

***Yassine BELAAROUS, Clément VIGNON CHAUDEY,
Corentin LACOUME, Ilane RIOTTE***

SAE Web

Dossier de spécifications

Année 2025-2026

Sommaire:

I) Introduction	3
1.1 Contexte	3
1.2 Périmètre du projet	3
1.3 Utilisation de l'intelligence artificielle dans le projet	4
II) Spécifications fonctionnelles	4
2.1 Description générale du système	4
2.2 Exigences fonctionnelles	5
2.3 Règles de gestion	6
2.4 Interfaces et interactions	7
2.6 Contraintes non fonctionnelles	12
III) Spécifications techniques	12
3.1 Architecture technique	12
3.2 Données	13
3.3 Gestion des interfaces	15
3.4 Normes & standards	15
3.5 Environnements	16
IV - Tests	16

I) Introduction

1.1 Contexte

Le laboratoire de paléontologie de l'Université d'Orléans, qui est implanté en Sologne et dans la forêt d'Orléans, mène régulièrement des campagnes de fouilles afin de découvrir et d'analyser des fossiles et traces d'ADN provenant d'espèces anciennes.

Actuellement, la gestion du matériel, des plateformes de fouille, des habilitations du personnel et des échantillons ADN se fait de manière peu centralisée. Cette absence d'outil regroupé ensemble complique la planification des campagnes, le suivi des maintenances et la gestion des budgets.

Le projet « Le parc du Jurassique » vise à concevoir et développer une application web de gestion du matériel, des campagnes de fouilles et des échantillons ADN, afin de faciliter le travail des différents acteurs du laboratoire (direction, chercheurs, techniciens et personnels administratifs).

Ce projet est réalisé dans le cadre du BUT Informatique en 2e année, semestre 3.

1.2 Périmètre du projet

L'application web a pour objectif de centraliser la gestion du matériel, des campagnes de fouilles et des échantillons ADN du laboratoire de paléontologie. Elle permettra d'organiser les campagnes (planification, vérification des habilitations et disponibilités), de suivre l'état et la maintenance des plateformes, de gérer le personnel et d'associer les échantillons collectés à leurs campagnes. Le projet ne traite pas de l'analyse scientifique approfondie des données ADN ni de la gestion comptable complète du laboratoire.

Trois livrables sont attendus :

1. Architecture BD et maquettes (11 octobre)
2. Traitements algorithmiques ADN (15 novembre)
3. Application web complète (29 novembre)

1.3 Utilisation de l'intelligence artificielle dans le projet

Durant ce début de projet, l'intelligence artificielle a été utilisée et nous a permis d'accélérer les différentes étapes que l'on devait réaliser. Nous l'avons utilisé notamment pour :

- Comprendre le sujet et les différents attendus
- Faire les insertions pour tester la base de données
- Reformuler le rapport de groupe

II) Spécifications fonctionnelles

2.1 Description générale du système

Le système développé est une application web scientifique destinée aux chercheurs du parc du Jurassique. Elle permet la gestion complète des campagnes de fouilles et l'exploitation algorithmique des échantillons d'ADN recueillis. L'objectif principal de ce système est d'offrir aux scientifiques un outil unique pour centraliser leurs données, effectuer des analyses bio-informatiques et visualiser les relations de parenté entre espèces fossiles.

L'application combine une interface web intuitive pour la création et la gestion des campagnes de fouilles avec un module algorithmique Python dédié aux traitements des séquences ADN. Les utilisateurs peuvent importer des fichiers au format adn, générer des séquences aléatoires, simuler des mutations (remplacement, insertion, délétion) et calculer différentes mesures de similarité entre échantillons, telles que la distance base-à-base ou la distance de Levenshtein. À partir de ces calculs, le système est capable de reconstruire automatiquement des arbres phylogénétiques représentant les liens de parenté entre les espèces étudiées.

Les principaux acteurs du système sont les scientifiques, les administrateurs et les visiteurs externes. Les scientifiques, acteurs internes, sont les utilisateurs principaux : ils créent et gèrent les campagnes, ajoutent les échantillons, lancent les analyses et consultent les résultats. Les administrateurs assurent la maintenance technique de la plateforme, la gestion des données et la mise à jour des algorithmes. Les visiteurs externes, tels que d'autres chercheurs ou membres du laboratoire, peuvent accéder aux résultats publics partagés par les scientifiques.

Le fonctionnement global du système repose sur une interaction entre l'interface web, la base de données et le module d'analyse ADN. Lorsqu'un scientifique crée une campagne, il y associe des échantillons et leurs séquences ADN. Ces données sont ensuite traitées par le module algorithmique, qui applique les opérations de mutation, de calcul de distance et de reconstruction d'arbre phylogénétique. Les résultats générés sont sauvegardés et restitués à

travers l'interface web, permettant ainsi leur consultation et leur interprétation. Enfin, le système prend en charge la gestion des droits d'accès : les échantillons et résultats peuvent être privés, réservés au scientifique, ou rendus publics pour le reste du laboratoire.

2.2 Exigences fonctionnelles

Liste des fonctions principales du système :

- Gestion des campagnes de fouilles : création, consultation, modification et suppression des campagnes ; association d'échantillons ; suivi des informations clés (nom, date, lieu, responsable).
- Gestion des échantillons d'ADN : importation de fichiers au format .adn, vérification de la validité des séquences (bases A, T, C, G), consultation et mise à jour des données associées.
- Génération et mutation de séquences : génération aléatoire d'ADN d'une longueur donnée, simulation de mutations avec taux configurables (remplacement, insertion, délétion).
- Analyse de similarité entre séquences : calcul de la distance base-à-base pour les remplacements, calcul de la distance de Levenshtein pour les insertions et suppressions.
- Reconstruction d'arbres phylogénétiques : évaluation des distances entre espèces, regroupement des plus proches, identification et visualisation des ancêtres communs.
- Gestion des espèces hypothétiques : création et manipulation d'espèces représentant des ancêtres communs, calcul des distances entre espèces avérées et hypothétiques.
- Gestion des utilisateurs et des droits d'accès : attribution de rôles (scientifique, administrateur, visiteur), définition des permissions de consultation et de

modification, possibilité de rendre des résultats publics.

-Intégration des résultats d'analyse : association des calculs et arbres produits aux campagnes et échantillons concernés, consultation directe des résultats dans l'application web.

2.3 Règles de gestion

Les règles de gestion du système encadrent son fonctionnement afin d'assurer la cohérence des données, la validité des calculs et la sécurité des accès. Le système manipule des séquences d'ADN, des campagnes de fouilles, des échantillons et des résultats d'analyse, chacun soumis à des contraintes métier précises. Les fichiers d'ADN doivent uniquement contenir les bases valides A, T, C et G ; un fichier non conforme est rejeté. Lors de la génération aléatoire, la longueur de la séquence doit être positive et les taux de mutation compris entre 0 et 1. Les mutations par remplacement, insertion ou délétion respectent des règles de probabilité distinctes, garantissant que les séquences résultantes restent valides et cohérentes.

Les calculs de distance entre séquences utilisent des principes biologiques et mathématiques stricts : la distance base-à-base ne s'applique qu'à des séquences de même longueur, alors que la distance de Levenshtein prend en compte les insertions, suppressions et remplacements, représentant le nombre minimal d'opérations qui sont nécessaires pour transformer une séquence en une autre. Ces résultats sont stockés et associés aux échantillons concernés pour assurer la traçabilité des analyses.

Dans la reconstitution des arbres phylogénétiques, une espèce hypothétique ne peut être créée qu'à partir d'au moins deux espèces filles, et la distance entre espèces hypothétiques et avérées est calculée avec la moyenne des distances avec leurs descendantes. Le processus de reconstruction de l'arbre s'effectue de manière itérative, en regroupant les espèces les plus proches jusqu'à l'obtention d'une structure cohérente sans duplication ni boucle.

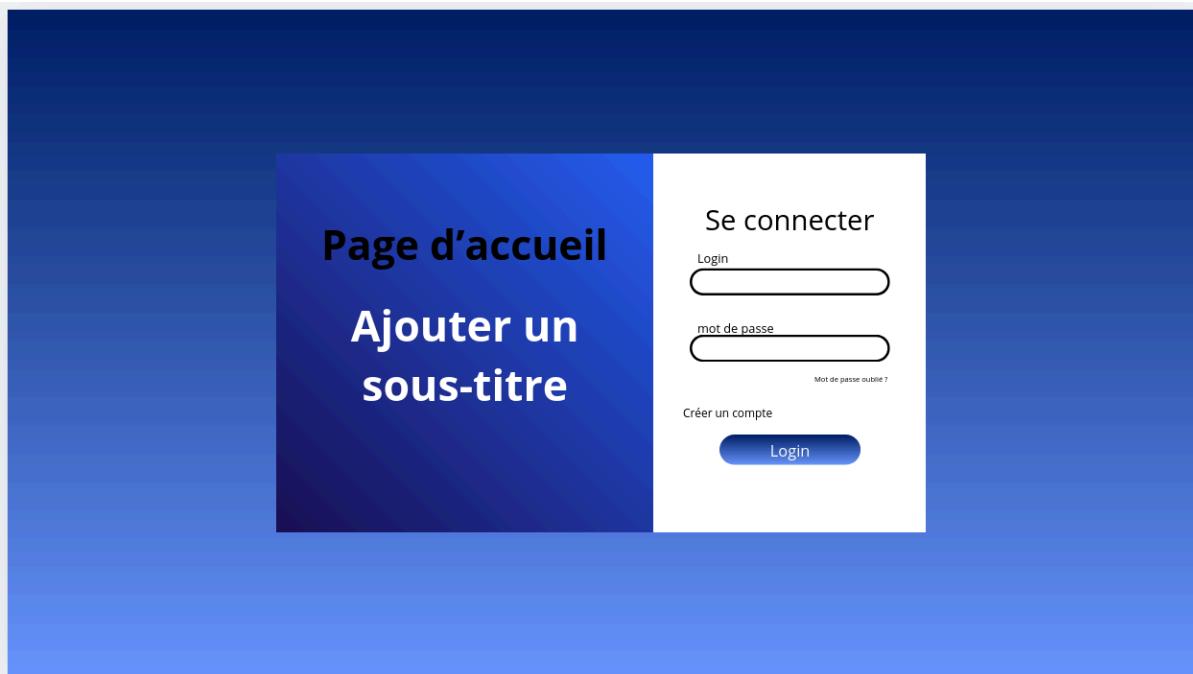
Les droits d'accès et la gestion des utilisateurs suivent également des règles strictes. Seuls les utilisateurs authentifiés peuvent créer, modifier ou supprimer des campagnes, des échantillons ou des résultats. Chaque donnée est associée à son créateur et reste privée par défaut. En revanche, les scientifiques peuvent choisir de rendre certaines données publiques pour les partager avec l'ensemble du laboratoire. Les opérations critiques, telles que la suppression ou la publication, demandent une confirmation et sont limitées aux utilisateurs disposant des droits appropriés.

Enfin, des règles d'intégrité assurent la cohérence globale du système : une campagne doit avoir un nom unique et une date valide ; un échantillon doit être lié à une campagne

existante ; un résultat d'analyse doit référencer des séquences précises et vérifiables. Les calculs doivent être reproductibles et les enregistrements conservés de manière à garantir la fiabilité scientifique et la traçabilité des expérimentations. Ainsi, l'ensemble de ces règles de gestion garantit la rigueur, la sécurité et la crédibilité du système au service des chercheurs.

2.4 Interfaces et interactions

Maquette (Page Connexion):



Maquette (Tableau de bord):

Tableau de bord

Bienvenue, userName, userRole

8
Campagnes Actives

5 200 euros de budget

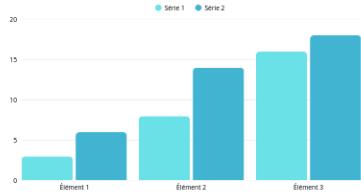
3
Alertes

24
Échantillons

Campagnes en cours

Nom	Date	Durée	Plate-forme	Statut
Lorem ipsum				
Lorem ipsum				

Utilisation Budget



Maquette (gestion du personnel):

Gestion du personnel

Bienvenue, userName, userRole

Recherche par nom :

Filtrer par habilitation



Filtrer

Nom	Habilitations	Disponibilité	Action
-----	---------------	---------------	--------

Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum
-------------	-------------	-------------	-------------

Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum
-------------	-------------	-------------	-------------

Ajouter une personne

Nom / prénom

Habilitation

- Électrique Chimique
 Biologie Radiations

Ajouter

Maquette (Gestion des plateformes):

Gestion des plateformes

Bienvenue, userName, userRole

Recherche par nom :

Filtrer par habilitation

Filtrer

Nom	nbPersonnes	Cout/jour	Habilitations	Maintenance	Action
Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum	>Lorem ipsum

Ajouter / Éditer une plateforme

Nom de la plateforme

Nb Personne Necessaire

Habilitation requise

Électrique Chimique

Biologie Radiations

Cout journalier

Intervalle Maintenance

Ajouter

Maquette (Planification des Campagnes):

Planification des Campagnes

Bienvenue, userName, userRole

Nom	Durée	Plateforme	Lieu	Participants	Budget	Status	Action
Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum				

Ajouter / Éditer une plateforme

Nom de la plateforme

Nb Personne Necessaire

Habilitation requise

Électrique Chimique

Biologie Radiations

Cout journalier

Intervalle Maintenance

Ajouter

Maquette (Gestion des maintenance):

Gestion des maintenances

Bienvenue, userName, userRole

Nom	Intervalle	Jours utilisés	Status	Actions
Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum
Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum	Lorem ipsum

Planifier une plateforme

Nom de la plateforme	Date
Type de maintenance	Notes

Ajouter

Date	Plateforme	Type	Technicien	Status
Lorem ipsum				
Lorem ipsum				

Maquette (Gestion du budget):

Gestion du Budget

Bienvenue, userName, userRole

10 000 euros
Budget Mensuel

4800 euros
Utilisé ce mois

5200 euros
Restant

48%
Taux d'utilisation

Configurer Budget Mensuel

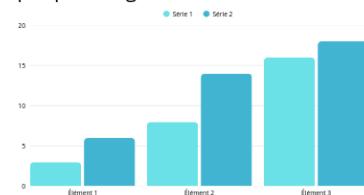
Septembre 2025

Budget mensuel (euros)

Montant

Ajouter

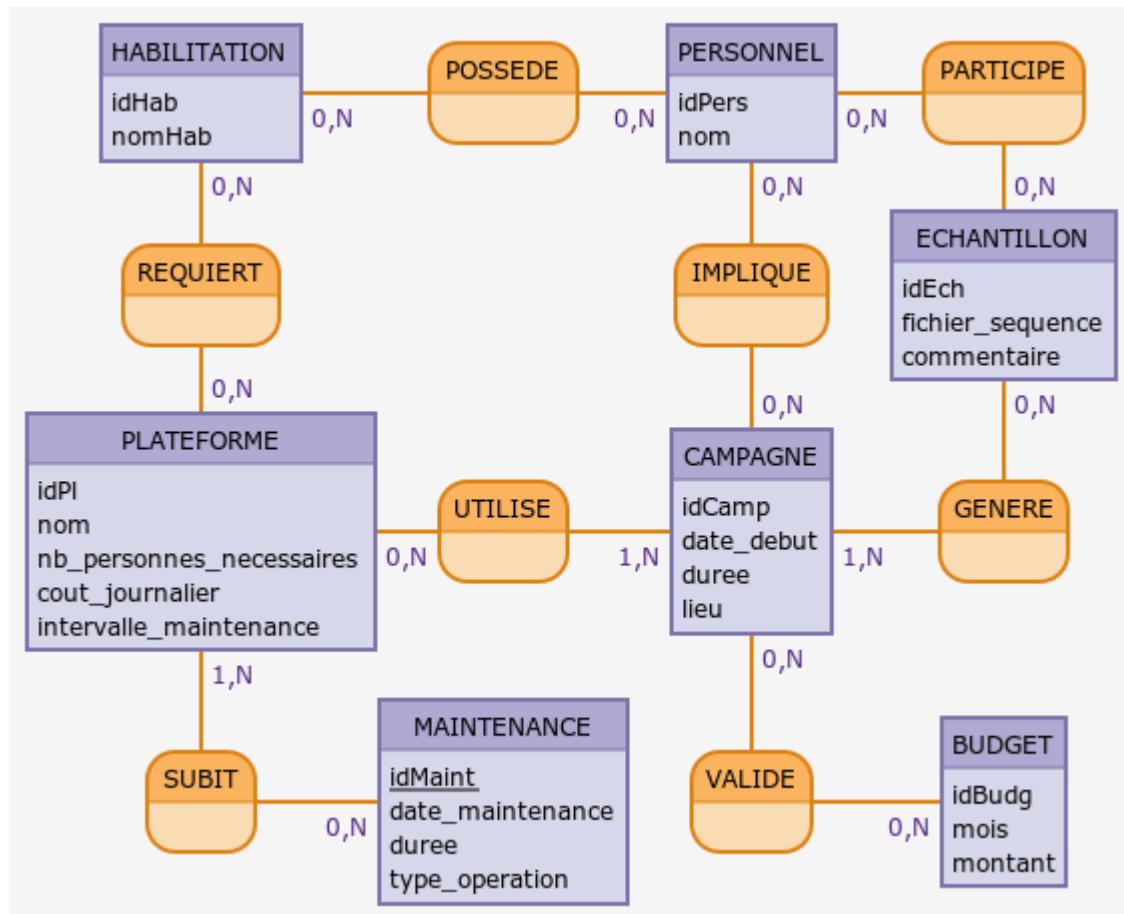
Graphique Budget



Campagne	Date	Durée	Coût Estimé	Status Budget
Lorem ipsum				
Lorem ipsum				

2.5 Données

MCD:



Dépendances fonctionnelles :

- PLATEFORME `idPl` → `nom`, `nb_personnes_necessaires`, `cout_journalier`, `intervalle_maintenance`
- PERSONNEL `idPers` → `nom`
- BUDGET `idBudg` → `mois`, `montant`
- HABILITATION `idHab` → `nomHab`
- CAMPAGNE `idCamp` → `date_debut`, `duree`, `lieu`, `idPl`
- ECHANTILLON `idEch` → `fichier_sequence`, `commentaire`, `idCamp`
- MAINTENANCE `idMaint` → `date_maintenance`, `duree`, `type_operation`, `idPl`

2.6 Contraintes non fonctionnelles

Performance : L'application doit être réactive, notamment lors du filtrage dynamique du personnel pendant la planification d'une campagne.

Sécurité : L'accès aux différentes fonctionnalités doit être contrôlé par un système de rôles (Chercheur, Technicien, etc.).

Disponibilité : L'application doit être accessible en permanence durant les heures de travail du laboratoire.

Compatibilité : L'interface doit être compatible avec les principaux navigateurs web (Chrome, Firefox, Safari, Edge).

III) Spécifications techniques

3.1 Architecture technique

Le projet s'appuie sur une architecture web classique. La base de données est gérée par le SGBD MySQL. Ce choix est motivé par sa fiabilité, sa performance et sa popularité dans le développement d'applications web.

Base de données : MySQL a été choisi pour stocker toutes les données relatives aux campagnes de fouilles, au personnel, aux plateformes et aux échantillons.

Technologies envisagées pour la suite :

- Langage back-end : Python sera utilisé pour la partie algorithmique.
- Framework web : Flask sera utilisé pour connecter la logique applicative à la base de données et générer les pages web.
- Front-end : L'interface utilisateur sera développée en HTML, CSS pour assurer une expérience interactive et ergonomique.

3.2 Données

MPD :

PLATEFORME(idPl, nom, nb_personnes_necessaires, cout_journalier, intervalle_maintenance)

PERSONNEL(idPers, nom)

BUDGET(idBudg, mois, montant)

HABILITATION(idHab, nomHab)

CAMPAGNE(idCamp, date_debut, duree, lieu, #idPl)

ECHANTILLON(idEch, fichier_sequence, commentaire, #idCamp)

MAINTENANCE(idMaint, date_maintenance, duree, type_operation, #idPl)

IMPLIQUE(#idCamp, #idPers)

PARTICIPE(#idEch, #idPers)

VALIDE(#idBudg, #idCamp)

POSSEDE(#idPers, #idHab)

REQUIERT(#idPl, #idHab)

Légende : mot souligné = clé primaire, un # devant un mot = clé étrangère

Triggers :

Pour garantir la cohérence et l'intégrité des données, plusieurs triggers ont été implémentés dans la base de données. Ils se déclenchent automatiquement pour effectuer des vérifications avant ou après l'insertion, la modification ou la suppression de données.

Voici une description des triggers créés :

1. Validation de la durée des campagnes (validate_campagne_duree_insert / _update)

L'objectif est de s'assurer que la durée d'une campagne est réaliste.

Fonctionnement : Ce trigger se déclenche avant toute insertion ou modification dans la table CAMPAGNE et vérifie que la durée est comprise entre 1 et 365 jours. Si la condition n'est pas respectée, l'opération est annulée.

2. Contrôle du nombre de personnes par campagne (check_nb_personnes_insert / _update / _delete)

l'objectif est de vérifier que le nombre de personnes affectées à une campagne ne dépasse pas la capacité de la plateforme utilisée.

Fonctionnement : Après chaque ajout, modification ou suppression dans la table IMPLIQUE, ce trigger compare le nombre de personnes impliquées dans la campagne avec le nombre requis par la plateforme. Si le nombre est dépassé, une erreur est générée.

3. Contrôle des habilitations du personnel (check_habilitations_insert)

l'objectif est de garantir que le personnel affecté à une campagne possède toutes les habilitations nécessaires pour la plateforme utilisée.

Fonctionnement : Avant d'associer une personne à une campagne (table IMPLIQUE), ce trigger vérifie que celle-ci possède bien toutes les habilitations requises par la plateforme.

4. Contrôle du budget (check_budget_valide)

l'objectif est de s'assurer qu'une campagne ne peut être validée si le budget alloué est insuffisant.

Fonctionnement : Avant d'insérer une validation dans la table VALIDE, le trigger calcule le coût total de la campagne (coût journalier × durée) et le compare au montant du budget. Si le budget est inférieur au coût, l'opération est bloquée.

5. Limitation du nombre d'échantillons (limit_echantillons_insert)

l'objectif est de limiter à 10 le nombre d'échantillons pouvant être associés à une seule campagne.

Fonctionnement : Avant l'ajout d'un nouvel échantillon, ce trigger compte le nombre d'échantillons déjà liés à la campagne. S'il y en a déjà 10 ou plus, l'insertion est refusée.

6. Mise à jour du budget après validation (maj_budget_apres_validation_insert / _update)

l'objectif est de décrémenter automatiquement le montant d'un budget lorsqu'une campagne est validée.

Fonctionnement : Après la validation d'une campagne (insertion ou modification dans VALIDÉ), le trigger met à jour le montant du budget en lui soustrayant le coût total de la campagne.

3.3 Gestion des interfaces

L'application web utilise le protocole HTTP pour les échanges entre client et serveur.

La sécurité des échanges repose sur la gestion des sessions Flask, la clé secrète, et la protection CSRF pour les formulaires.

3.4 Normes & standards

Pour garantir la qualité, la lisibilité et la maintenabilité du code tout au long du projet, l'équipe s'est accordée sur les conventions suivantes :

Qualité du code SQL :

- Nous avons fait en sorte que notre code SQL soit de qualité, c'est-à-dire qu'il y a des indentations et les noms de table sont en majuscules.

Documentation du code :

- Les fichiers SQL qui sont durs à comprendre ont été commentés, comme celui sur les triggers.

Suivi de projet :

- Des documents de suivi, comme les rapports individuels et le dossier de spécifications, sont rédigés et mis à jour à chaque rendu du projet.

3.5 Environnements

3.6 Outils de versionning et conventions

Durant ce projet nous avons utilisé divers outils mais le plus important reste Github.
Nous n'avons pas encore créé de branche car dans le projet, il n'était pas demandé de le faire mais par la suite, nous comptons créer des branches pour séparer les parties du projet.

IV - Tests

Jeux de données de test

arbrePhylogenetiquesTest.py :

Ce fichier contient des tests sur la création d'espèces, la construction d'espèces hypothétiques, le calcul de distances entre espèces (simples et hypothétiques), et la reconstruction d'arbres phylogénétiques. Les jeux de données couvrent des cas simples, des cas limites et des scénarios d'assemblage d'arbres.

calculSimilariteTest.py :

Ce fichier utilise pytest pour valider les fonctions de calcul de similarité (distance de remplacement et distance de Levenshtein) sur des séquences ADN variées, y compris des cas de séquences identiques, différentes, de tailles différentes ou vides.

test_performance.py :

Ce fichier mesure les temps d'exécution des fonctions calculer_distance et reconstruire_arbre sur de grandes séquences ou de nombreux objets, afin de garantir la performance de l'application.

Plan de tests

Tests unitaires

Les fonctions de base (calcul de distance, mutation, etc) sont testées individuellement dans calculSimilariteTest.py et arbrePhylogenetiquesTest.py.

Tests d'intégration

Des scénarios complets sont simulés dans arbrePhylogenetiquesTest.py : création d'espèces, calculs de distances, construction d'arbres, vérification de la structure finale.

Tests de performance

test_performance.py évalue le temps de calcul pour des séquences longues ou de nombreux objets, afin de s'assurer que l'application reste réactive.

Couverture des tests

L'ensemble des modules critiques du projet est couvert par ces tests, ce qui garantit la fiabilité, la robustesse et la maintenabilité du code.

Tests sur l'application

Le fichier test_app.py contient 3 classes, chacunes testant un aspect spécifique de l'application

TestAuth : S'assure que l'authentification des utilisateurs fonctionne correctement.

TestRoles : Vérifie que les permissions basées sur les rôles sont bien appliquées, en contrôlant par exemple que seul un administrateur peut accéder à la page du personnel.

TestPersonnel : Teste les fonctionnalités de gestion du personnel, comme l'ajout et la suppression d'un membre.

Malheureusement, nous n'avons pas réussi à bien faire fonctionner ces tests ce qui fait qu'il y a des erreurs et que le coverage n'est pas terrible (aux alentours des 50%).

Name	Stmts	Miss	Cover	Missing
/home/jut45/Etudiants/o22401350/SAE/SAE-Web/config.py	5	0	100%	
/home/jut45/Etudiants/o22401350/SAE/SAE-Web/source/AlgoADN.py	31	26	16%	14-18, 33-42, 60-78
/home/jut45/Etudiants/o22401350/SAE/SAE-Web/source/arbresPhylogenetiques.py	57	49	14%	3-4, 8-9, 13, 18-34, 39-63, 67-77
/home/jut45/Etudiants/o22401350/SAE/SAE-Web/source/calculSimilarite.py	23	21	9%	14-21, 36-60
__init__.py	2	0	100%	
app.py	17	2	88%	19-29
commands.py	114	93	18%	16-144, 150-151, 159-168, 176-185
conftest.py	20	1	95%	36
forms.py	58	3	95%	22-24
models.py	116	28	76%	28, 108, 132, 148, 152-156, 166, 185, 189-196, 206-205, 218, 235, 239-242
test_views.py	0	0	100%	
tests/conftest.py	56	0	100%	
tests/test_app.py	57	9	84%	15-24
views.py	498	314	37%	37, 60-86, 155, 159-161, 188, 195-209, 224-225, 232-255, 262-291, 300-307, 8-345, 352-358, 370-386, 393-532, 542-561, 571-610, 617-623, 635-664, 675-693, 700-717, 734-740, 750-751, 758-768, 775-793, 800-808, 816-823, 833-863
TOTAL	1054	546	48%	I

