envoie le commentaire à l'API de modération IA. L'IA retourne un JSON indiquant si le contenu est signalé (`flagged: true/false`) et les catégories détectées. Si le commentaire est propre, il est approuvé immédiatement. S'il est toxique, il est marqué 'flagged' et mis en file d'attente pour modération manuelle.

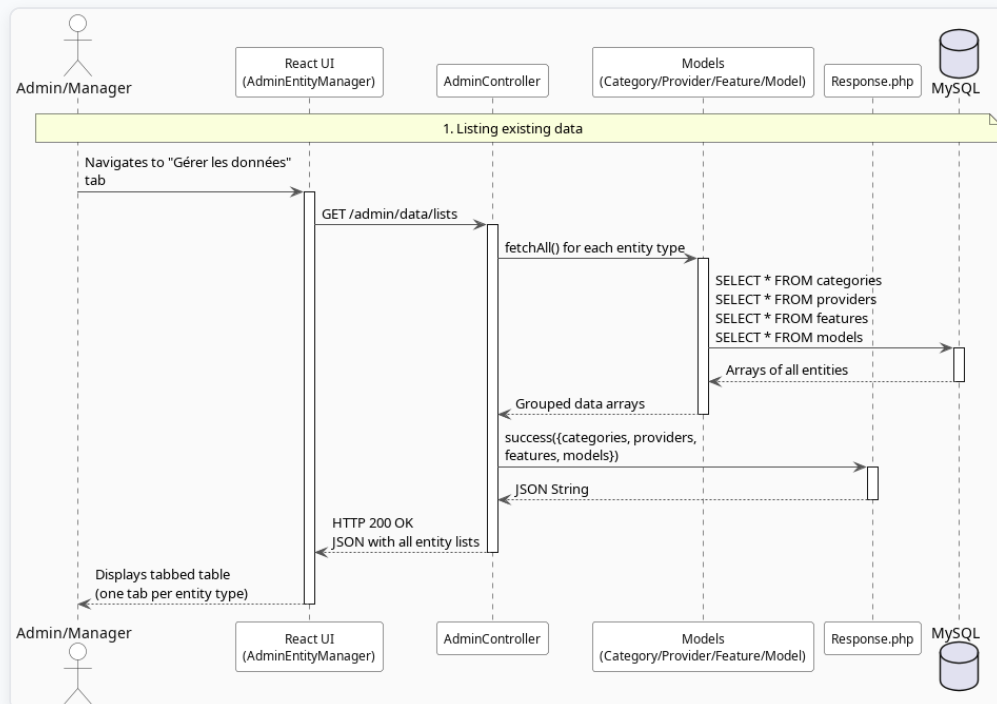


Phase 2 — Modération Admin et Bannissement : L'administrateur consulte l'onglet "Flagged" du dashboard, examine les avis signalés et peut décider de bannir définitivement l'utilisateur incriminé. `AdminService::updateUserStatus()` suspend l'utilisateur (statut 'suspended', durée 'forever') et `ReviewService::banUserAndCleanUp()` supprime l'avis offensant. L'interface React retire l'avis de la liste et confirme le bannissement.

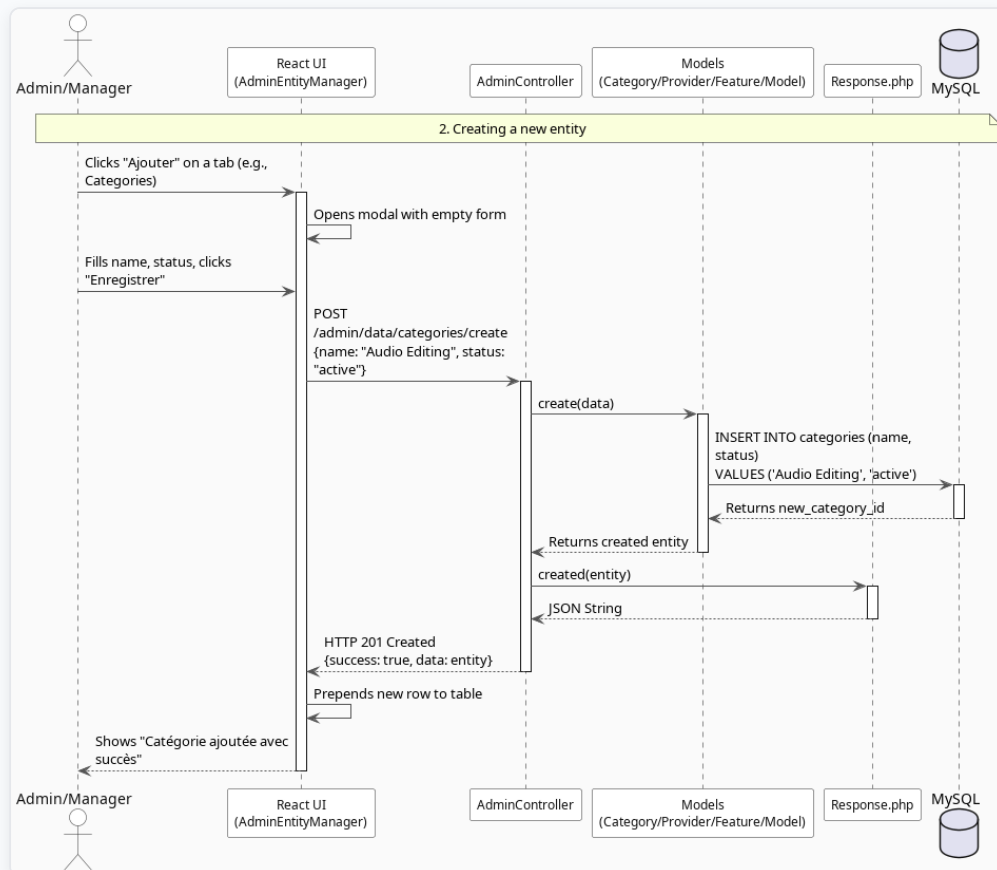


3.2.8 Gestion des Données (Admin CRUD)

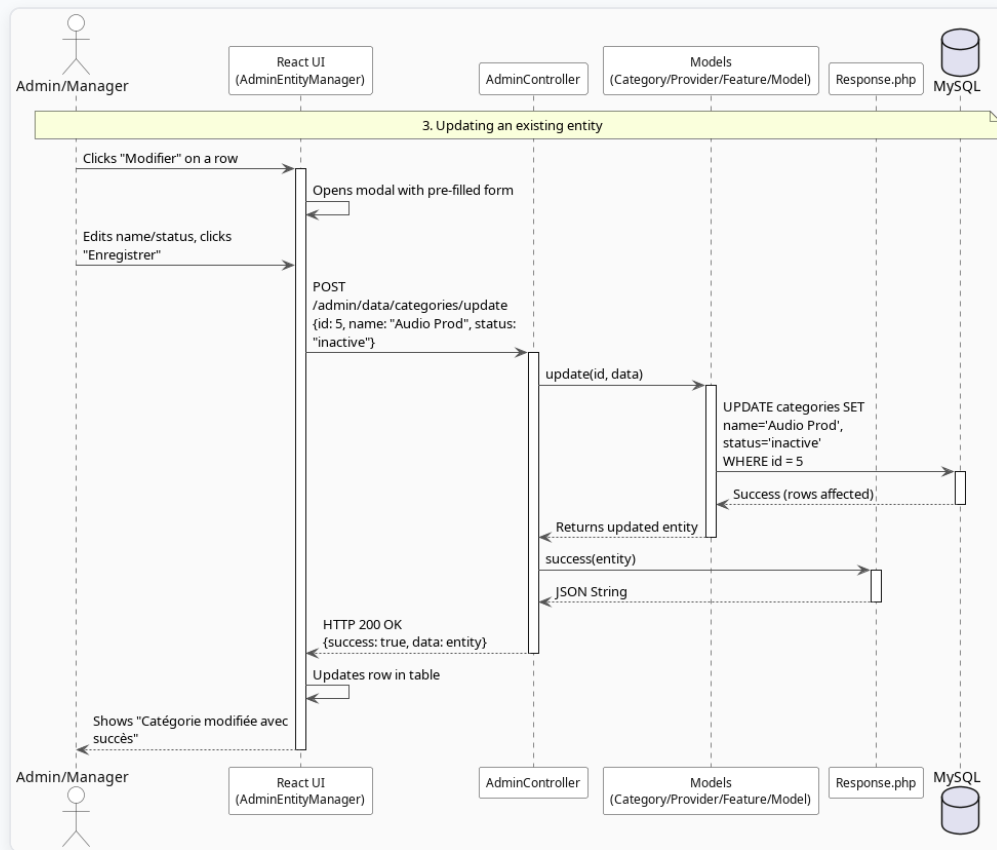
Séquence — Lister les entités existantes : L'administrateur navigue vers l'onglet "Gérer les données". `AdminController` exécute des `SELECT` sur toutes les tables de référence (`categories`, `providers`, `features`, `models`) et retourne les données groupées. L'interface affiche un tableau avec des onglets pour chaque type d'entité.



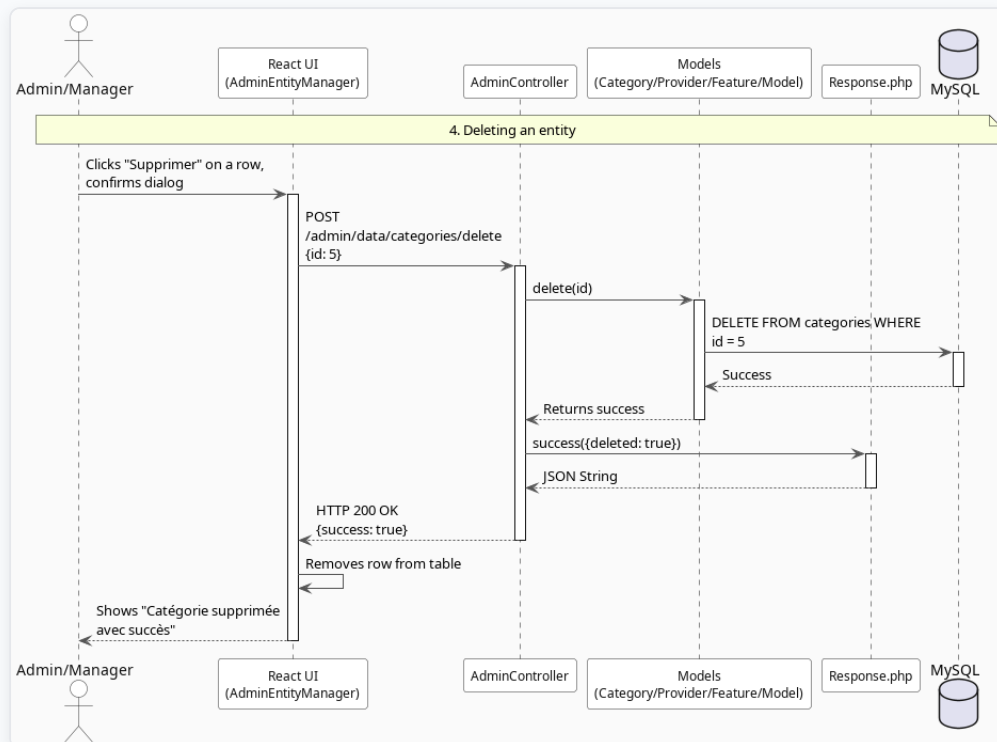
Séquence — Création d'une nouvelle entité : L'administrateur clique sur "Ajouter" dans un onglet, remplit le formulaire (nom, statut) et enregistre. Le modèle correspondant exécute un INSERT et retourne le nouvel enregistrement. La ligne apparaît immédiatement dans le tableau.



Séquence — Modification d'une entité existante : L'administrateur clique sur "Modifier" sur une ligne, modifie les champs dans une modale pré-remplie et enregistre. Le modèle exécute un UPDATE sur l'enregistrement correspondant. La ligne est mise à jour dans le tableau.



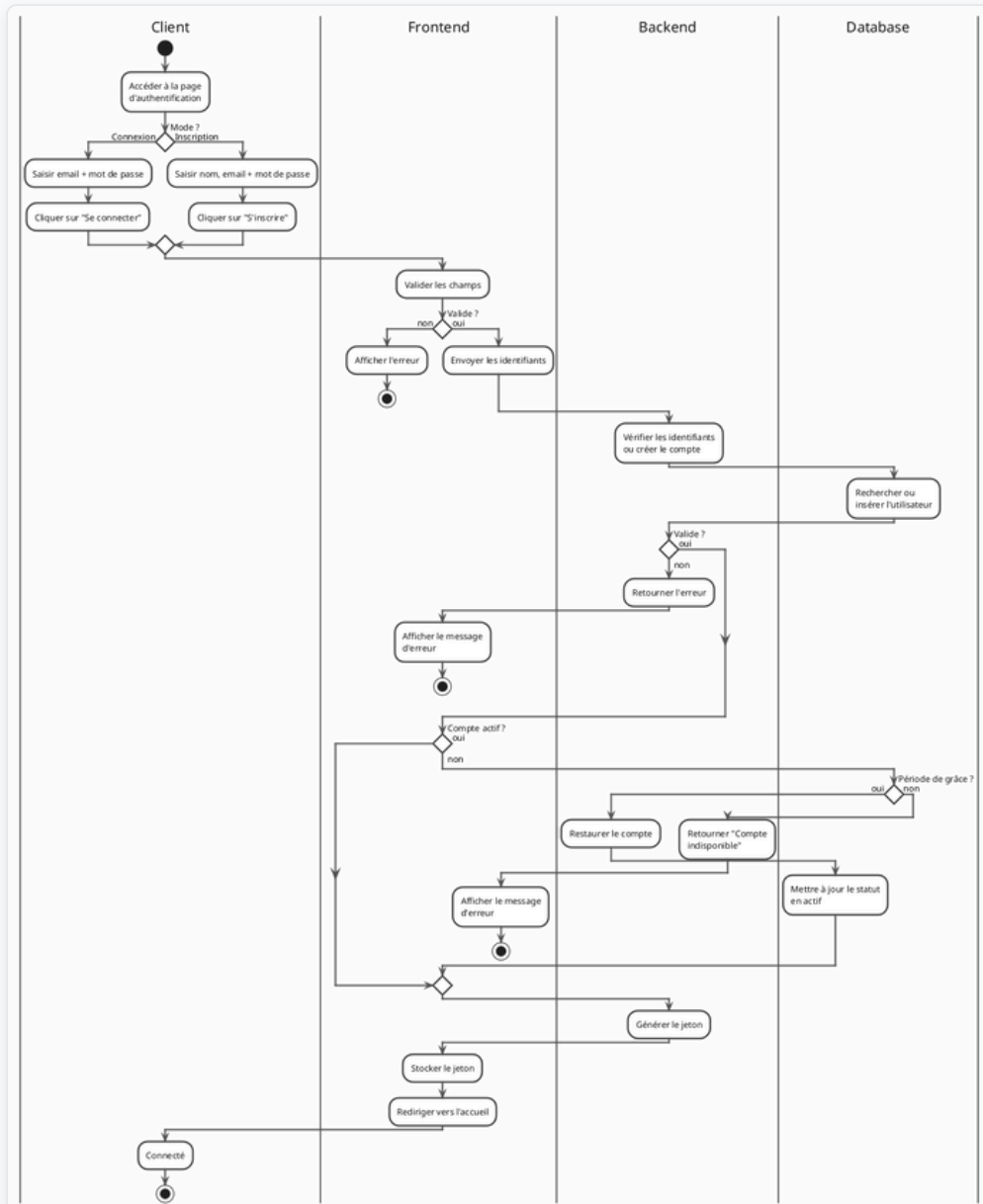
Séquence — Suppression d'une entité : L'administrateur clique sur "Supprimer" sur une ligne et confirme la suppression. Le modèle exécute un DELETE. La ligne disparaît du tableau.



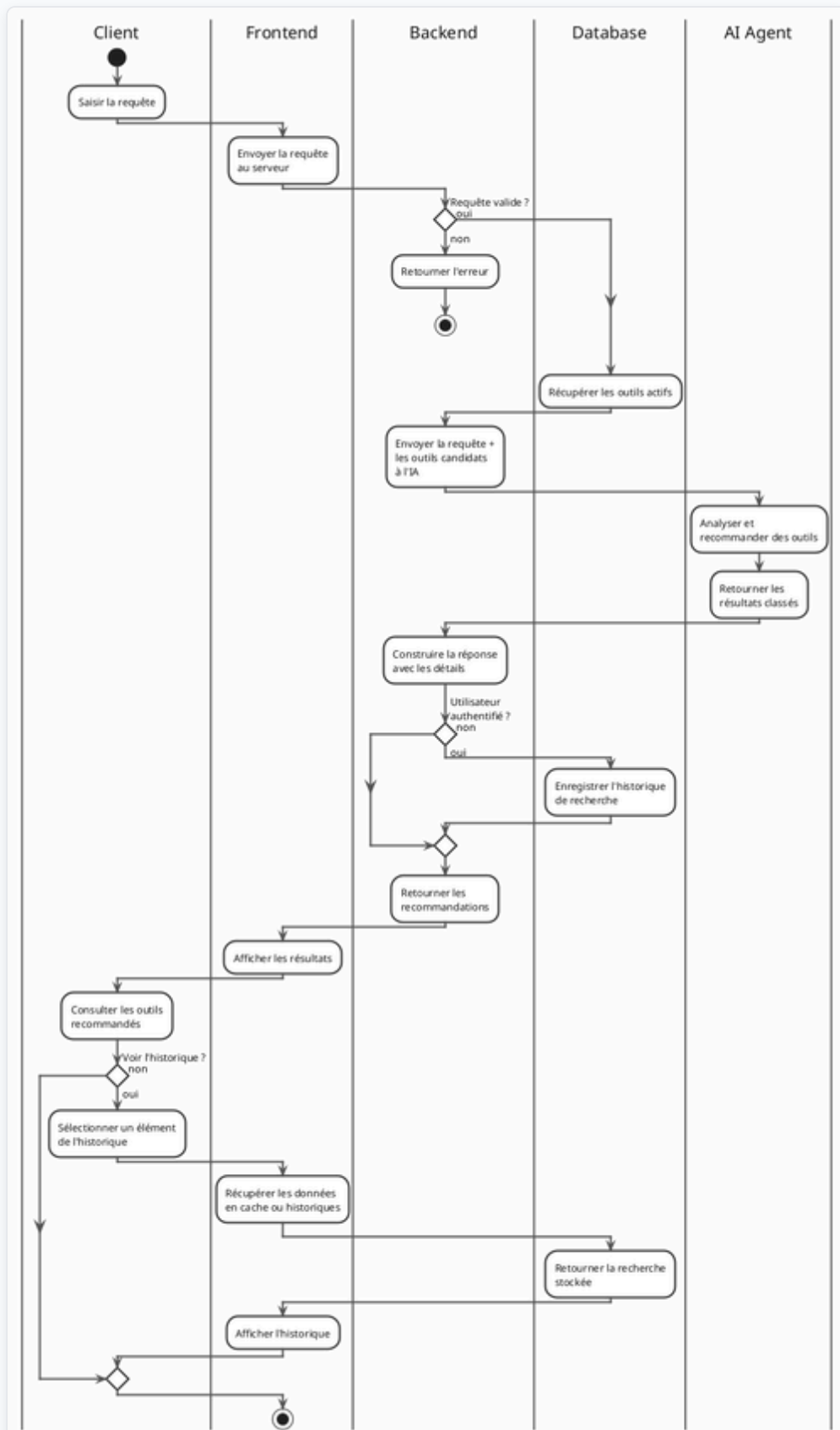
3.3 Diagrammes d'Activité

Les diagrammes d'activité décrivent le flux de contrôle et la logique algorithmique des processus métier complexes, en mettant en évidence les décisions conditionnelles, les actions parallèles et les rôles des différents acteurs (Utilisateur, Frontend React, Backend, IA, Base de données).

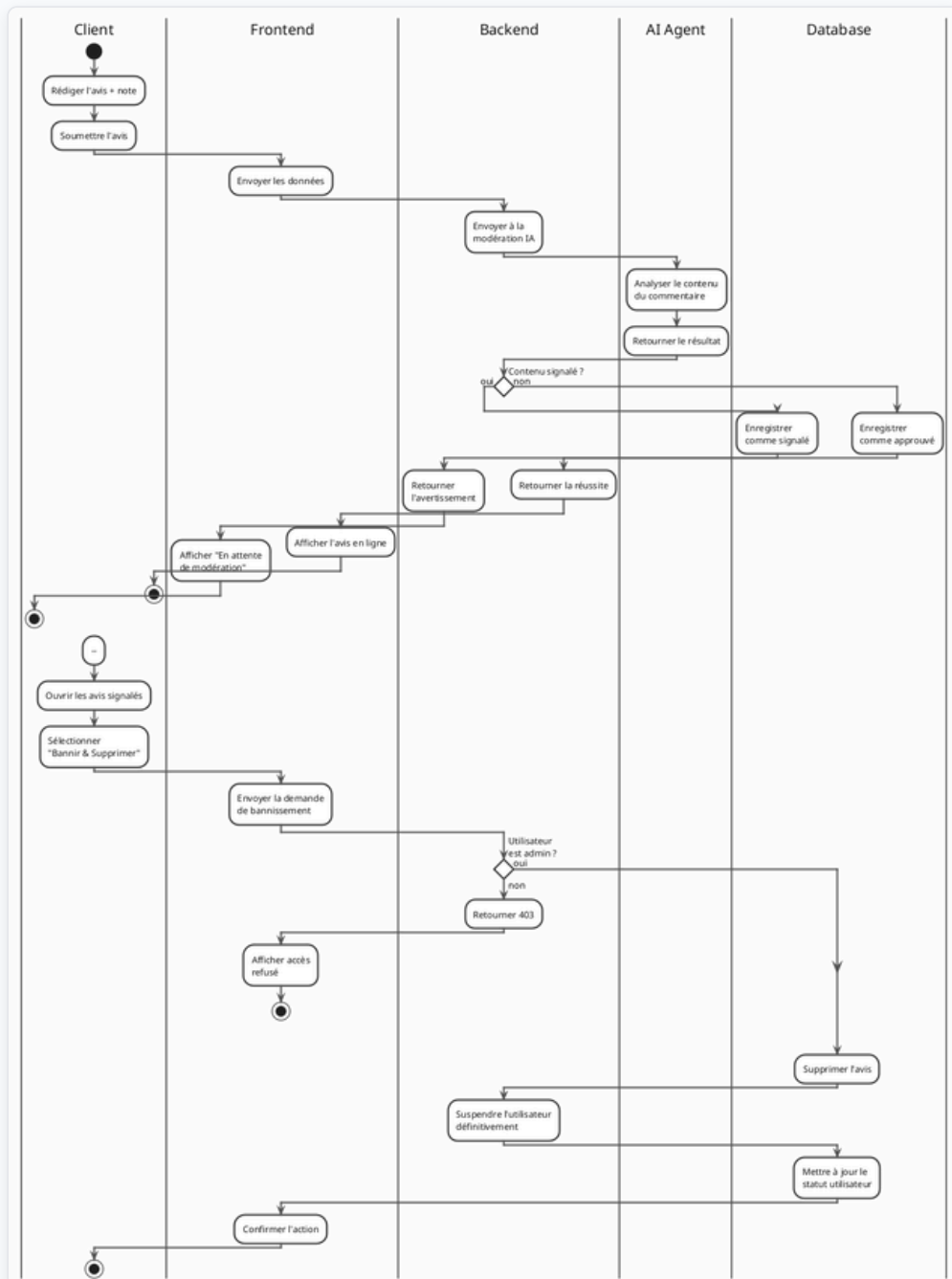
Activité — Flux d'Authentification Complet : Ce diagramme présente le processus d'authentification (connexion/inscription) avec les validations frontend, la vérification des identifiants en base, la gestion du statut du compte (actif, suspendu, période de grâce) et la restauration automatique du compte si l'utilisateur se reconnecte durant la période de grâce de 30 jours.



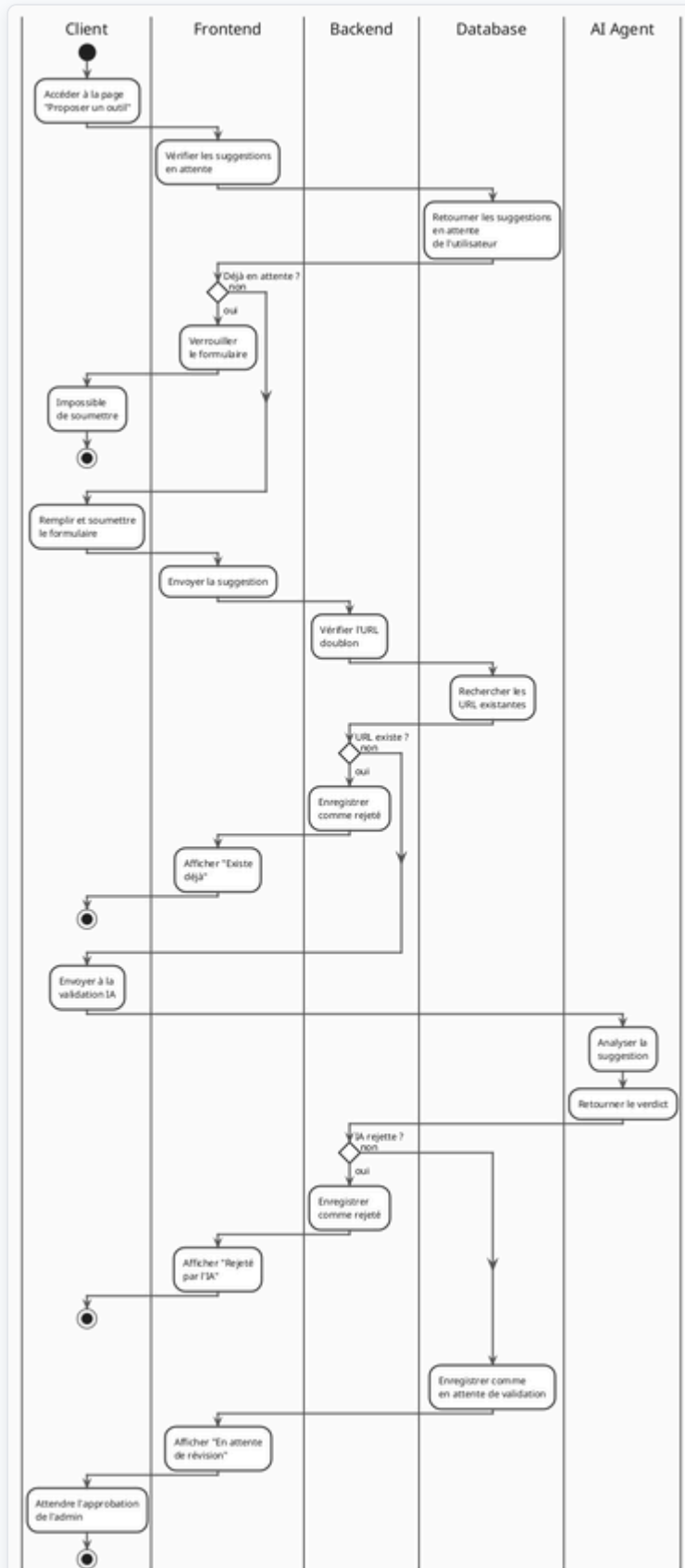
Activité — Recherche IA Intelligente : Ce diagramme illustre le processus de recherche par prompt en langage naturel : l'utilisateur saisit sa requête, le backend récupère les outils actifs et les soumet à l'IA pour analyse et recommandation. Si l'utilisateur est authentifié, la recherche est enregistrée dans l'historique, permettant de recharger les résultats ultérieurement.



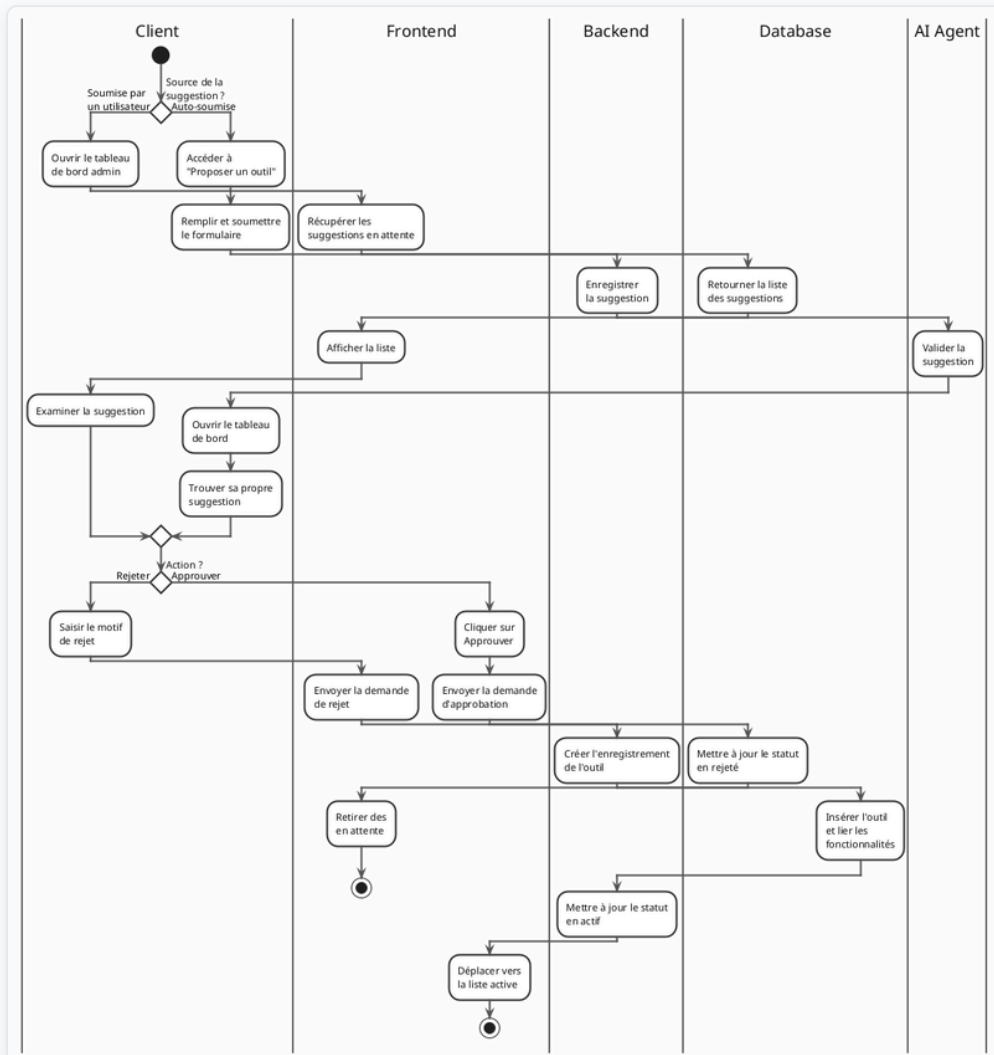
Activité — Pipeline de Modération Admin : Ce diagramme d'activité en deux parties montre d'abord le processus automatique de modération des avis : soumission, analyse IA, approbation immédiate ou mise en file d'attente ('flagged'). La seconde partie illustre l'intervention manuelle de l'administrateur qui examine les avis signalés et peut décider de bannir l'utilisateur, ce qui déclenche la suppression de l'avis et la suspension du compte.



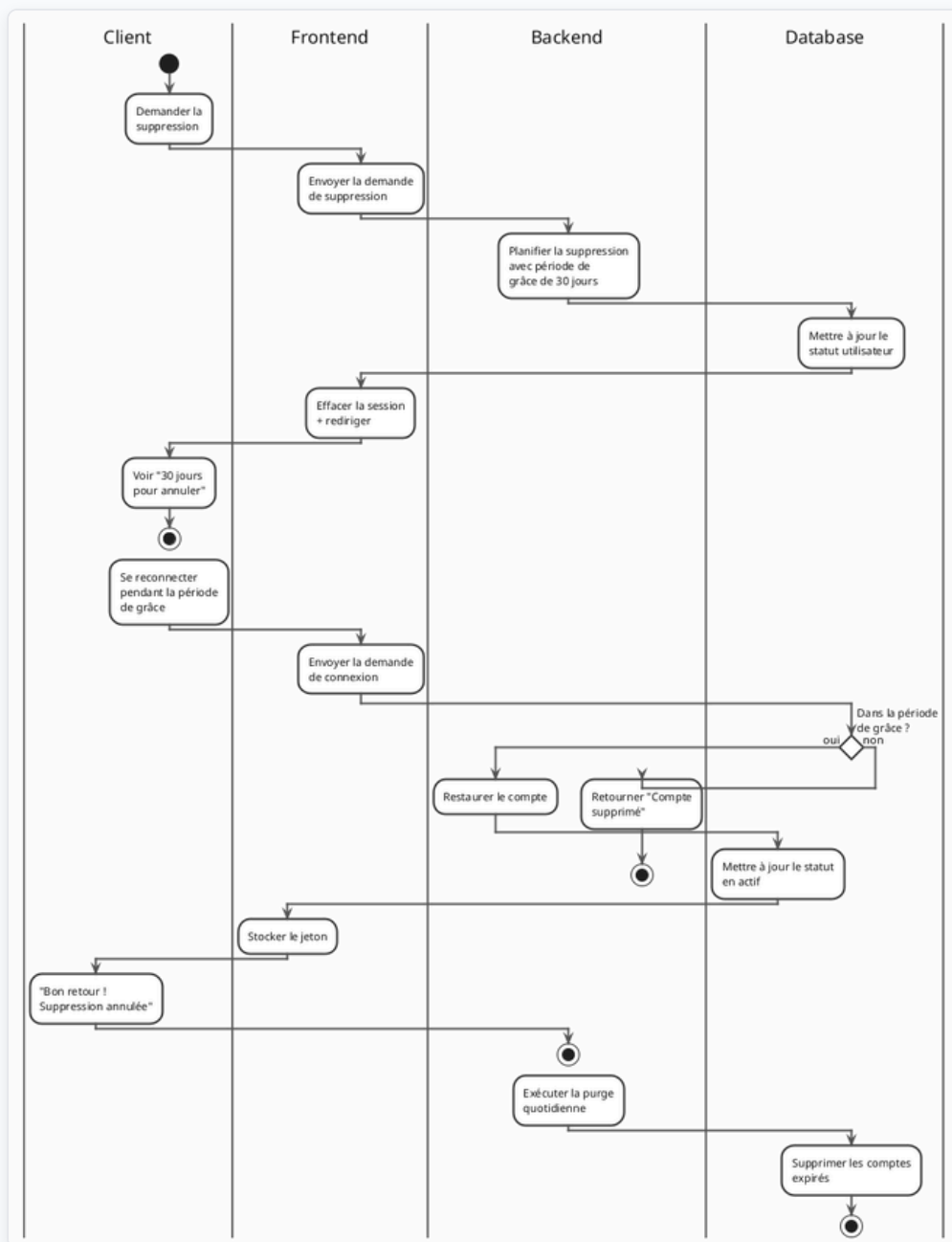
Activité — Suggestion d'un Outil par l'Utilisateur : Ce diagramme détaille le parcours utilisateur pour proposer un nouvel outil IA. Il vérifie d'abord qu'aucune suggestion n'est déjà en attente, puis soumet le formulaire. Le système vérifie l'URL en doublon, puis soumet l'outil à l'IA pour validation. Selon les résultats, la suggestion est rejetée (URL existante ou IA bloque) ou mise en attente de validation manuelle par un administrateur.



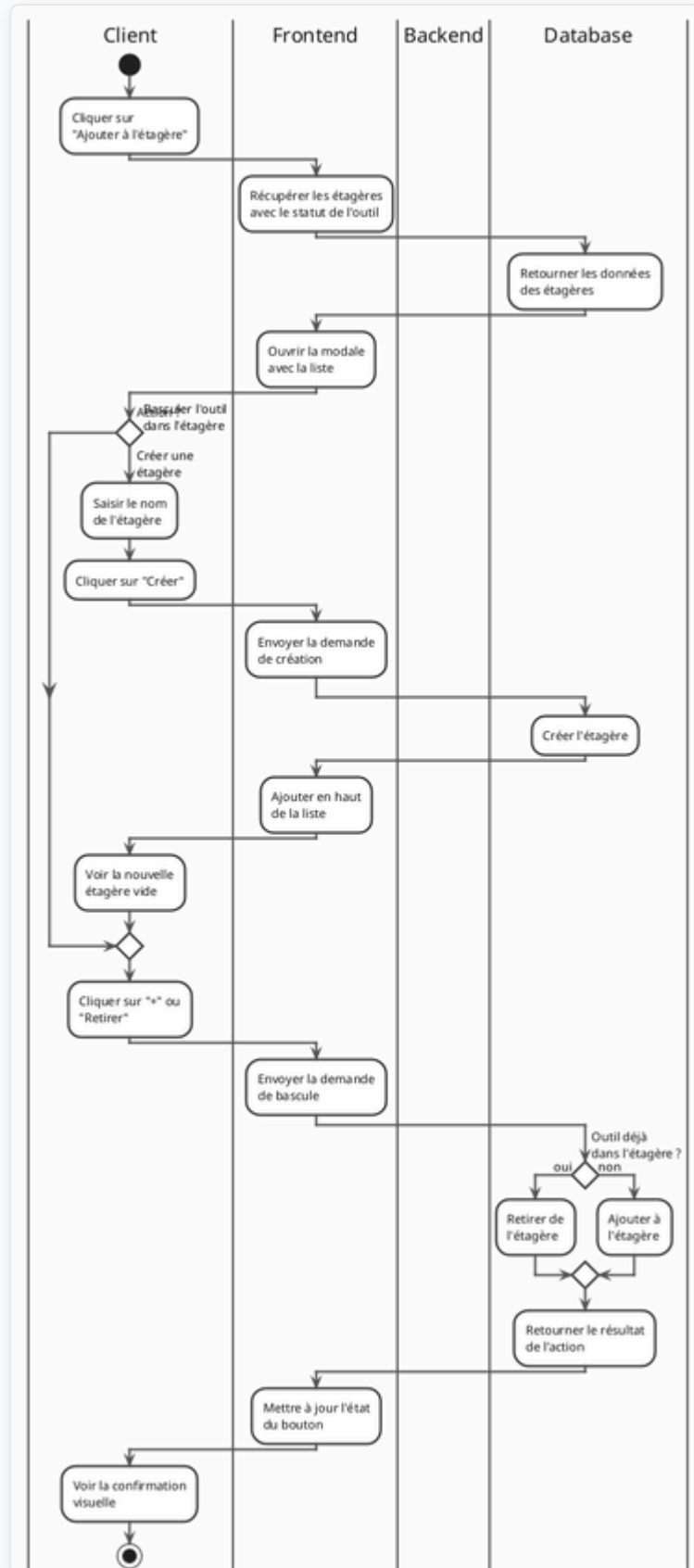
Activité — Gestion Admin des Suggestions : Ce diagramme montre le processus côté administrateur. Le Manager peut soit examiner une suggestion soumise par un utilisateur, soit créer lui-même une suggestion. Dans les deux cas, il peut la rejeter avec un motif ou l'approuver, ce qui déclenche la création de l'outil et son passage en statut actif.



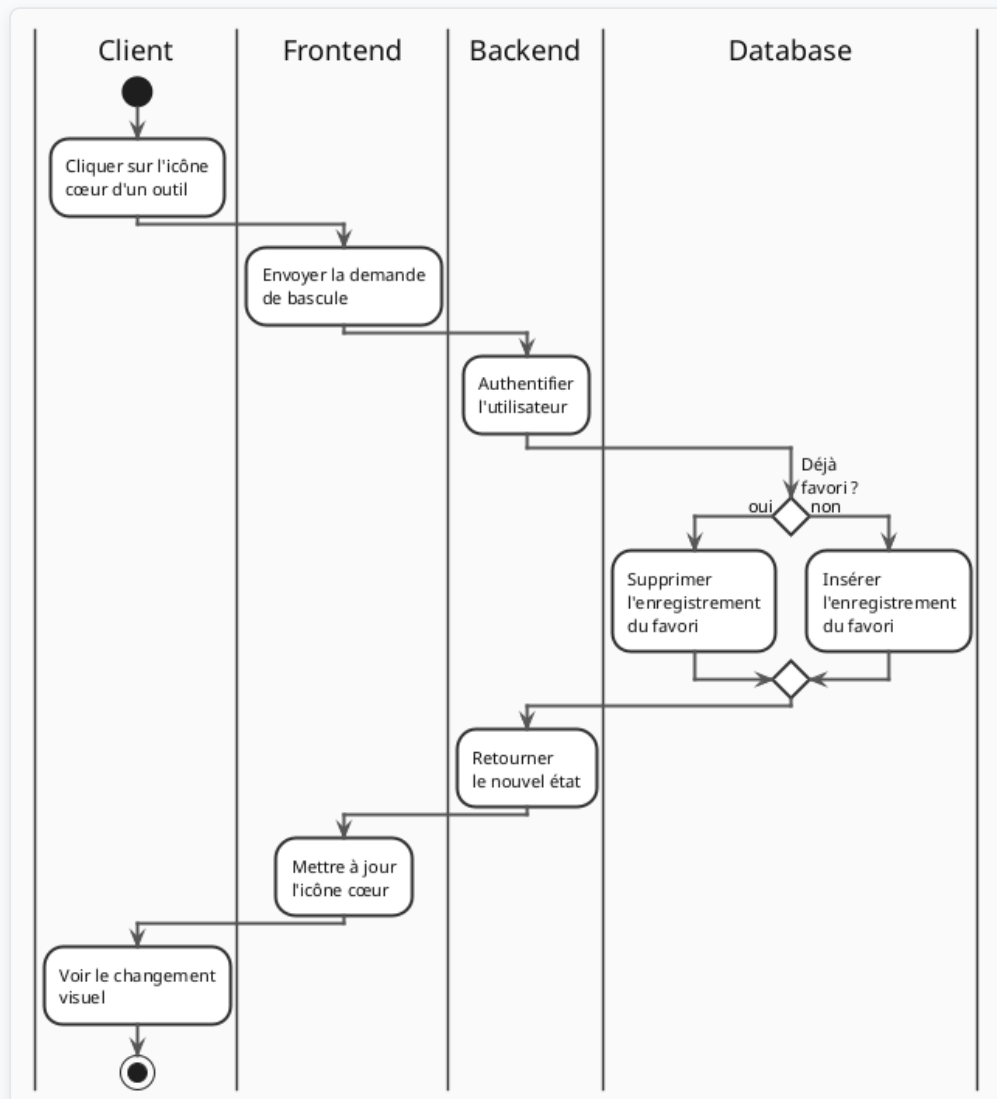
Activité — Suppression de Compte (Auto-suppression) : Ce diagramme d'activité présente le mécanisme de suppression de compte avec période de grâce. L'utilisateur demande la suppression, le système marque le compte comme 'deleted' et planifie la suppression permanente dans 30 jours. Si l'utilisateur se reconnecte avant l'expiration, le compte est restauré automatiquement. Un script CRON quotidien supprime définitivement les comptes dont la période de grâce est expirée.



Activité — Gestion des Étagères (Shelf Modal) : Ce diagramme détaille l'interaction utilisateur avec la modale de gestion des étagères. L'utilisateur peut voir la liste de ses étagères, créer une nouvelle étagère en saisissant un nom, ou basculer la présence d'un outil dans une étagère existante via un bouton "+"/"Retirer". Chaque action déclenche une requête API appropriée et l'interface se met à jour en temps réel.



Activité — Gestion des Favoris : Ce diagramme illustre le processus de basculement des favoris. L'utilisateur clique sur l'icône cœur d'un outil, le backend authentifie la requête et bascule l'état (ajout/retrait) en base de données, puis l'interface met à jour l'icône en conséquence.

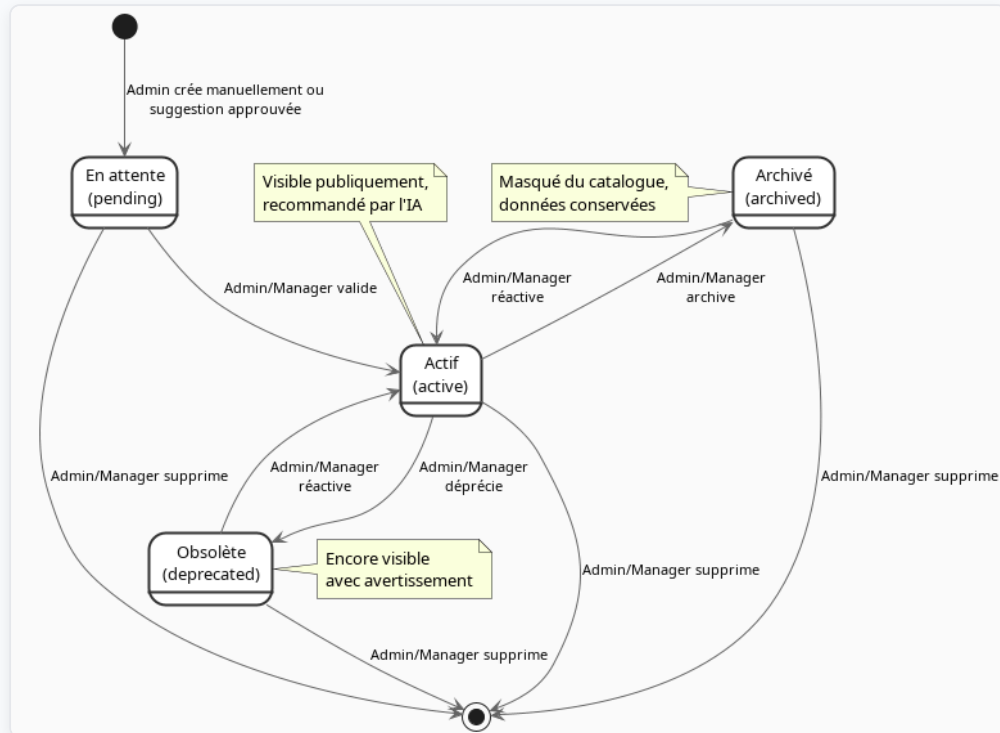


3.4 Diagrammes d'États (Cycle de Vie)

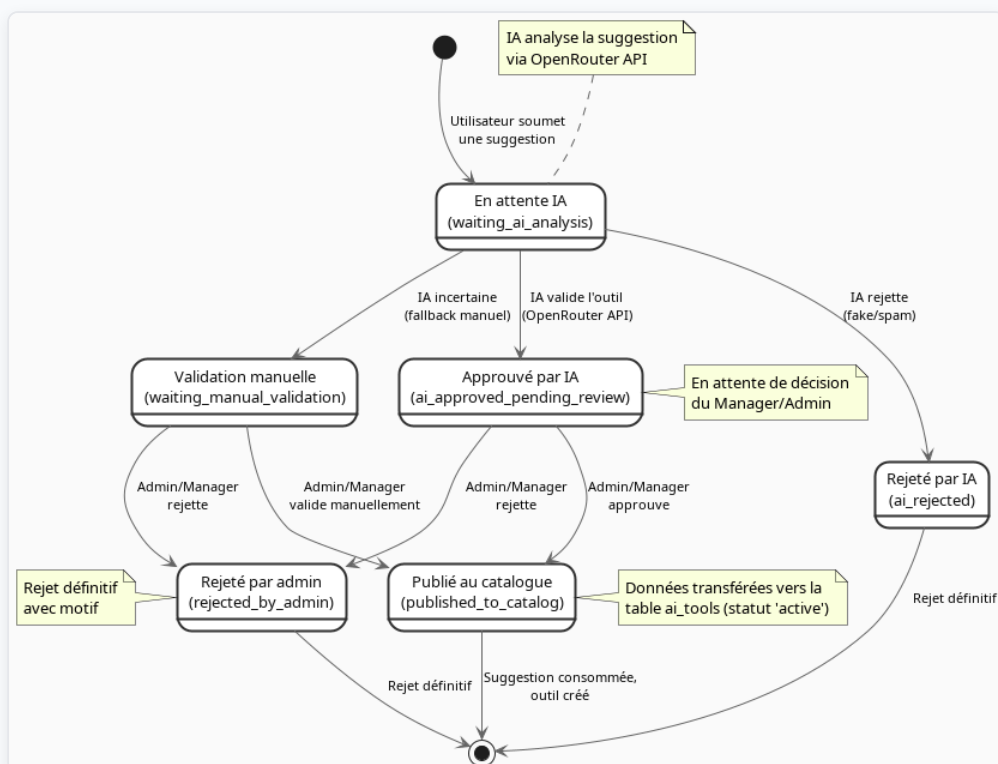
Les diagrammes d'états décrivent les différents états par lesquels passe une entité au cours de son cycle de vie dans le système, ainsi que les transitions possibles entre ces états.

Cycle de Vie d'un Outil (AiTool) : Un outil commence à l'état 'pending' (en attente) lorsqu'il est créé manuellement par un admin ou via une suggestion approuvée. L'admin peut le valider pour le passer à 'active' (visible publiquement et recommandé par l'IA). Un outil

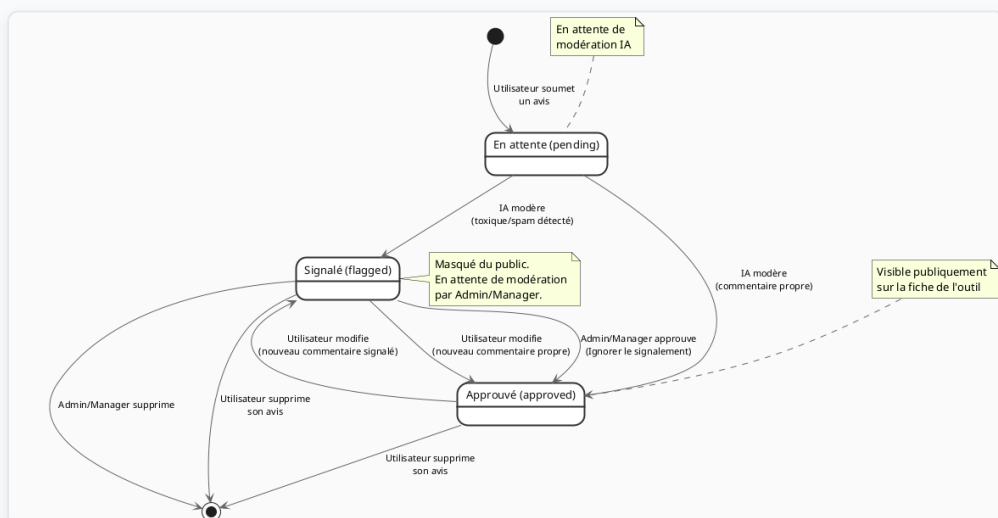
actif peut être 'archived' (masqué du catalogue, données conservées) ou 'deprecated' (encore visible mais avec un avertissement). Un outil peut être réactivé depuis ces deux états, ou supprimé définitivement depuis n'importe quel état.



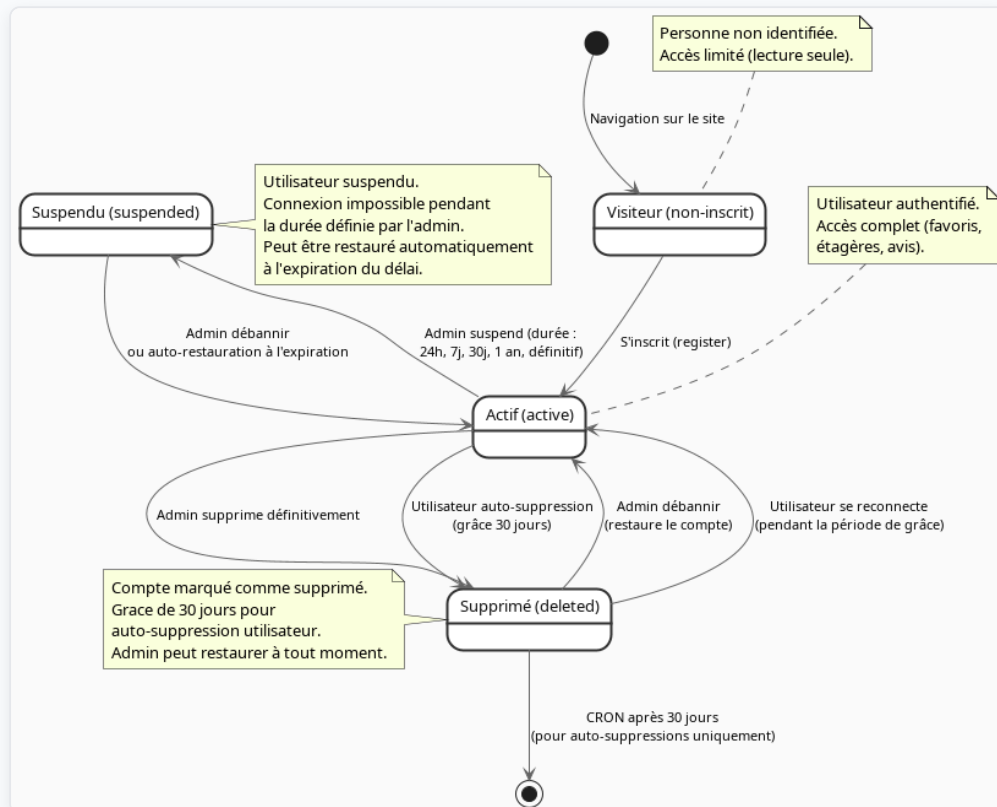
Cycle de Vie d'une Suggestion : Une suggestion commence à l'état 'waiting_ai_analysis' après soumission par l'utilisateur. L'IA analyse l'outil et peut le passer à 'ai_approved_pending_review' (approuvé par IA, en attente admin), 'ai_rejected' (rejeté par IA) ou 'waiting_manual_validation' (incertain, fallback manuel). Le Manager peut alors approuver la suggestion (→ 'published_to_catalog') ou la rejeter (→ 'rejected_by_admin'). Une fois publiée, la suggestion est consommée et l'outil est créé dans `ai_tools`.



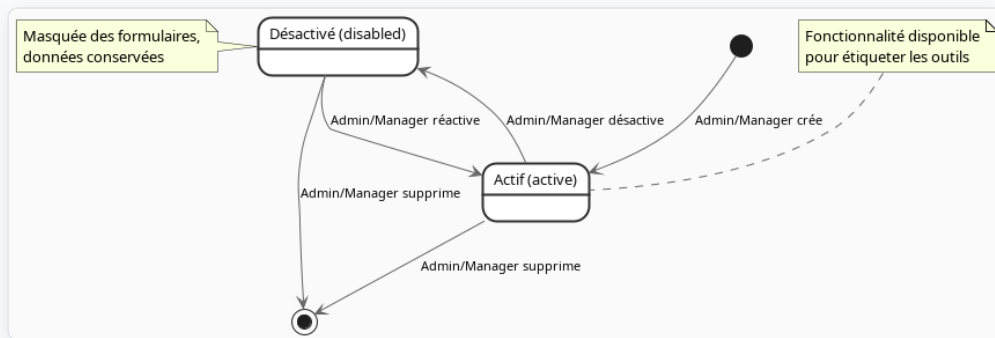
Cycle de Vie d'un Avis (Review) : Un avis peut être créé directement à l'état 'approved' (commentaire propre) ou 'flagged' (commentaire toxique/spam détecté par l'IA). Un avis approuvé peut passer à 'flagged' si l'utilisateur le modifie avec un contenu inapproprié. Un avis 'flagged' peut être approuvé par l'admin (ignorer le signalement) ou supprimé (par l'admin ou par l'utilisateur lui-même). Les avis 'approved' sont visibles publiquement sur la fiche de l'outil.



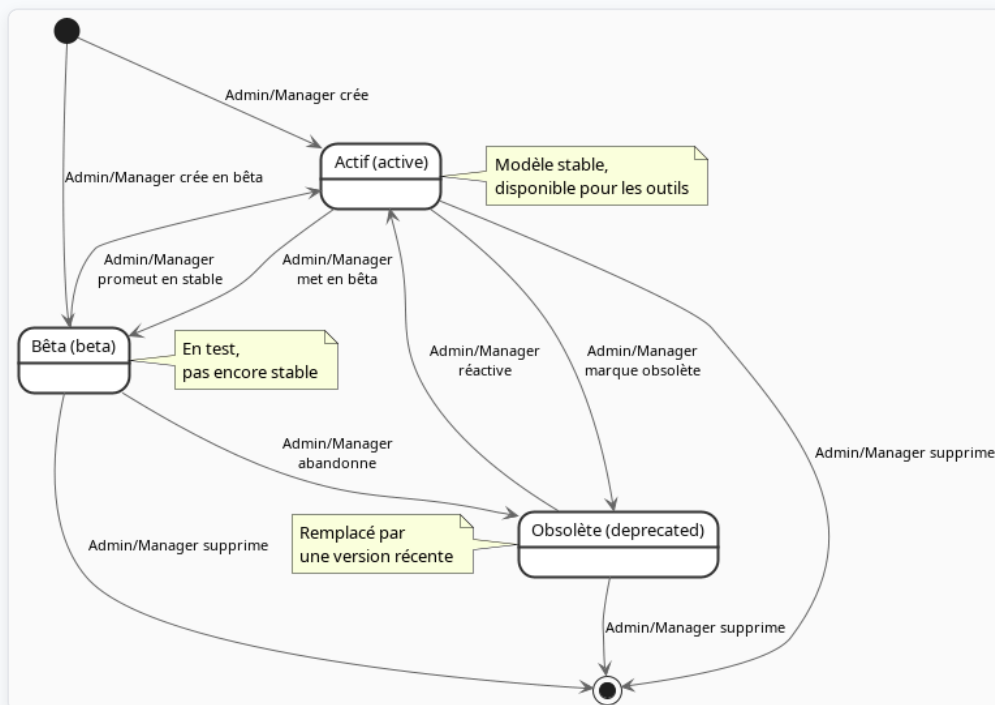
Cycle de Vie d'un Utilisateur (User) : Un visiteur non-inscrit devient un utilisateur 'active' après inscription. Un utilisateur actif peut être 'suspended' (suspendu par l'admin avec une durée définie : 24h, 7j, 30j, 1 an ou définitif) ou 'deleted' (supprimé par l'admin ou auto-suppression par l'utilisateur avec grâce de 30 jours). Un compte suspendu peut être restauré automatiquement à l'expiration du délai ou par l'admin (débannir). Un compte supprimé peut être restauré par l'admin ou par l'utilisateur s'il se reconnecte pendant la période de grâce.



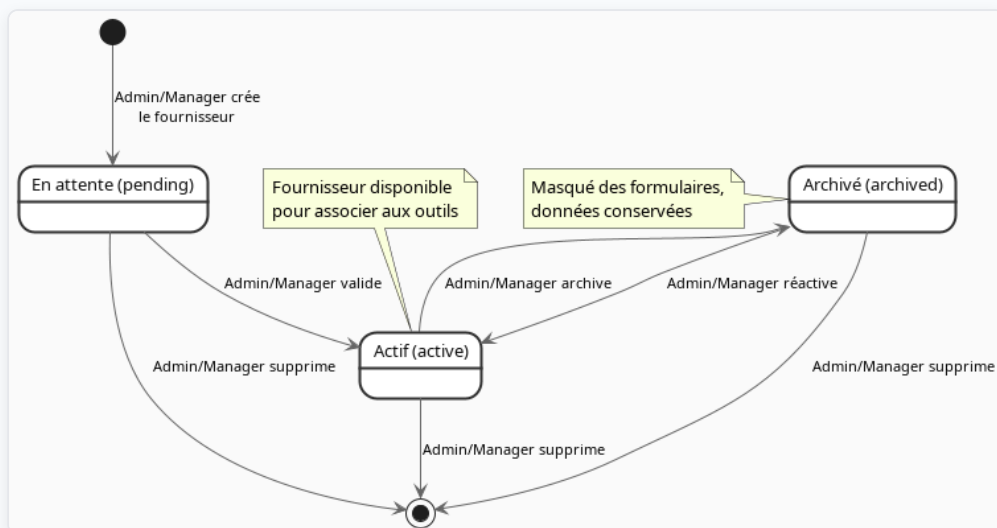
Cycle de Vie d'une Fonctionnalité (Feature) : Une fonctionnalité créée par l'admin est 'active' (disponible pour étiqueter les outils). L'admin peut la désactiver ('disabled') pour la masquer des formulaires tout en conservant les données, ou la réactiver. Elle peut être supprimée définitivement depuis n'importe quel état.



Cycle de Vie d'un Modèle (Model) : Un modèle d'IA peut être créé comme 'active' (stable) ou 'beta' (en test). Il peut passer de beta à active (promu en stable), de active à beta, ou vers 'deprecated' (obsolète, remplacé). Un modèle déprécié peut être réactivé.

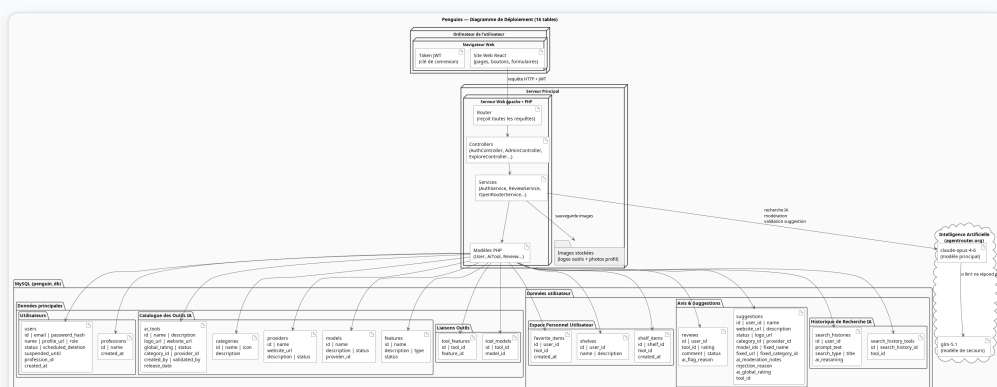


Cycle de Vie d'un Fournisseur (Provider) : Un fournisseur commence à l'état 'pending' (en attente de validation). L'admin peut le valider (→ 'active') ou le supprimer. Un fournisseur actif peut être archivé (→ 'archived', masqué des formulaires) ou réactivé.

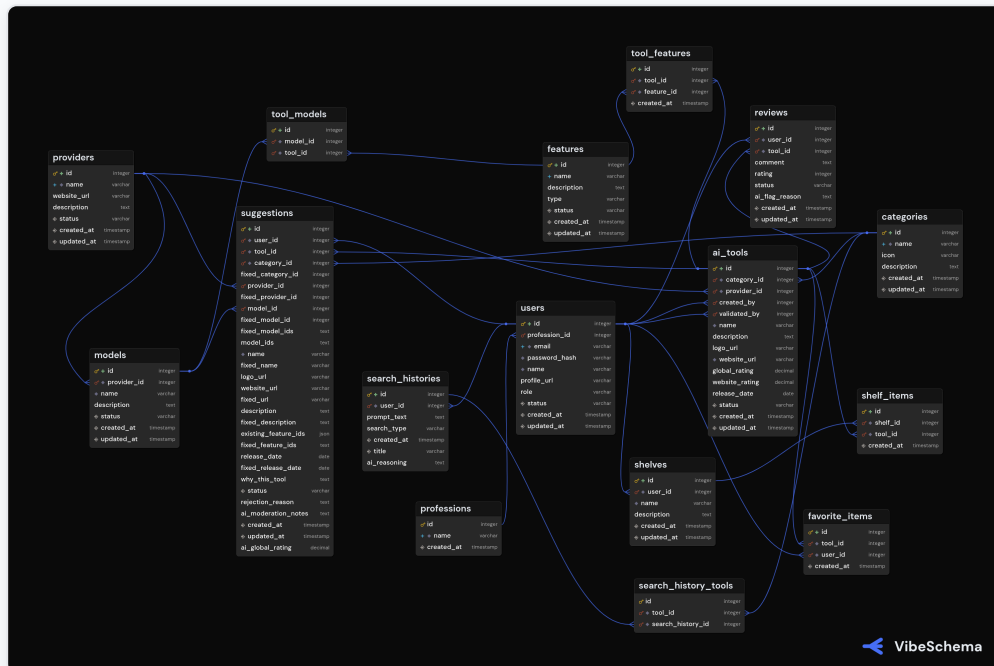


3.5 Architecture de Déploiement et Base de Données

Diagramme de Déploiement : Ce diagramme présente l'architecture physique et logique du système Penguins. On distingue quatre nœuds principaux : (1) **L'ordinateur de l'utilisateur** avec le navigateur web exécutant l'application React et stockant le token JWT ; (2) **Le serveur principal** avec Apache/PHP hébergeant le routeur, les contrôleurs, les services et les modèles, ainsi que le dossier de stockage des images uploadées ; (3) **La base de données MySQL** (penguin_db) organisée en deux packages (données principales : utilisateurs, catalogue, liaisons ; données utilisateur : espace personnel, avis, suggestions, historique) ; (4) **Le service d'IA externe** (agentrouter.org) utilisant Claude Opus 4.6 (modèle principal) et GLM 5.1 (modèle de secours). Les flèches indiquent les flux de données entre les composants.



Modèle Physique de Données (Base de données) : Ce schéma présente la structure complète de la base de données MySQL avec ses 16 tables. On retrouve les tables principales (`users` , `ai_tools` , `reviews` , `suggestions`), les tables de référence (`categories` , `providers` , `features` , `models` , `professions`), les tables de gestion utilisateur (`favorite_items` , `shelves` , `shelf_items` , `search_histories` , `search_history_tools`) et les tables de jonction (`tool_features` , `tool_models`). Chaque table est représentée avec ses colonnes clés.



4. Réalisation et Implémentation

Cette section détaille les choix technologiques, l'architecture du code et la présentation fonctionnelle de chaque écran de l'application.

4.1 Architecture et Technologies Frontend

Le Frontend de Penguins est une **Single Page Application (SPA)** développée avec **React.js v19** et **Vite.js** comme bundler. L'architecture est organisée en composants réutilisables, pages, layout, contexte global et services API.

4.1.1 Bibliothèques et Outils Utilisés

- **React.js v19** : Architecture en composants fonctionnels avec Hooks (`useState` , `useEffect` , `useContext` , `useCallback` , `useMemo`). Gestion d'état local par composant et état global via `AuthContext.jsx` .
- **Vite.js** : Outil de build nouvelle génération offrant un démarrage instantané et un Hot Module Replacement (HMR) ultra-rapide.
- **React Router DOM v7** : Routage côté client avec paramètres d'URL (`/tool/:name`), `useParams()` , `useNavigate()` , `useSearchParams()` et `Outlet` pour les layouts emboîtés.
- **Axios** : Client HTTP utilisé via `src/services/api.js` pour toutes les communications avec l'API Backend. Gère l'injection automatique du token JWT dans les en-têtes `Authorization` et la gestion centralisée des erreurs.
- **Framer Motion** : Bibliothèque d'animations pour les transitions fluides entre les vues et l'apparition progressive des composants.
- **Lucide-React & FontAwesome** : Bibliothèques d'icônes vectorielles SVG utilisées dans toute l'interface (barre de recherche, coeur, étoiles, menu, etc.).
- **React-Select (CreatableSelect)** : Composant de sélection avancé avec création automatique de nouvelles options, utilisé dans le formulaire de suggestion d'outils.
- **React-Datepicker** : Sélecteur de date utilisé dans les formulaires.
- **CSS Modules** : Fichiers `.module.css` associés à chaque composant pour un style isolé et sans conflit.

4.1.2 Structure du Projet Frontend

Le code source dans `src/` est organisé comme suit :

- **pages/** : 11 pages principales correspondant aux routes de l'application.
- **components/** : Composants réutilisables organisés par domaine (`admin/` , `home/` , `landing/` , `profile/` , `library/` , `toolDetails/` , `Validations/` , `ProposerOutils/` , `forms/` , `manager/`).
- **layout/** : Enveloppes structurales (`Layout.jsx` , `Menu.jsx` , `ManagerLayout.jsx`) qui maintiennent la navigation (navbar, sidebar) pendant que le contenu change via `<Outlet />` .
- **context/** : `AuthContext.jsx` gère l'état global d'authentification (`isAuthenticated` , `isAdmin` , `isManager` , `loading`) et le rend accessible à toute l'application via `useAuth()` .
- **routes/** : `routes.jsx` centralise la définition des routes publiques (`publicRoutes`) et d'administration (`adminRoutes`).
- **services/** : `api.js` configure Axios avec l'URL de base et les intercepteurs pour le token JWT.
- **style/** : Modules CSS organisés par pages et composants.

4.1.3 Logique de Fonctionnement

Lorsqu'un utilisateur visite l'application, le `AuthContext` vérifie la présence d'un token JWT dans le `localStorage` . Si un token valide est trouvé, l'utilisateur est automatiquement connecté et son rôle (user, manager, admin) est déterminé. L'interface s'adapte en conséquence : affichage du profil, accès à la bibliothèque, et pour les managers/admins, accès au panneau d'administration. Toutes les actions utilisateur (recherche, ajout aux favoris, soumission d'avis) déclenchent des requêtes Axios vers l'API Backend, et les résultats mettent à jour l'état React, ce qui redessine instantanément l'interface.

4.2 Présentation Détaillée des Écrans Utilisateurs

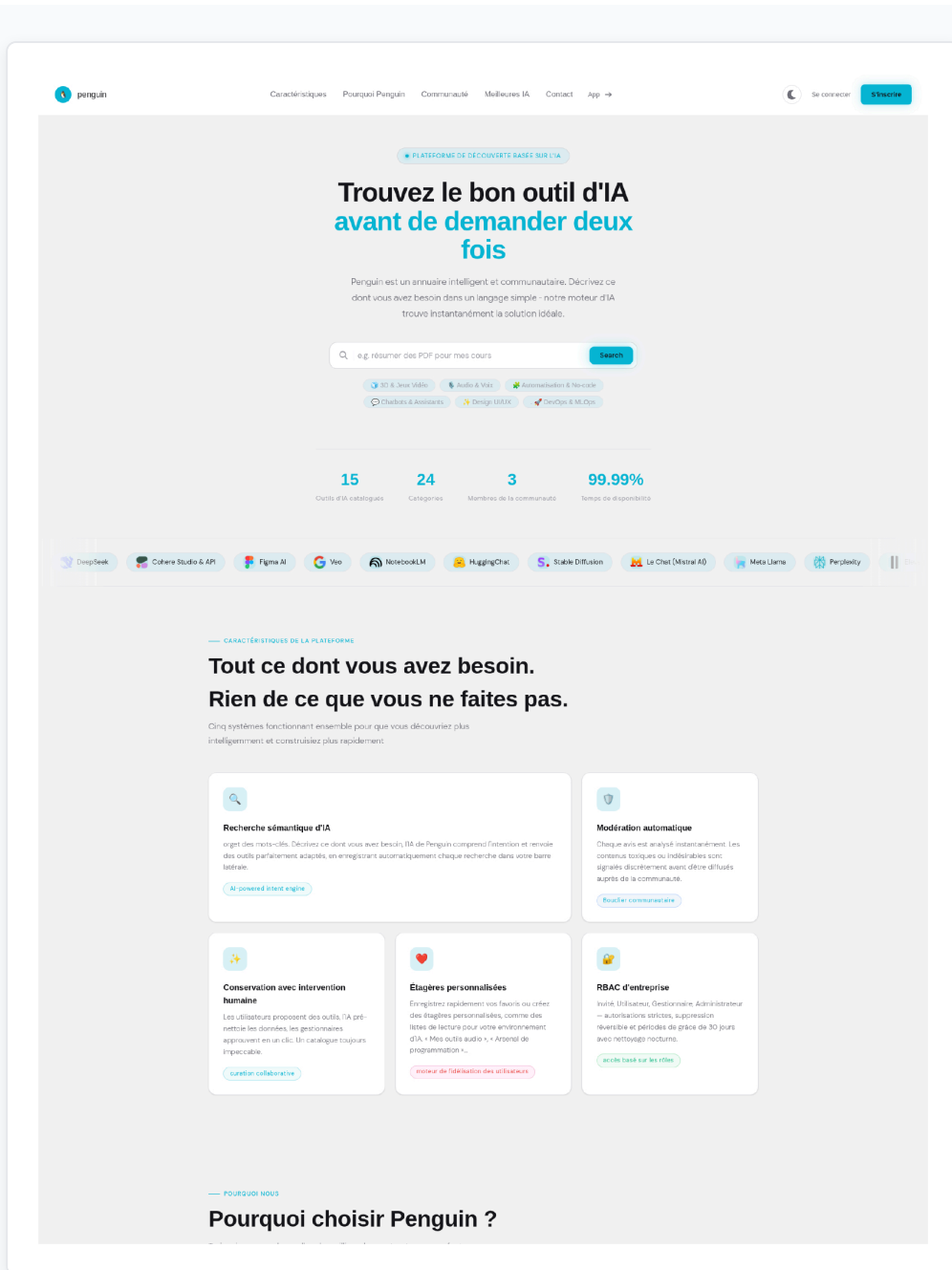
4.2.1 Landing Page (Page d'Accueil) — `LandingPage.jsx`

Ce que voit l'utilisateur : Une page vitrine complète avec plusieurs sections défilables : Hero, bandeau défilant de logos, fonctionnalités, témoignages, Top 6 IA, et footer.

Interactions utilisateur :

- **Section Hero** (`LandingHome`) : L'utilisateur découvre le slogan et la proposition de valeur de Penguins. Un bouton d'appel à l'action l'invite à explorer les outils.
- **Bandeau défilant** (`Marquee`) : Une animation horizontale affiche les logos des outils IA populaires.

- **Section Fonctionnalités (`Features`)** : L'utilisateur peut faire défiler les cartes présentant les fonctionnalités clés de la plateforme (recherche IA, bibliothèque personnelle, suggestions).
- **Section "Pourquoi Penguins" (`WhyPenguin`)** : Présentation des avantages avec des arguments convaincants.
- **Section Témoignages (`CommunityVoices`)** : Avis de la communauté sur l'utilité de la plateforme.
- **Section Top AI (`TopAI`)** : Affichage dynamique des 6 outils les mieux notés, récupérés via l'API `GET /landing/summary` . Chaque carte affiche le nom, la note et la catégorie de l'outil.
- **Footer (`Footer`)** : Liens de contact et crédits.
- L'utilisateur peut également naviguer vers les sections spécifiques via des ancres dans l'URL (`?scrollTo=features`).



Trois raisons pour lesquelles des milliers de constructeurs nous font confiance plutôt qu'à tout autre annuaire.



Une IA qui vous comprend vraiment

Plus besoin de deviner les mots-clés. Notre moteur sémantique comprend votre intention, pas seulement vos mots. Trouvez tout ce dont vous avez besoin, à chaque fois.



Qualité vérifiée par la communauté

Chaque note, chaque avis est authentique – modéré par une IA et validé par des milliers d'utilisateurs. Pas de fausses étoiles, pas de placement payant.



Votre espace de travail personnel en IA

Sauvegardez vos outils, créez des stagiaires, consultez à nouveau vos recherches. Penguin évolue avec votre flux de travail. Un second cerveau pour votre infrastructure d'IA, toujours à portée de main.

Rejoignez plus de 12 000 constructeurs qui découvrent des solutions plus intelligentes.

Free forever. No credit card required.

Commencez la découverte >>

LES VOIX DE LA COMMUNAUTÉ

Apprécié des constructeurs et des créateurs

SE

saad el-masli

Etudiant

• Deepseek

this ai help me creating

AZ

ACHRAF ZERYOUEL

Graphiste

• Perplexity

one of the best

AZ

ACHRAF ZERYOUEL

Graphiste

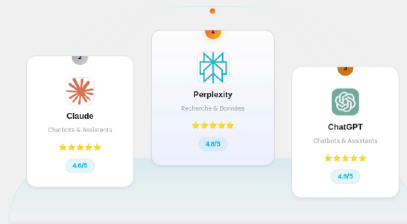
• Cohere Studio & API

Cohere Studio & API this ai is awesome I use it everyday

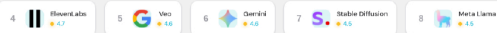
COMMUNITY RATED

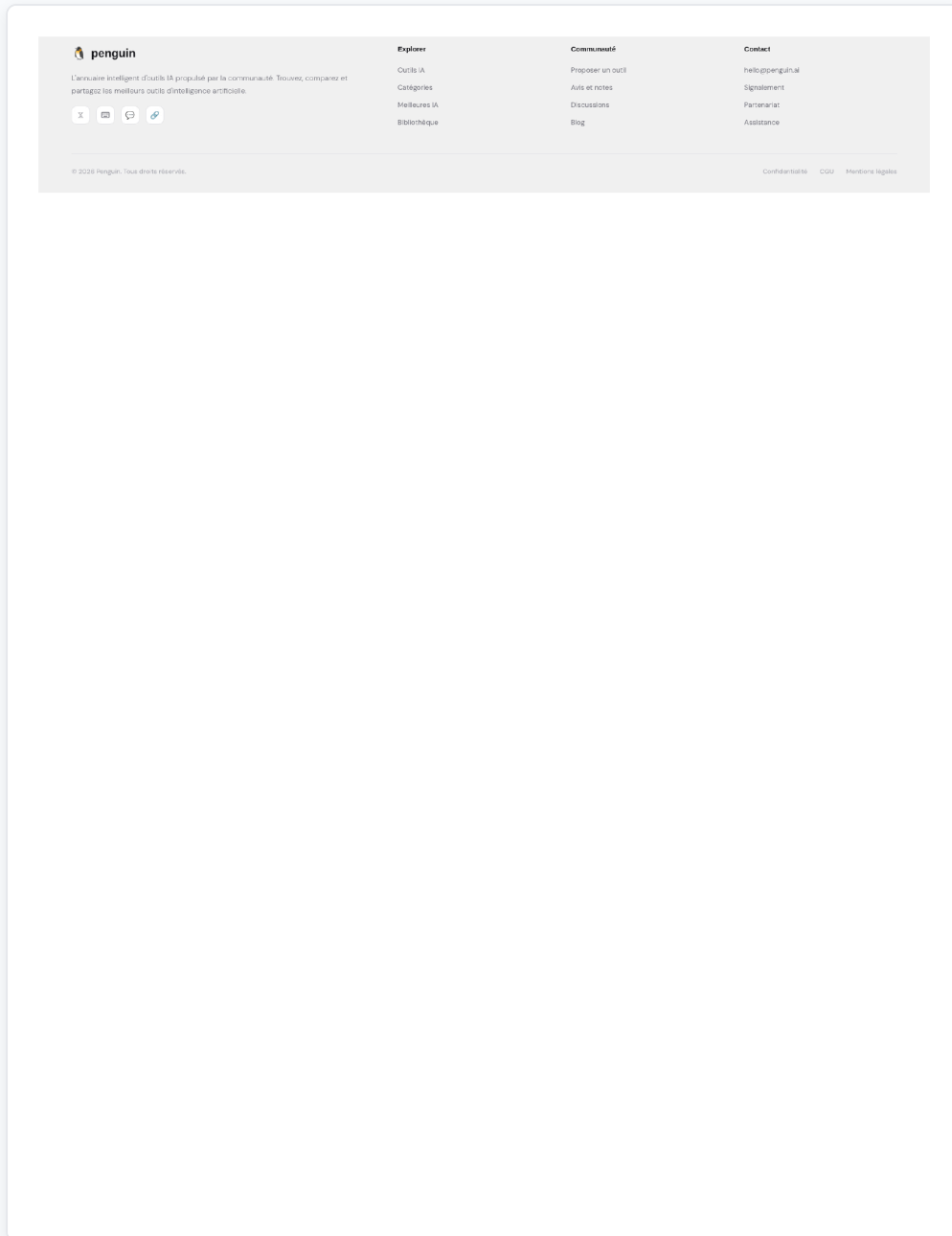
The Penguin Hall of Fame

Ranked by thousands of verified community reviews.



RANKED 4 - 8





4.2.2 Authentification (Connexion & Inscription) — AuthPage.jsx

Ce que voit l'utilisateur : Une interface split avec un panneau de formulaire à gauche et un panneau décoratif à droite (avec robot animé et bulle de dialogue "Salut, l'ami !" / "Bon retour !").

Interactions utilisateur :

- **Mode Connexion :** L'utilisateur saisit son email et mot de passe, puis clique "Se connecter". Le formulaire valide les champs côté client avant envoi. En cas d'erreur

(email incorrect, mot de passe invalide, compte suspendu/supprimé), un message d'erreur s'affiche en rouge.

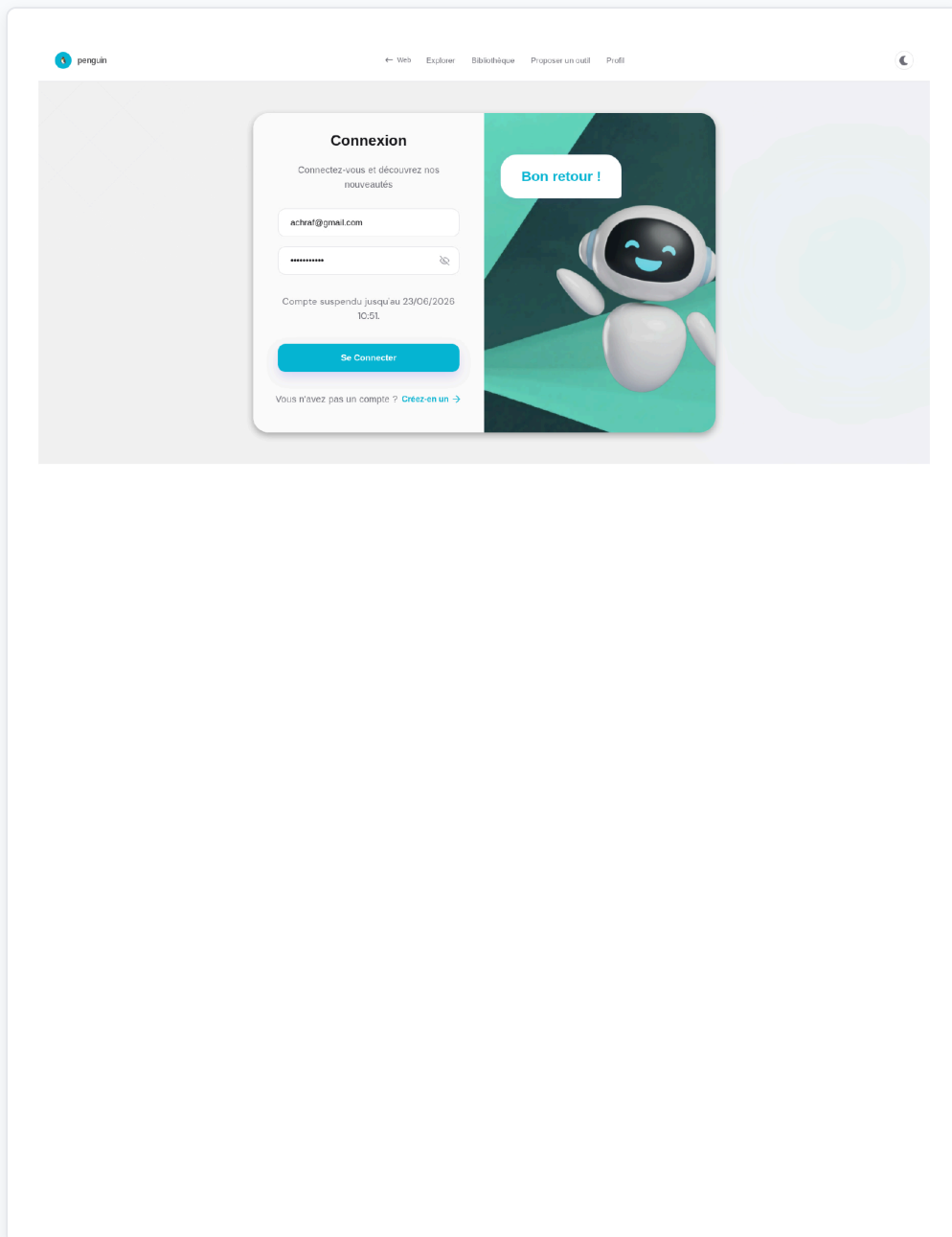
- **Mode Inscription** : L'utilisateur saisit son prénom, nom, email et mot de passe, puis clique "S'inscrire". Le formulaire vérifie le format de l'email et la force du mot de passe. Si l'email est déjà utilisé, une erreur 409 est affichée.
- **Basculement entre les modes** : L'utilisateur peut passer de connexion à inscription et vice-versa via un bouton de basculement. L'interface s'anime pour révéler le formulaire correspondant.
- **Redirection après connexion** : Une fois authentifié, le token JWT est stocké dans `localStorage` et l'utilisateur est redirigé vers la page d'accueil (ou la page qu'il visitait avant).



Inscription

Créer Votre Compte Gratuitement

Vous avez déjà un compte ? [Connectez-vous](#)



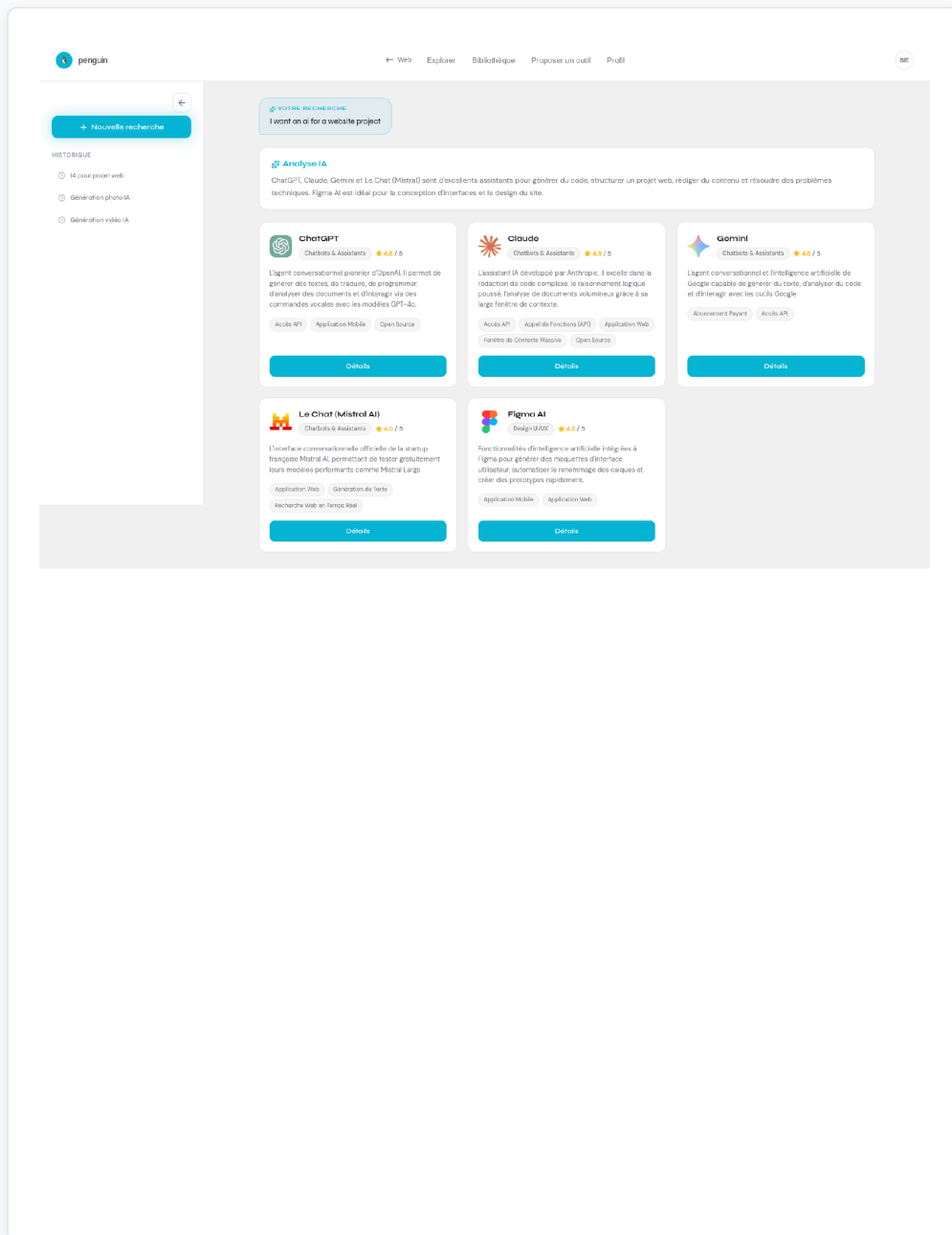
4.2.3 Recherche Intelligente (HomeSearch) — `HomeSearch.jsx`

Ce que voit l'utilisateur : Une interface moderne avec une barre de recherche centrale au design "glow" et une sidebar latérale coulissante pour l'historique.

Interactions utilisateur :

- **Saisie du prompt :** L'utilisateur tape une requête en langage naturel (ex: "I need to make a podcast", "outils pour générer des images") et appuie sur Entrée ou clique "Rechercher".

- **Analyse IA** : Un écran de chargement s'affiche avec le message "Penguin analyse votre requête..." et un spinner animé pendant que l'IA externe traite la demande.
- **Affichage des résultats** : Une bulle "Votre recherche" rappelle le prompt saisi. Un encart "Analyse IA" montre le titre et le raisonnement de l'IA. Les outils recommandés apparaissent sous forme de cartes (logo, nom, catégorie, note, description, fonctionnalités, bouton "Détails").
- **Navigation vers les détails** : L'utilisateur clique "Détails" sur une carte pour accéder à la fiche détaillée de l'outil.
- **Sidebar d'historique** : Pour les utilisateurs connectés, l'historique des recherches précédentes apparaît dans la sidebar avec une icône d'horloge. L'utilisateur peut cliquer sur un élément pour recharger instantanément les résultats (depuis le cache React ou via une requête API).
- **Nouvelle recherche** : L'utilisateur peut lancer une nouvelle recherche à tout moment via le bouton "Nouvelle recherche" avec l'icône "+".
- **Recherche via URL** : La page supporte les paramètres d'URL (`?q=prompt`) pour permettre les recherches depuis des liens externes.
- **Sidebar toggle** : Un bouton menu hamburger permet d'ouvrir/fermer la sidebar. Un bouton "+" flottant permet aussi de lancer une nouvelle recherche.



4.2.4 Fiche Détaillée d'un Outil (ToolDetails) — `toolDetailsPage.jsx`

Ce que voit l'utilisateur : Une page complète avec en-tête, grille de contenu principal, sidebar de métadonnées, section des avis avec filtres et outils similaires.

Interactions utilisateur :

- **En-tête (`ToolDetailsHeader`) :** Affiche le logo, le nom, la catégorie, la note globale sur 5, et les boutons d'action :
 - **Favori (Coeur) :** L'utilisateur clique sur l'icône cœur pour ajouter/retirer l'outil de ses favoris. L'icône se remplit/se vide instantanément.

- **Ajouter à une étagère** : Ouvre une modale (`ShelfPicker`) listant les étagères de l'utilisateur. L'utilisateur peut créer une nouvelle étagère ou basculer l'outil dans une étagère existante.
- **Contenu principal** (`ToolDetailsMain`) : Description détaillée de l'outil, fonctionnalités listées, modèles disponibles, URL du site web (ouvrable dans un nouvel onglet), date de sortie.
- **Sidebar** (`ToolDetailsSide`) : Informations complémentaires (fournisseur, catégorie, profession recommandée).
- **Section Métadonnées** : Affiche qui a créé/validé l'outil et les dates de création/modification.
- **Section Avis** :
 - **Compteur** : Indique le nombre total d'avis.
 - **Filtres par étoiles** : L'utilisateur peut filtrer les avis par nombre d'étoiles (5★, 4★, 3★, 2★, 1★ ou "Tous").
 - **Écrire un avis (si connecté)** : Bouton "Écrire un avis" ouvre un formulaire avec sélection de note (1-5 étoiles cliquables) et champ de commentaire optionnel avec compteur (1000 caractères max). Un avertissement indique que le commentaire sera modéré par l'IA.
 - **Envoi** : Si le commentaire est propre → publication immédiate avec message "Merci pour votre avis !". Si le commentaire est signalé → message "Votre avis est en cours de vérification".
 - **Modifier/Supprimer ses avis** : Boutons "Modifier" (icône crayon) et "Supprimer" (icône poubelle). La suppression demande une confirmation ("Confirmer la suppression ?" avec boutons Annuler/Supprimer).
 - **Statut de modération** : Les avis en attente affichent un badge "En modération" avec icône d'horloge.
 - **Pagination** : Si plus de 6 avis, un bouton "Voir plus (X avis supplémentaires)" permet d'afficher tous les avis.
- **Outils similaires** : Une section en bas de page avec des cartes d'outils apparentés, cliquables pour naviguer vers leur fiche (`<SimilarTools />`).
- **Retour** : Un bouton "Retour" (flèche gauche) en haut de page permet de revenir à la page précédente via `navigate(-1)` .
- Si l'outil n'existe pas, un message "Outil introuvable" s'affiche.

[← Retour](#)

4.80 / 5

À propos

L'agent conversationnel et l'intelligence artificielle de Google capable de générer du texte, d'analyser du code et d'interagir avec les outils Google.

Fonctionnalités

[Abonnement Payant](#) [Accès API](#)

Modèles supportés

[Gemini 2.0 Flash](#) [Gemini 2.0 Pro](#) [Gemini 2.0 Pro](#) [Gemini 2.0](#)

Métadonnées

CREATED BY
Super AdminVALIDATED BY
Super AdminCREATED AT
2028-06-12 10:07:01UPDATED AT
2028-06-12 10:07:01

Avis (1)

[Tous](#) [5e](#) [4e](#) [3e](#) [2e](#) [1e](#)

Yahya Elmoul ★★★★★
Consultant en Stratégie
18 juin 2028
you can't find like this masterpiece not in a million years

[Modifier](#) [Supprimer](#)

Outils Similaires

[ChatGPT](#) [View](#)[Claude](#) [View](#)[Le Chat \(Mistral AI\)](#) [View](#)[HuggingChat](#) [View](#)

Découvrez nos Meilleurs Outils IA

Rechercher par nom, fournisseur, ou catégorie...

TRIER

Trier par...

CATÉGORIES

Toutes les catégories 16

Outils Développeurs 2

Recherche & Données 3

Design UI/UX 1

Vidéo & Animation 1

Chatbots & Assistants 5

Génération d'images 2

Audio & Voix 1

15 outils trouvés

**DeepSeek**
BY DEEPSEEK
★ 4.30

Accès API Fenêtre de Contexte Massives
Génération de Texte Open WebUIs
Podcastement Complet

Détails →

**Cohere Studio G API**
BY COHERE
★ 4.30

Accès API Fenêtre de Contexte Massives
Génération de Texte Support de la RAG

Détails →

**Figma AI**
BY FIGMA
★ 4.30

Application Mobile Application Web

Détails →

**Veo**
BY GOOGLE
★ 4.60

Application Web Génération d'images
Génération de Vidéo

Détails →

**NotebookLM**
BY GOOGLE
★ 4.30

Application Web
Fenêtre de Contexte Massives
Génération de Texte Génération de Vidéo
Recherche Web en Temps Réel

Détails →

**HuggingChat**
BY HUGGING FACE
★ 4.20

Application Web Open Source

Détails →

**Stable Diffusion**
BY STABILITY AI
★ 4.50

Génération d'images Génération de Vidéo

Détails →

**Le Chat (Mistral AI)**
BY MISTRAL AI
★ 4.00

Application Web Génération de Texte
Recherche Web en Temps Réel

Détails →

**Meta Llama**
BY META
★ 4.50

Fenêtre de Contexte Massives Open Source

Détails →

**Perplexity**
BY PERPLEXITY
★ 4.80

Accès API Recherche Web en Temps Réel
Upload & Analyse de Fichiers

Détails →

**ElevenLabs**
BY ELEVENLABS
★ 4.70

Accès API Appel de Fonctions (API)
Application Web Entrée Voix / Audio

Détails →

**Claude**
BY ANTHROPIC
★ 4.80

Accès API Appel de Fonctions (API)
Application Web
Fenêtre de Contexte Massives Open Source

Détails →

**Midjourney**
BY MIDJOURNEY
★ 4.50

Fenêtre de Contexte Massives
Génération d'images
Vision & Analyse d'images

Détails →

**ChatGPT**
BY OPENAI
★ 4.80

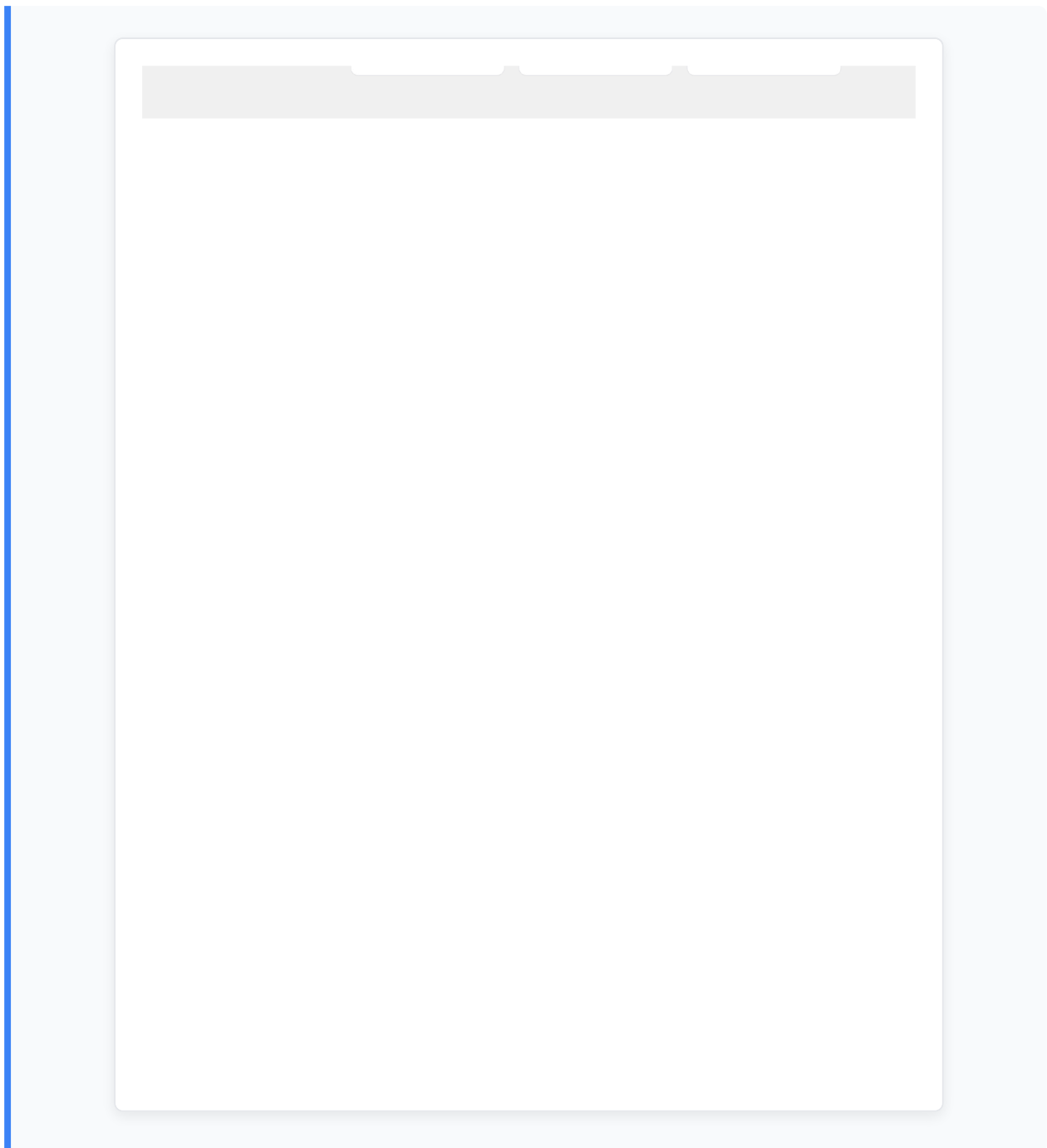
Accès API Application Mobile
Open Source

Détails →

**Gemini**
BY GOOGLE
★ 4.60

Abonnement Payant Accès API

Détails →



4.2.5 Bibliothèque Personnelle — `Library.jsx`

Ce que voit l'utilisateur : Un espace personnel avec deux onglets principaux, des statistiques et une interface organisée.

Interactions utilisateur :

- **Barre de statistiques (StatsStrip) :** Trois cartes animées affichent le nombre de favoris, d'étagères et le total d'outils sauvegardés.
- **Titre :** "Ma Bibliothèque" avec le sous-titre "Vos outils sauvegardés, organisés selon votre workflow."

- **Onglet "Favoris"** : Affiche la grille des outils favoris de l'utilisateur. L'utilisateur peut cliquer sur un outil pour voir ses détails ou retirer des favoris via l'icône coeur. Badge indiquant le nombre de favoris.
- **Onglet "Mes Étagères"** : Liste les étagères personnelles de l'utilisateur. Chaque étagère est nommée (ex: "Audio Editing", "Video Production"). L'utilisateur peut :
 - Cliquer sur une étagère pour voir son contenu détaillé (route `/Library/Shelf/:shelfIndex` via le composant `ShelfItems`).
 - Créer une nouvelle étagère.
 - Modifier ou supprimer une étagère existante.
 - Badge indiquant le nombre d'étagères.
- **Navigation** : L'utilisateur bascule entre les onglets via des boutons avec icônes (❤️ pour Favoris, 📁 pour Étagères).
- **Accès restreint** : Si l'utilisateur n'est pas connecté, un message "Connectez-vous pour accéder à votre bibliothèque" s'affiche avec un bouton de redirection vers la page d'authentification.

ESPACE PERSONNEL

Ma Bibliothèque

Vos outils sauvegardés, organisés selon votre workflow.

Favoris 7

Mes Étagères 4

Favoris

Étagères

Total Outils

 **Meta Llama** 

Le modèle de langage open-source le plus massif et performant de Meta, rivalisant directement avec les meilleurs modèles propriétaires du marché.

★ 4.50 [Outils Développeurs](#)

 **Claude** 

L'assistant IA développé par Anthropic. Il excelle dans la rédaction de code complexe, le raisonnement logique poussé, l'analyse de documents volumineux grâce à sa large fenêtre de contexte.

★ 4.80 [Chatbots & Assistants](#)

 **Gemini** 

L'agent conversationnel et l'intelligence artificielle de Google capable de générer du texte, d'analyser du code et d'interagir avec les outils Google.

★ 4.60 [Chatbots & Assistants](#)

 **ChatGPT** 

L'agent conversationnel pionnier d'OpenAI. Il permet de générer des textes, de traduire, de programmer, d'analyser des documents et d'interagir via des commandes vocales avec les modèles GPT-4o.

★ 4.80 [Chatbots & Assistants](#)

 **Perplexity** 

Un moteur de recherche conversationnel qui répond directement aux questions des utilisateurs en explorant le web en temps réel et en citant systématiquement ses sources de manière transparente.

★ 4.90 [Recherche & Données](#)

 **Veo** 

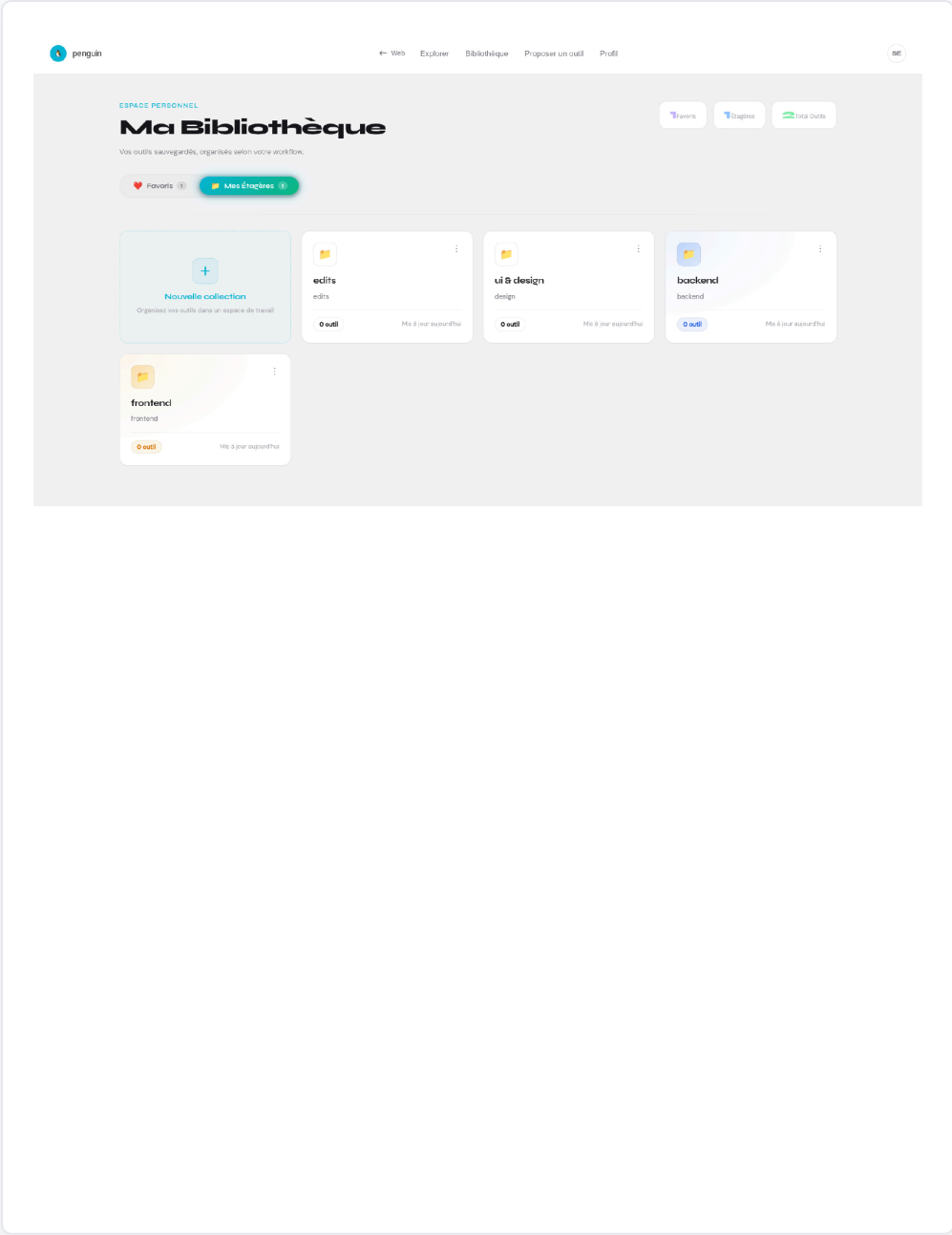
Le modèle de génération de vidéo le plus avancé de Google, capable de créer des clips cinématographiques haute qualité en 1080p à partir de invites textuelles.

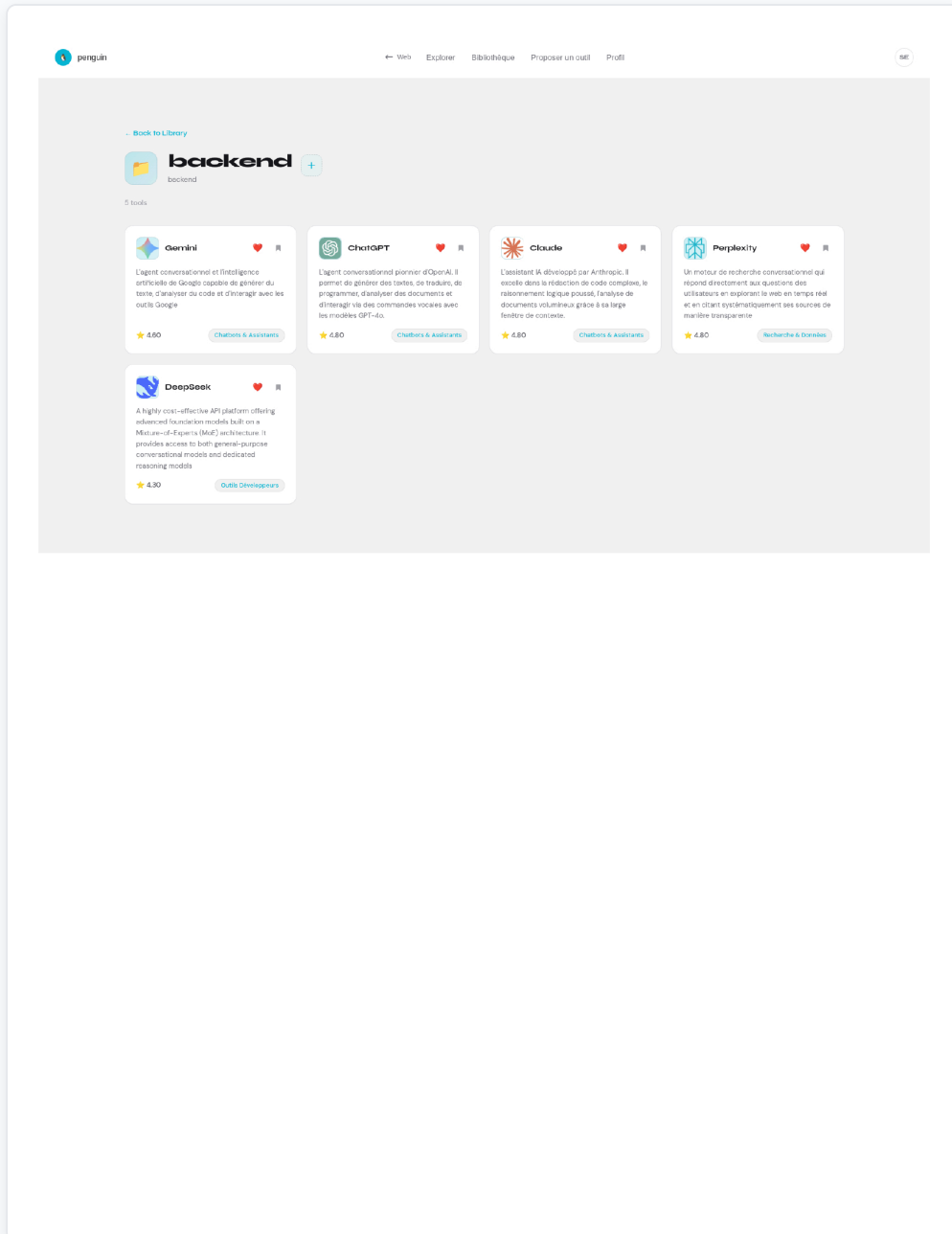
★ 4.60 [Vidéo & Animation](#)

 **DeepSeek** 

A highly cost-effective API platform offering advanced foundation models built on a Mixture-of-Experts (MoE) architecture. It provides access to both general-purpose conversational models and dedicated reasoning models.

★ 4.30 [Outils Développeurs](#)





4.2.6 Profil Utilisateur — **Profile.jsx**

Ce que voit l'utilisateur : Un tableau de bord personnel complet avec en-tête de profil et une grille de 6 sections.

Interactions utilisateur :

- **En-tête de profil (**ProfileHeader**) :** Photo de profil, nom, email, profession et date d'inscription.
- **Statistiques et Insights (**Insights**) :** Chiffres clés sur l'activité de l'utilisateur : nombre d'avis rédigés, d'outils favoris, de recherches effectuées, de suggestions

soumises.

- **Grille de 6 sections :**

- **Mes Avis ([Reviews](#))** : Liste chronologique de tous les avis rédigés, avec le nom de l'outil, la note (étoiles), le commentaire et le statut (approuvé ✓, en modération ⏳). Boutons "Modifier" et "Supprimer" par avis.
- **Mes Favoris ([Favorites](#))** : Aperçu des 6 derniers outils favoris sous forme de mini-cartes cliquables. Lien "Voir tout" vers la bibliothèque.
- **Mes Étagères ([Shelves](#))** : Liste des étagères personnelles avec le nombre d'outils par étagère. Lien "Gérer les étagères" vers la bibliothèque.
- **Historique de Recherche ([SearchHistory](#))** : Liste des dernières recherches IA effectuées (date, prompt). Possibilité de cliquer pour relancer la recherche ou d'effacer l'historique.
- **Mes Suggestions ([Suggestions](#))** : Historique des outils proposés avec leur statut (en attente, approuvé et publié, refusé avec motif). Lien vers la page "Proposer un outil".
- **Détails du Compte ([AccountDetails](#))** : Formulaire pour modifier le nom, l'email, la profession, la photo de profil (upload), et le mot de passe (avec champ confirmation). Section "Supprimer mon compte" avec avertissement de la période de grâce de 30 jours.

- **Accès restreint** : Si l'utilisateur n'est pas connecté, un LoginPrompt l'invite à se connecter.

47%

Yahya Elmaoui

ye1maoui@gmail.com

manager active Consultant en Stratégie

13 days active
Joined June 2026

★

6

Favorites

📚

4

Shelves

👍

4

Reviews

🔍

6

Searches

💡

1

Suggestions

👍 Reviews (4)

FlowerLabs

★★★★★

this is one of the best

Jun 11, 2023

Claude

★★★★★

if you want to code that is the best choice

Jun 12, 2023

NotebookLM

★★★★★

this one help me sooooo much

Jun 12, 2023

ChatGPT

★★★★★

best app ever

Jun 12, 2023

★ Favorites (6)

Claude

ChatGPTs & Assistants

4.5

Gemini

ChatGPTs & Assistants

4.5

ChatGPT

ChatGPTs & Assistants

4.5

Veo

Video & Animation

4.5

Figure AI

Design UI/UX

4.5

Midjourney

Generation (Images)

4.5

📚 My Shelves (4)

music

No description

books

Updated Today

ui & ux design

No description

books

Updated Today

backend

for backend

books

Updated Today

Frontend

a short for Frontend etc

books

Updated 6d ago

🔍 Search History (6)

Génération d'images IA

Find me an ai that can help me design an image

AI Prompt

Jun 16

Génération de site web

I want an ai to generate a website

AI Prompt

Jun 15

Assistants IA pour études

I want an ai to help me with my study

AI Prompt

Jun 15

Idéation site web IA

I want to make a website and I need an ai to talk with about the idea

AI Prompt

Jun 15

Aide aux études IA

I want an ai to help me with my study

AI Prompt

Jun 15

Idéation site web

I want to make a website and I need an ai to talk with about the idea

AI Prompt

Jun 15

💡 My Suggestions (1)

ChatGPT

chatgpt.com/

L'agent conversationnel prioriter d'Opera. Il permet de générer des textes, de traduire, de programmer, d'analyser des documents et d'interagir via des commandes vocales avec les modèles GPT-4o.

Category ChatGPTs & Assistants

Provider OpenAI

Joined Jun 2026

⚙️ Account Details

edit

FULL NAME

Yahya Elmaoui

EMAIL ADDRESS

ye1maoui@gmail.com

ROLE

manager

ACCOUNT STATUS

active

PROFESSION

Consultant en Stratégie

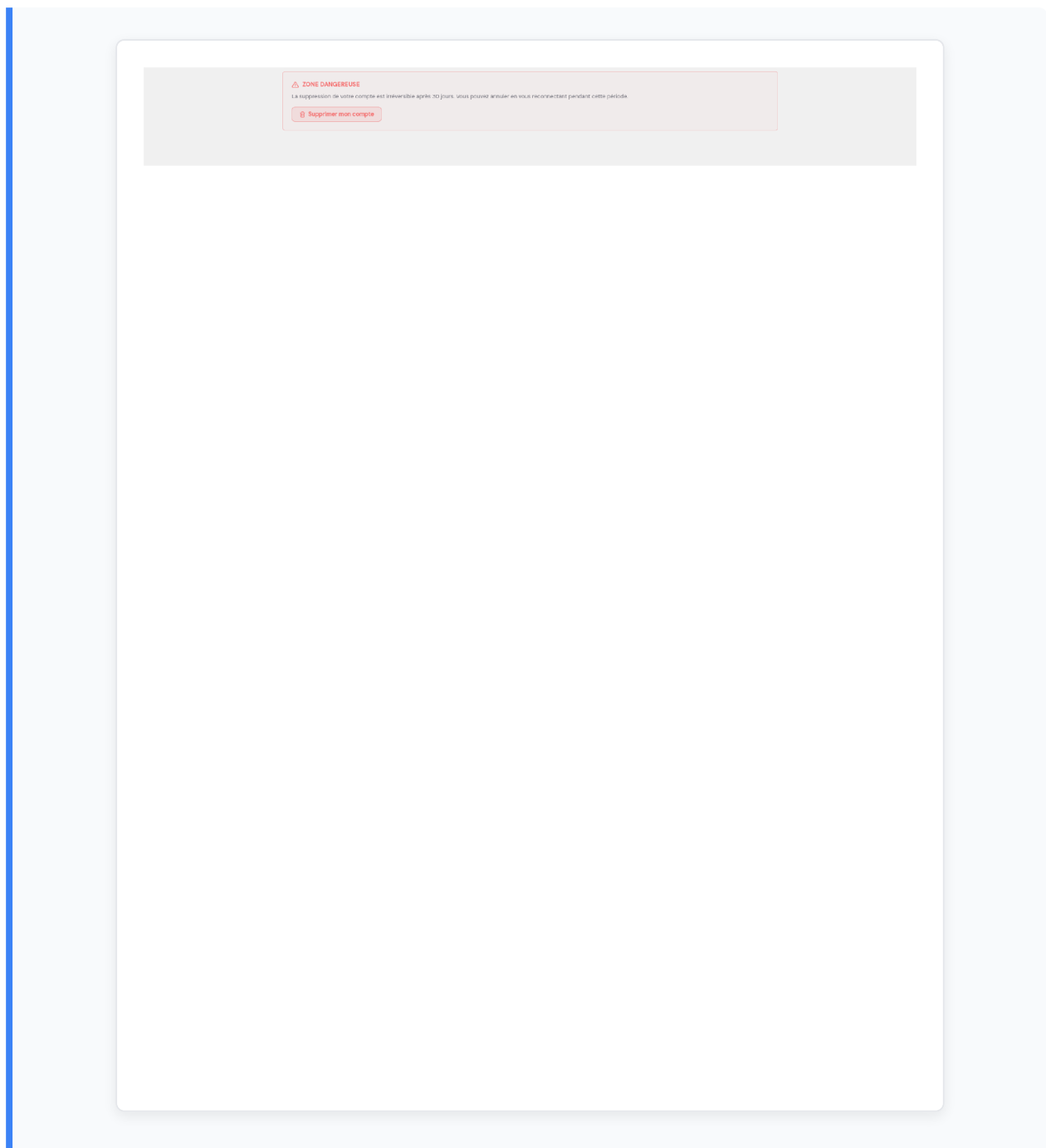
PASSWORD

MEMBER SINCE

June 5, 2026

LAST UPDATED

June 16, 2026



4.2.7 Proposition d'Outil — `ProposerOutils.jsx`

Ce que voit l'utilisateur : Un assistant en 4 étapes (Stepper) avec un formulaire complet et une sidebar d'historique des soumissions.

Interactions utilisateur :

- **Sidebar d'historique :** Liste de toutes les suggestions précédentes avec leur statut (Analyse IA..., En attente admin, Refusé, Publié ✓). Chaque entrée a un code couleur (orange pour en attente, vert pour accepté, rouge pour refusé). L'utilisateur clique sur une suggestion pour voir son détail.

- **Stepper visuel** : Barre de progression à 4 étapes (Informations → Analyse IA → Validation → Résultat) avec numéros, titres et sous-titres. L'étape courante est mise en évidence.
- **Étape 1 — Formulaire complet** :
 - **Logo** : Zone de glisser-déposer avec icône cloud, texte "Glissez-déposez une image ou cliquez pour parcourir", hint "PNG, JPG ou WebP (max. 5MB)".
 - **Nom** : Champ texte avec placeholder "Ex : ChatGPT".
 - **URL** : Champ URL avec placeholder "<https://exemple.com>".
 - **Description** : Zone de texte avec compteur (500 caractères max).
 - **Catégorie** : `CreatableSelect` — sélection dans une liste déroulante avec possibilité de taper une nouvelle catégorie.
 - **Fonctionnalités** : `CreatableSelect` multi-sélection avec création possible de nouvelles fonctionnalités.
 - **Fournisseur** : `CreatableSelect` avec création possible.
 - **Date de sortie** : Input de type date.
 - **Modèle** : `CreatableSelect` multi-sélection avec création possible.
 - **Pourquoi cet outil ?** : Zone de texte avec compteur (500 caractères max).
 - **Boutons** : "Réinitialiser" (efface le formulaire) et "Soumettre" (envoi pour analyse).
- **Étape 2 — Analyse IA** : Affiche le logo de l'outil, son nom, un badge "Analyse IA en cours..." et un spinner animé.
- **Étape 3 — En attente admin** : Badge "En attente de validation". Message "Votre soumission a été vérifiée par l'IA et est en attente de validation par un administrateur." Affiche les notes de modération IA si disponibles. Lien URL et date de soumission.
- **Étape 4 — Résultat final** : Si accepté : badge "Accepté et publié" vert avec message de confirmation. Si refusé : badge "Refusé" rouge avec la raison du refus affichée dans une boîte d'alerte.
- **Nouvelle soumission** : Bouton "Nouvelle soumission" en bas de page pour recommencer.
- **Anti-spam** : Protection intégrée — une seule suggestion en attente à la fois.

Proposer un outil IA

1 Informations 2 Analyse IA 3 Validation 4 Résultat

1 Informations

Détails de l'outil

2 Analyse IA

Vérification automatique

3 Validation

En attente admin

4 Résultat

Final

Image *

Obtenir et déposer une image ou [cliquez pour parcourir](#)
(PNG, JPG ou WebP (max. 5MB))

Name *

Ex: ChatGPT

Url *

https://example.com

Le nom officiel de l'outil

Le lien officiel vers l'outil

Description *

Décrivez brièvement cet outil et son utilité

Category *

Sélectionnez...

Features *

Ajoutez des fonctionnalités...

Provider *

Ex: OpenAI

Released date *

mm/dd/yyyy

Model

Ex: GPT-4

Why this tool? *

Pourquoi cet outil mérite d'être dans notre annuaire ?

Régénérer

Soumettre

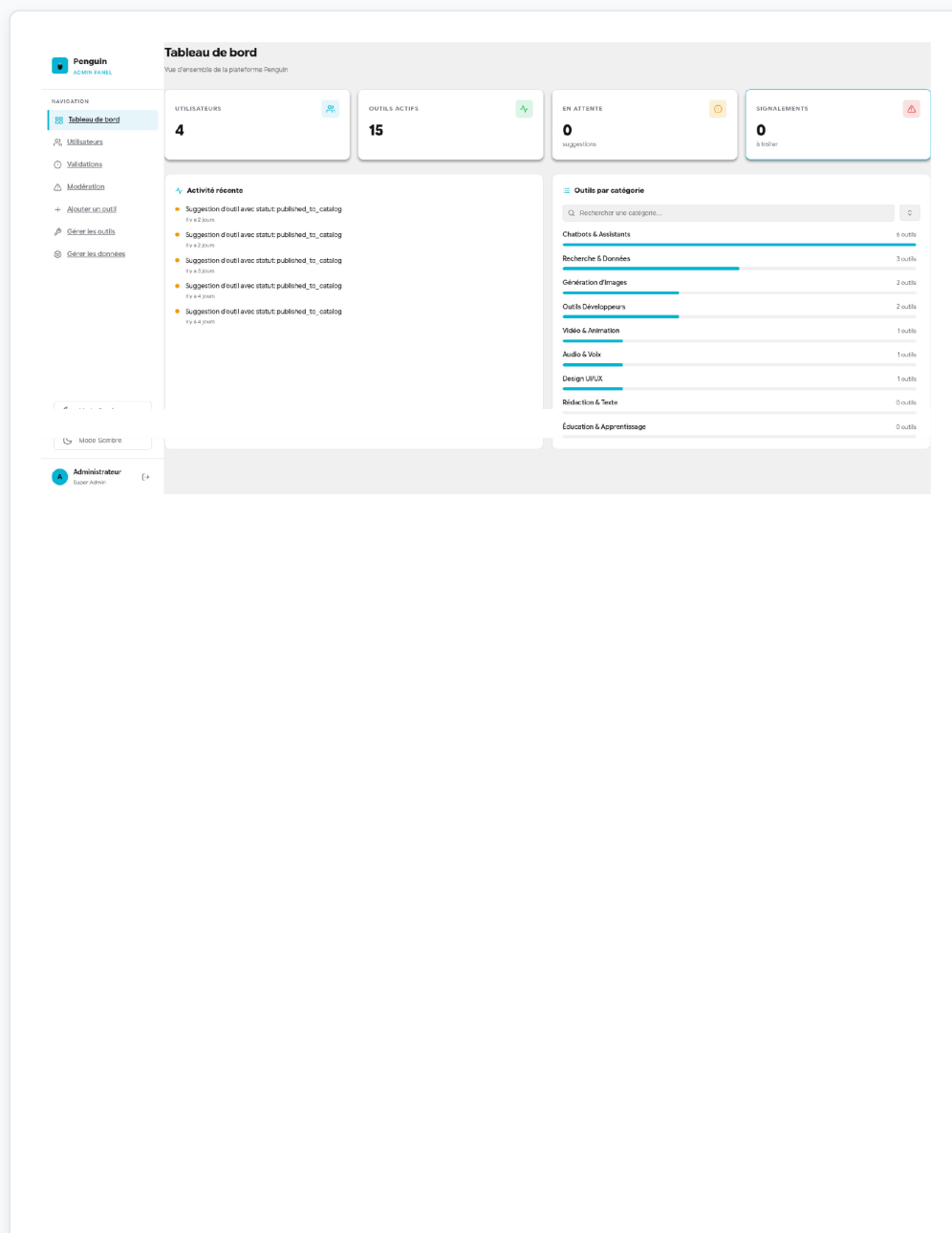
4.3 Interface d'Administration

L'espace d'administration est accessible uniquement aux utilisateurs avec les rôles `admin` ou `manager`. L'interface utilise `AdminSidebar.jsx` comme navigation latérale avec des icônes et `Outlet` pour afficher le contenu.

4.3.1 Dashboard — AdminDashboard.jsx

Ce que voit l'administrateur : Vue d'ensemble avec des indicateurs clés présentés sous forme de cartes statistiques : nombre d'outils actifs, d'utilisateurs inscrits, de suggestions en attente de validation, d'avis signalés (flagged). Graphiques ou métriques de performance de la plateforme.

Interactions : Vue synthétique — pas d'action directe, mais des liens vers les sections détaillées.

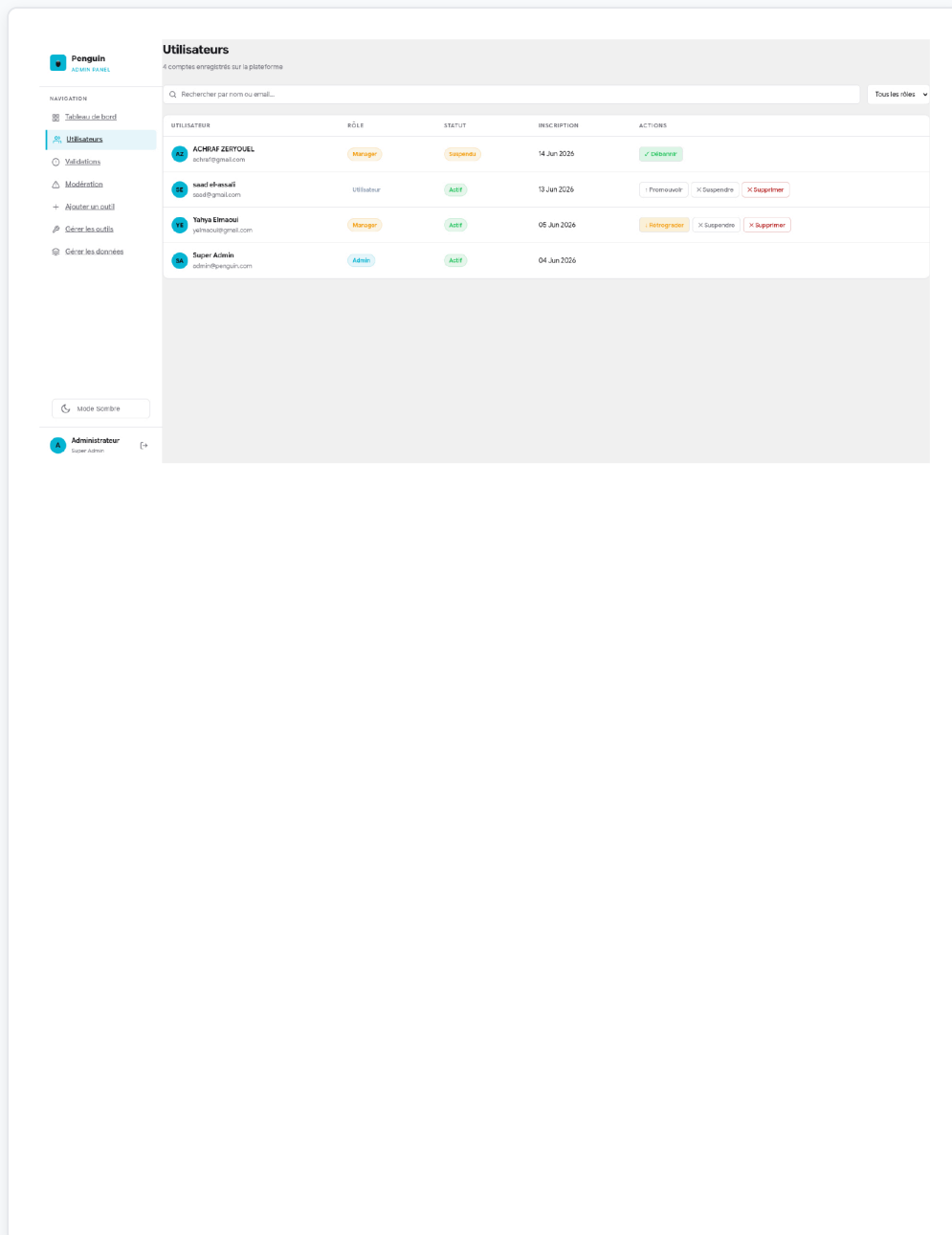


4.3.2 Gestion des Utilisateurs — `AdminUsers.jsx`

Ce que voit l'administrateur : Un tableau complet avec toutes les colonnes : nom, email, rôle (user/manager/admin), statut (actif/suspendu/supprimé), profession, date d'inscription, date de suspension.

Interactions utilisateur :

- **Promouvoir/Dégrader :** Changer le rôle d'un utilisateur (user → manager, manager → admin, ou l'inverse).
- **Suspendre :** Ouvre un formulaire avec sélection de durée (24h, 7j, 30j, 1 an, définitif). Une fois suspendu, le badge passe en orange "Suspendu".
- **Débannir :** Restaurer un compte suspendu. Le badge repasse en vert "Actif".
- **Supprimer définitivement :** Avec confirmation dialog. Le badge passe en rouge foncé "Supprimé".
- **Recherche textuelle :** Barre de recherche pour filtrer les utilisateurs par nom ou email.
- **Filtres :** Filtres par statut (tous, actif, suspendu, supprimé) et par rôle.



4.3.3 Validations des Suggestions — `Validations.jsx`

Ce que voit le manager : Une interface complète de validation avec header, statistiques, barre d'outils, liste principale et un drawer de détail.

Interactions utilisateur :

- **Header :** Titre "Validations" et description de la section.
- **Statistiques (Insight) :** Quatre cartes : Total des suggestions, En attente (badge orange), Acceptées (badge vert), Refusées (badge rouge).

- **Barre d'outils (ToolbarFilter) :**
 - Recherche textuelle par nom d'outil.
 - Filtre par statut (tous, en attente, acceptés, refusés).
 - Tri (plus récent, plus ancien).
- **Liste principale (Main) :** Tableau des suggestions avec nom, statut (avec code couleur), date de soumission, soumetteur.
- **Drawer de détail (`SuggestionDrawer`) :** Panneau latéral qui s'ouvre au clic sur une suggestion, affichant :
 - Toutes les informations soumises : logo, nom, URL, description, catégorie, fournisseur, fonctionnalités, modèle, date de sortie, justification.
 - Notes de modération IA et score IA.
 - **Bouton "Approuver"** : Valide la suggestion et crée automatiquement l'outil dans le catalogue. Résout les références mixtes (nouveaux fournisseurs/fonctionnalités créés automatiquement).
 - **Bouton "Rejeter"** : Ouvre un champ texte pour saisir le motif du refus.
 - La suggestion est déplacée visuellement de la liste "En attente" vers "Acceptée" ou "Refusée".

Soumissions d'outils IA

16

Total soumissions

0

En attente

16

Acceptées

0

Rejetées

Rechercher un outil, un auteur...

Tous

En attente

Acceptées

Rejetées

Plus récent

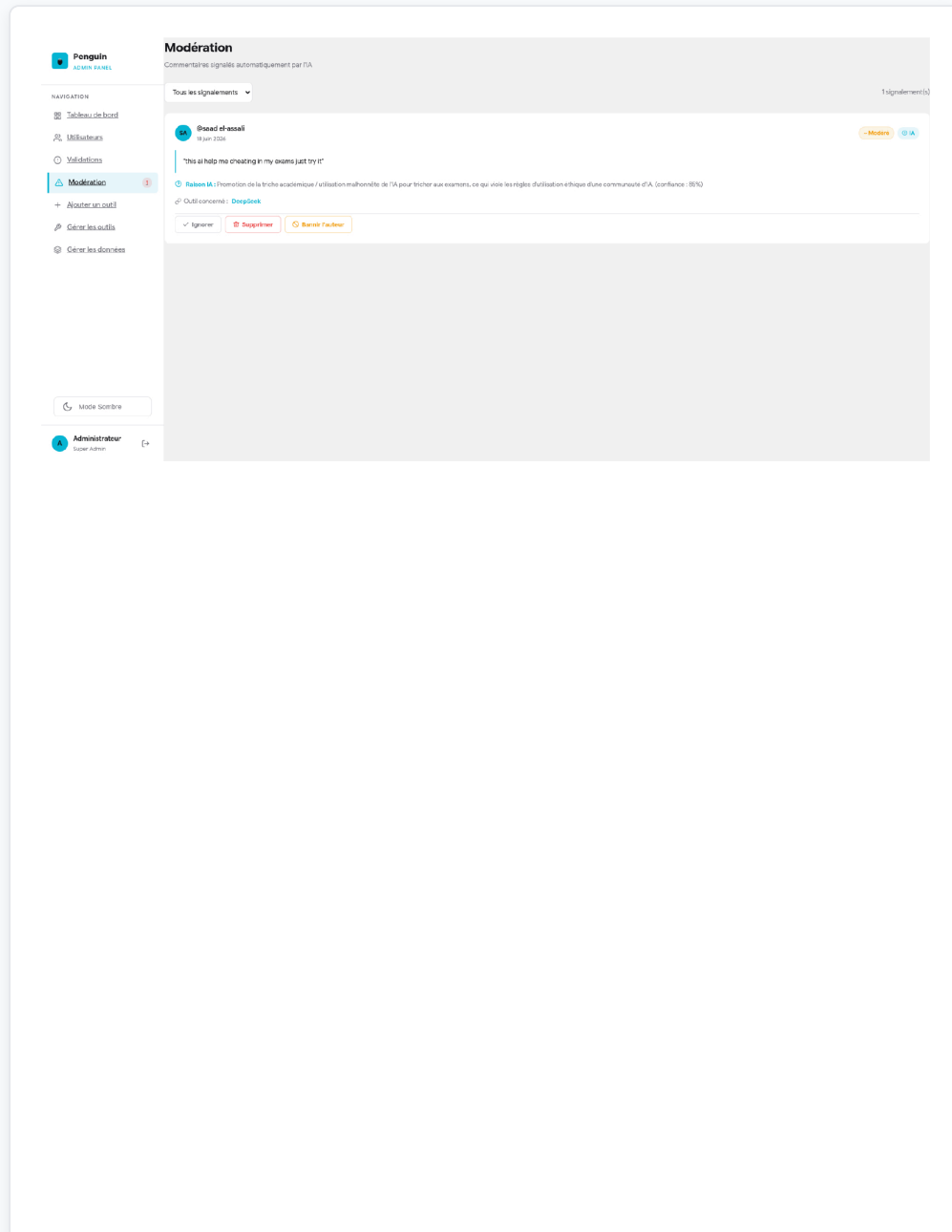
OUTIL	CATÉGORIE	SOURCES PAR	DATE	STATUT	ACTIONS
 Grok https://grok.x.ai	Chatbots & Assistants	@saad el-ssali	16 juin 2025	Acceptée	
 devseek https://x.com/devseek.com	Outils Développeurs	@ACHRAF ZERYOUEL	16 juin 2025	Acceptée	
 Cohere Studio & API https://cohere.com	Recherche & Données	@Super Admin	14 juin 2025	Acceptée	
 Figma AI https://figma.com	Design UI/UX	@saad el-ssali	13 juin 2025	Acceptée	
 Veo https://labs.google/veo/veo/veo	Vidéo & Animation	@Super Admin	13 juin 2025	Acceptée	
 NotebookLM https://notebooklm.google/	Éducation & Apprentissage	@Super Admin	12 juin 2025	Acceptée	
 Hugging Chat https://huggingface.co/chat/	Outils Développeurs	@Super Admin	12 juin 2025	Acceptée	
 Stable Diffusion https://stablediffusion.com	Génération d'images	@Super Admin	12 juin 2025	Acceptée	
 Mistral https://mistral.ai	Rédaction & Texte	@Super Admin	12 juin 2025	Acceptée	
 Llama https://llama.meta.com/	Outils Développeurs	@Super Admin	12 juin 2025	Acceptée	
 Perplexity https://www.perplexity.ai/	Recherche & Données	@Super Admin	12 juin 2025	Acceptée	
 ElevenLabs https://elevenlabs.ai	Audio & Voix	@Super Admin	12 juin 2025	Acceptée	
 Claude https://claude.ai	Outils Développeurs	@Super Admin	12 juin 2025	Acceptée	
 Midjourney https://www.midjourney.com/	Design UI/UX	@Super Admin	12 juin 2025	Acceptée	
 ChatGPT https://openai.com	Chatbots & Assistants	@Valéry Elmasou	12 juin 2025	Acceptée	

Précédent

Page 1 sur 2

Suivant

- **Action "Supprimer et bannir"** : Supprime l'avis offensant et suspend définitivement l'utilisateur (`status = 'suspended'` , `suspended_until = NULL`). L'utilisateur ne peut plus se connecter.
- L'interface se met à jour en temps réel : l'avis disparaît de la liste et un message de confirmation s'affiche.



4.3.5 Gestion des Outils — `AdminTools.jsx` & `AdminAddTool.jsx`

Ce que voit l'administrateur :

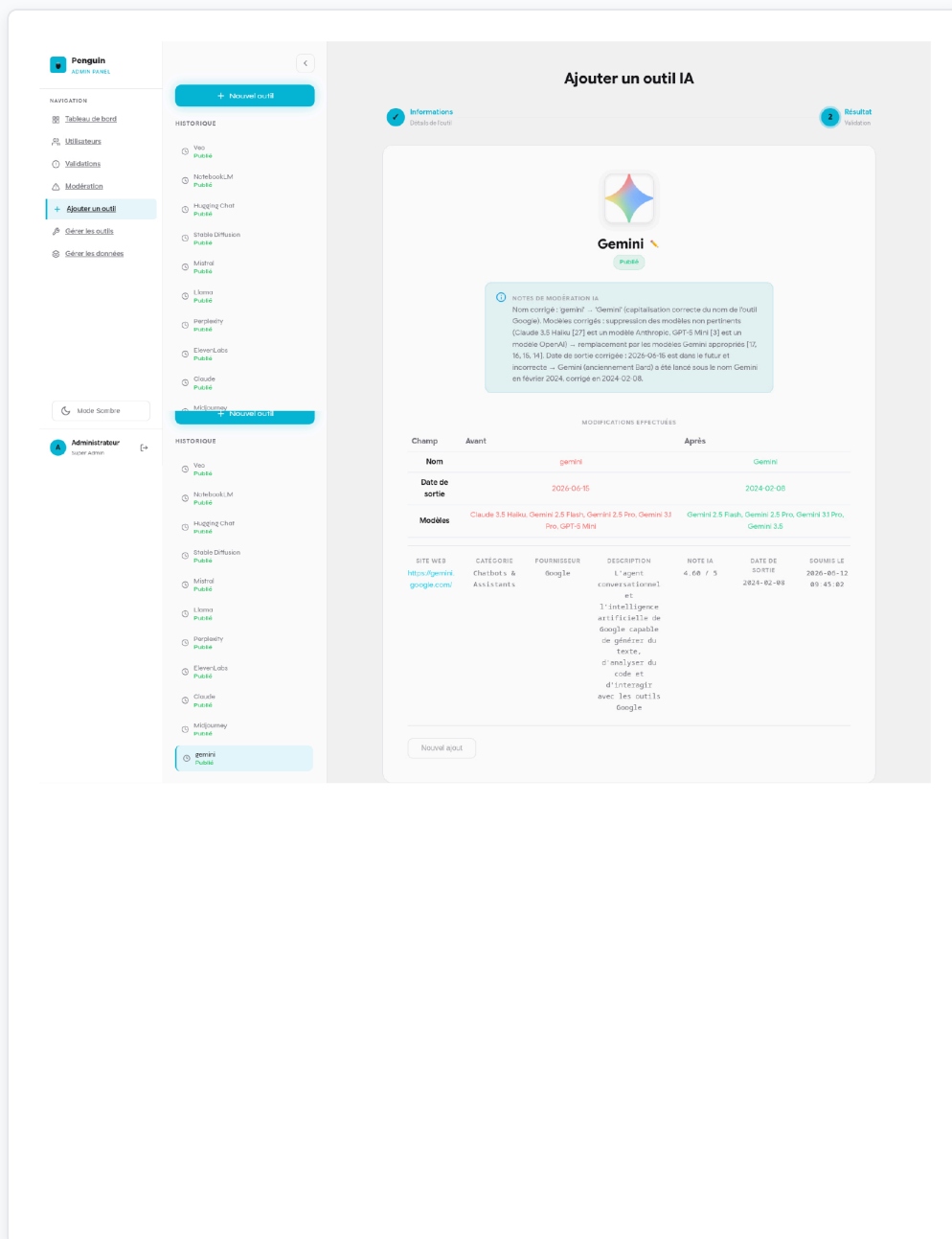
- **AdminTools.jsx** : Tableau de tous les outils (nom, catégorie, fournisseur, statut, note, date). Possibilité de filtrer par statut (tous, actif, en attente, archivé, obsolète). Actions : Changer le statut (activer/désactiver → 'active'/'pending', archiver → 'archived', déprécier → 'deprecated'), Supprimer.
- **AdminAddTool.jsx** (47KB) : Formulaire complet d'ajout manuel avec tous les champs : logo (upload), nom, URL, description, catégorie, fournisseur, fonctionnalités (multi-sélection), modèles, date de sortie, statut. Validation côté client complète avant soumission.

Gérer les Outils

Filterer par nom ou statut...

TOTAL AIS: 16 | ACTIFS: 15 | ARCHIVÉS: 1 | DÉPRÉCIÉS: 0

NOM	CATÉGORIE	FOURNISSEUR	STATUT	ACTIONS
Grok	Chatbots & Assistants	xAI	Archivé	[Icones]
DeepSeek	Outils Développeurs	DeepSeek	Actif	[Icones]
Cohere Studio & API	Recherche & Données	Cohere	Actif	[Icones]
Figma AI	Design UI/UX	Figma	Actif	[Icones]
Veo	Video & Animation	Google	Actif	[Icones]
NotebookLM	Recherche & Données	Google	Actif	[Icones]
HuggingChat	Chatbots & Assistants	Hugging Face	Actif	[Icones]
Stable Diffusion	Génération d'Images	Stability AI	Actif	[Icones]
Le Chat (Mistral AI)	Chatbots & Assistants	Mistral AI	Actif	[Icones]
Meta Llama	Outils Développeurs	Meta	Actif	[Icones]
Perplexity	Recherche & Données	Perplexity	Actif	[Icones]
ElevenLabs	Audio & Voix	ElevenLabs	Actif	[Icones]
Claude	Chatbots & Assistants	Anthropic	Actif	[Icones]
Midjourney	Génération d'Images	Midjourney	Actif	[Icones]
ChatGPT	Chatbots & Assistants	OpenAI	Actif	[Icones]
Gemini	Chatbots & Assistants	Google	Actif	[Icones]



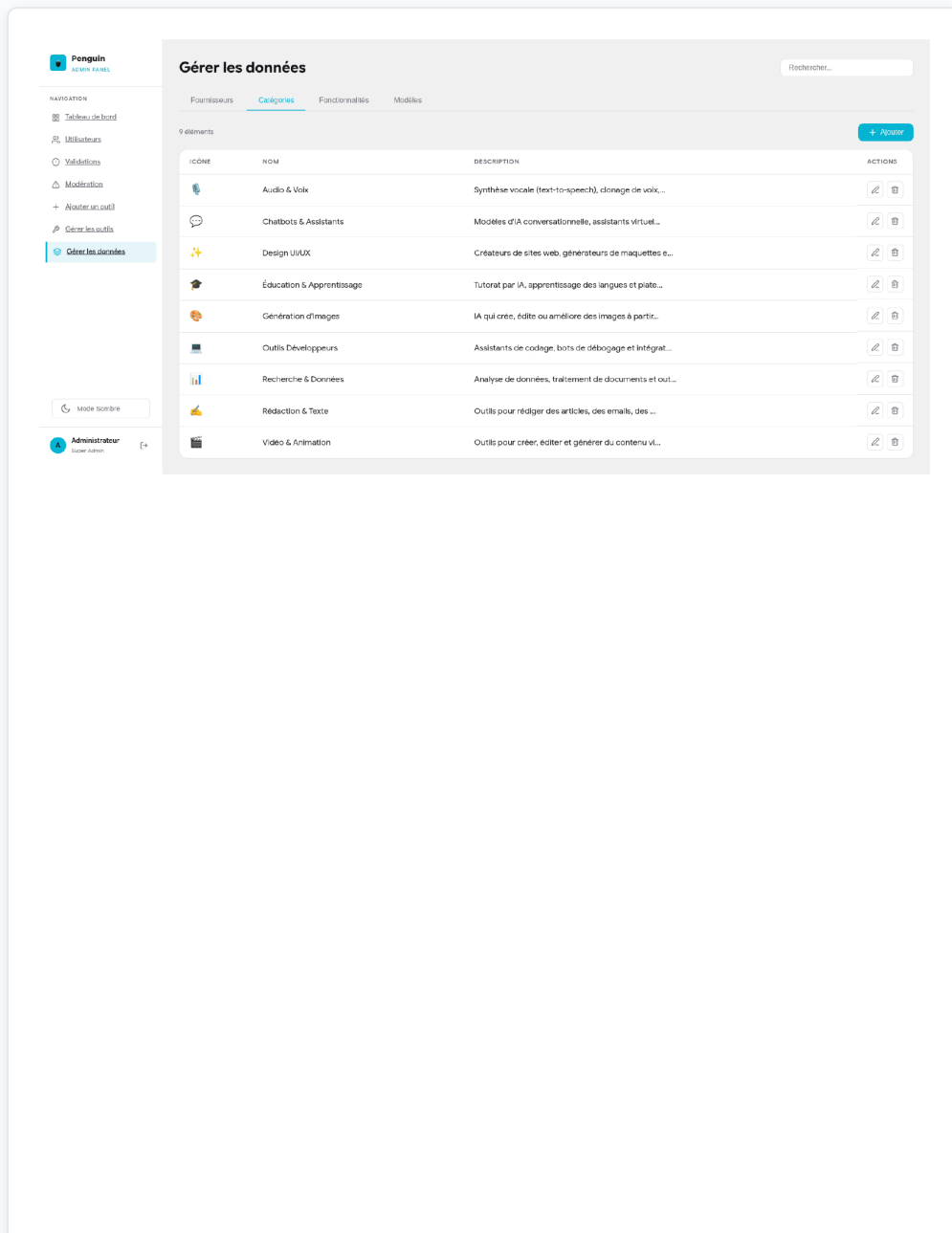
4.3.6 Gestion des Entités (Référentiels) — `AdminEntityManager.jsx`

Ce que voit l'administrateur : Interface à onglets pour gérer les données de référence de la plateforme.

Interactions utilisateur :

- **Onglet "Catégories" :** Tableau des catégories (nom, icône, description). Ajouter (modal avec formulaire), Modifier (modal pré-rempli), Supprimer (avec confirmation).
- **Onglet "Fournisseurs" :** Tableau des fournisseurs (nom, site web, statut). Ajouter, Modifier, Supprimer.

- **Onglet "Fonctionnalités"** : Tableau des fonctionnalités (nom, description, type, statut). Ajouter, Modifier, Supprimer.
- **Onglet "Modèles"** : Tableau des modèles d'IA (nom, description, statut, fournisseur associé). Ajouter, Modifier, Supprimer.
- Toutes les modifications se font via des modales avec validation. Le tableau se met à jour instantanément après chaque action.



4.4 Architecture et Technologies Backend

Le Backend de Penguins est une **API RESTful** développée entièrement en **PHP 8+** sans framework externe, en respectant strictement le patron **MVC (Modèle-Vue-Contrôleur)** avec une couche Services supplémentaire pour la logique métier.

4.4.1 Fonctionnement du Front Controller (`index.php`)

Le fichier `index.php` à la racine de `backend/` agit comme **Front Controller** unique. Toutes les requêtes HTTP sont redirigées vers ce fichier. Il analyse l'URL demandée, instancie dynamiquement le contrôleur approprié et exécute la méthode correspondante. Les réponses sont formatées en JSON via la classe utilitaire `Response`. La connexion à la base de données MySQL est gérée via PDO avec des requêtes préparées pour éliminer les risques d'injection SQL.

4.4.2 Couche Modèles (Dossier `Models/`)

Tous les modèles héritent de la classe abstraite `BaseModel` qui gère l'instanciation unique (Singleton) de la connexion PDO. Chaque modèle contient uniquement des requêtes SQL et retourne des tableaux PHP associatifs.

Méthodes communes :

- `all()` — SELECT * FROM table
- `findById(int $id)` — Récupère un enregistrement par son ID
- `findByName(string $name)` — Recherche par nom
- `create(array $data)` — INSERT INTO
- `update(int $id, array $data)` — UPDATE SET
- `delete(int $id)` — DELETE FROM
- `updateStatus(int $id, string $status)` — Mise à jour du statut

Modèles spécifiques et leurs méthodes uniques :

- **User** : `findByEmail()`, `scheduleDeletion()`, `restoreAccount()`, `getUsersReadyForPermanentDeletion()`, `deletePermanently()`, `updatePassword()`
- **AiTool** : `recommendationCandidates()`, `findActiveByIds()`
- **Review** : `allByUserId()`, `allApprovedByToolId()`, `topApprovedComments()`, `findByUserAndTool()`, `deleteAllByUserId()`, `updateModeration()`
- **Suggestion** : `allByStatus()`, `allByUserId()`, `getAdminHistory()`, `updateStatus()`, `updateFromAiValidation()`
- **FavoriteItem** : `add()`, `remove()`, `isFavorited()`, `allByUserId()`

- **ShelfItem** : `addToShelf()` , `removeFromShelf()` , `exists()` , `toggle()` , `getToolsByShelfId()`
- **SearchHistoryTool** : `linkTool()` , `findToolsByHistoryId()`
- **ToolFeatures** : `linkFeatureToTool()` , `unlinkFeatureFromTool()` , `findFeaturesByToolId()`
- **ToolModel** : `linkModelToTool()` , `unlinkModelFromTool()` , `findModelsByToolId()`

4.4.3 Couche Services (Dossier **Services/**)

Les services contiennent toute la logique métier et servent d'intermédiaire entre les contrôleurs et les modèles. Ils reçoivent des données validées, effectuent des vérifications, orchestrent des appels multiples et retournent des résultats structurés.

Services principaux :

- **AuthService** : `registerUser()` (validation email, hachage BCrypt), `verifyCredentials()` (vérification du statut du compte : actif, suspendu, supprimé avec période de grâce).
- **ExploreService** : `landingSummary()` (statistiques et top outils), `searchByPrompt()` (recommandation IA via OpenRouter), `searchByKeywords()` , `filterTools()` .
- **ToolService** : `listPublicTools()` , `getPublicToolByName()` , `createTool()` , `updateTool()` , `validateToolData()` .
- **ReviewService** : `submitReview()` (vérification doublon + modération IA), `updateReview()` (re-modération), `listUserReviews()` , `deleteReview()` , `banUserAndCleanUp()` .
- **SuggestionService** : `createSuggestion()` (vérification URL doublon + validation IA), `promoteToTool()` (résolution des références mixtes, création d'entités, insertion catalogue).
- **ShelfService** : `createEmptyShelf()` , `toggleShelfItem()` , `getUserShelvesWithStatus()` , `getToolsInShelf()` .
- **FavoriteService** : `toggleFavorite()` , `listUserFavorites()` .
- **SearchHistoryService** : `logSearch()` , `getSearchDetails()` , `clearUserHistory()` .
- **OpenRouterService** : `moderateComment()` , `evaluateToolSubmission()` , `getRecommendations()` . Gère une file d'attente de modèles IA (Claude Opus 4.6 principal, GLM 5.1 secours).
- **ModerationService** : `moderate()` (mots-clés bannis + analyse IA).
- **RecommendationService** : `generateTop6Recommendations()` (recommandation personnalisée).

- **AdminService** : `getDashboardData()` , `getUsers()` , `suspendUser()` , `banUser()` , `deleteUser()` , `rejectSuggestion()` .
- **UploadService** : `handleUpload()` (validation type/taille, stockage fichiers).

4.4.4 Couche Contrôleurs (Dossier **Controllers/**)

Les contrôleurs reçoivent les requêtes HTTP, extraient les paramètres (via `$_GET` , `file_get_contents('php://input')` pour le JSON, `$_FILES` pour les fichiers), vérifient l'authentification via `AuthMiddleware` , et délèguent la logique métier aux services.

Liste complète :

- **AuthController** : `register()` , `login()`
- **UserController** : `profile()` , `stats()` , `updateProfile()` , `professions()` , `deleteAccount()`
- **AdminController** : `dashboard()` , `users()` , `promoteUser()` , `demoteUser()` , `banUser()` , `unbanUser()` , `deleteUser()` , `suggestions()` , `approveSuggestion()` , `rejectSuggestion()` , `moderationReviews()` , `approveModerationReview()` , `deleteModerationReview()` , `tools()` , `updateToolStatus()` , `deleteTool()` , CRUD pour catégories/fournisseurs/features/modèles
- **ToolController** : `index()` , `reviews()` , `topReviews()`
- **ExploreController** : `home()` , `landingSummary()` , `search()` , `recommend()`
- **FavoriteController** : `index()` , `toggle()`
- **ReviewController** : `submit()` , `update()` , `list()` , `delete()`
- **ShelfController** : `list()` , `show()` , `create()` , `update()` , `delete()` , `toggleItem()`
- **SearchHistoryController** : `list()` , `details()`
- **SuggestionController** : `create()` , `history()`

4.4.5 Middleware et Sécurité

- **AuthMiddleware** : Extrait et valide le token JWT depuis l'en-tête `Authorization` . Injecte l' `user_id` dans la requête.
- **RoleMiddleware** : Vérifie le rôle autorisé pour une route spécifique.
- **Hachage BCrypt** : `password_hash()` avec `PASSWORD_BCRYPT` .
- **Protection injections SQL** : PDO avec requêtes préparées (`prepare()` + `bindParam()`).
- **Validation entrées** : `htmlspecialchars()` pour sanitization, validation email, longueur mot de passe (8 caractères min).

5. Conclusion Générale et Perspectives

Le projet **Penguins** a permis de concevoir et de réaliser une plateforme complète de bout en bout, répondant au besoin croissant de centralisation des outils basés sur l'intelligence artificielle.

Les choix architecturaux (MVC en PHP, React pour un Frontend réactif) garantissent la scalabilité de l'application. La séparation rigoureuse des rôles (Visiteur, Utilisateur, Manager, Administrateur) assure une gestion sécurisée et fluide. L'intégration d'un moteur d'IA externe pour les recommandations et la modération automatique apporte une réelle valeur ajoutée à l'expérience utilisateur.

Fonctionnalités clés réalisées :

- Moteur de recherche intelligente avec recommandations par IA en langage naturel
- Catalogue complet d'outils IA avec fiches détaillées, avis et notations
- Système de bibliothèque personnelle (favoris, étagères personnalisées)
- Pipeline de suggestions collaboratives avec validation IA et manuelle
- Interface d'administration complète (utilisateurs, outils, modération, données)
- Modération automatique des avis par IA avec file d'attente de modération manuelle
- Gestion des comptes avec période de grâce de 30 jours pour auto-suppression

Perspectives d'évolution :

- Intégration d'une API d'Intelligence Artificielle en temps réel pour suggérer dynamiquement des outils aux utilisateurs en fonction de leurs prompts.
- Développement d'une application mobile dédiée (React Native ou Flutter).
- Mise en place d'un système de notification push pour prévenir les utilisateurs lors de la validation de leurs suggestions.
- Système de recommandation basé sur le profil utilisateur et l'historique de navigation.
- Module de comparaison d'outils côte à côte.
- Intégration de pipelines CI/CD pour le déploiement automatisé.