

Rapport intermédiaire - 20 Novembre 2025

Développement d'un site Web pour la tenue d'un cahier de
laboratoire

Bardet Trouilloud Juliette, Helias Ewan, Marhabi Zahir, Rey Axel

Introduction	3
Structure de la Base de données	4
Table compte	4
Table projet	5
Table experience	5
Table experience_experimentateur	5
Table projet_collaborateur_gestionnaire	6
Table projet_experience	6
Table salle_materiel	6
Table salle_experience	7
Table notification_experience	7
Table notification_projet	7
Tables des fonctions	9
Diagrammes UML	15
Conclusion	17

Introduction

Le projet site web est inclus dans les UE Génie Logiciel et UML et Bases de données avancées de GB5 options BIMB. Pour ce projet, on se propose de réaliser un site web pour une entreprise permettant de tenir un cahier de laboratoire des projets réalisés par différents acteurs : chercheurs, étudiants, expérimentateur. Pour ce faire, des étapes de conception avec différents rendus ont été prévu tels que défini dans le tableau suivant :

Etapes du Projet	Date de rendu
Analyse des besoins du client et définition des spécifications fonctionnelles.	17 Octobre
Conception : spécifications détaillées avec l'ensemble des fonctions utiles, leurs entrées/sorties, diagramme de classe.	20 Novembre
Rapport final comprenant les spécifications mises à jour en fonction du produit, critères utilisés pour le choix des tests, résultats des tests, démarche adoptée pour l'élaboration du site web, difficultés rencontrées, améliorations possibles.	11 Décembre
Soutenance	18 Décembre

Structure de la Base de données

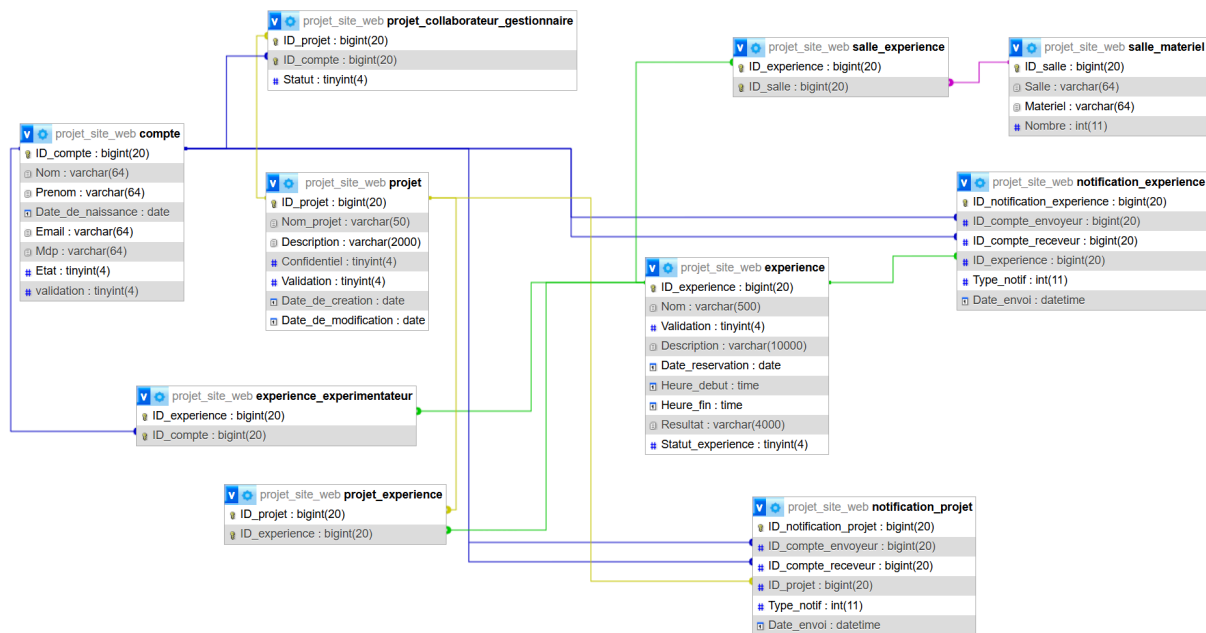


Table compte

Cette table contient toutes les informations de tous les utilisateurs. La clé primaire de cette table est **ID_compte (BIGINT)**, un entier unique qui identifie chaque utilisateur.

- **Nom (VARCHAR(64))** : une chaîne de caractères qui représente le nom de l'utilisateur.
- **Prenom (VARCHAR(64))** : une chaîne de caractères qui représente le prénom de l'utilisateur.
- **Date_de_naissance (DATE)** : une date qui représente la date de naissance de l'utilisateur.
- **Etat (TINYINT)** : un entier qui définit l'état de l'utilisateur (1=étudiant, 2=professeur/chercheur, 3=ADMIN).
- **Validation (TINYINT)** : qui définit si le compte est validé ou non (1=Validé, 0=Non validé).
- **Email (VARCHAR(64))** : une chaîne de caractères qui représente le mail de l'utilisateur.
- **Mdp (VARCHAR(64))** : une chaîne de caractères qui représente le mot de passe de l'utilisateur.

Table projet

Cette table contient l'ensemble des informations relatives aux projets. La clé primaire de cette table est **ID_projet (BIGINT)**, un entier unique qui identifie chaque projet.

- **Nom_projet (VARCHAR(50))** : nom du projet.
- **Description (VARCHAR(2000))** : texte décrivant le contenu et les objectifs du projet.
- **Confidentiel (TINYINT)** : indicateur booléen (0 ou 1) précisant si le projet est confidentiel ou non.
- **Validation (TINYINT)** : indicateur de validation du projet par le gestionnaire (0 = non validé, 1 = validé).
- **Date_de_creation (DATE)** : date de création du projet dans le système.
- **Date_de_modification (DATE)** : la dernière date de modification du projet dans le système.

Table experience

Cette table contient toutes les informations relatives aux expériences associées aux projets. La clé primaire de cette table est **ID_experience (BIGINT)**, un identifiant unique pour chaque expérience.

- **Nom (VARCHAR(500))** : indique le nom donné à l'expérience.
- **Validation (TINYINT, DEFAULT 0)** : indique si la création de l'expérience est validée par un gestionnaire (0 = non validée, 1 = validée).
- **Description (VARCHAR(10000))** : description de l'expérience.
- **Date_reservation (DATE)** : date prévue pour la réservation de la salle ou la réalisation de l'expérience.
- **Heure_debut (TIME)** : heure de début de l'expérience.
- **Heure_fin (TIME)** : heure de fin de l'expérience.
- **Resultat (VARCHAR(4000))** : champ textuel contenant le résultat ou le compte rendu de l'expérience.
- **Statut_experience (TINYINT)** : indique si l'expérience est terminée (2) en cours (1) ou non commencé (0).

Table experience_experimentateur

Cette table fait le lien entre les expériences et les utilisateurs (comptes) qui y participent. Elle représente une relation n-n entre la table expérience et la table compte.

- **ID_experience (BIGINT)** : identifiant de l'expérience, clé étrangère vers la table expérience.

- **ID_compte (BIGINT)** : identifiant du compte de l'expérimentateur, clé étrangère vers la table compte.
- **Clé primaire composée** : (ID_experience, ID_compte), garantissant l'unicité du lien entre un utilisateur et une expérience.

Cette table permet donc de connaître quels utilisateurs ont participé à quelles expériences.

Table projet_collaborateur_gestionnaire

Cette table établit la relation entre les projets et les utilisateurs associés à ces projets. Elle correspond également à une relation n-n entre la table Projet et la table Compte.

- **ID_projet (BIGINT)** : identifiant du projet, clé étrangère vers la table projet.
- **ID_compte (BIGINT)** : identifiant du compte utilisateur, clé étrangère vers la table compte.
- **Statut (TINYINT)** : indique le rôle de l'utilisateur dans le projet (0 = collaborateur, 1 = gestionnaire).
- **Clé primaire composée** : (ID_projet, ID_compte).

Cette table permet donc d'associer plusieurs collaborateurs ou gestionnaires à un même projet.

Table projet_experience

Cette table fait le lien entre les projets et les expériences qui leurs sont associées. Elle représente une autre relation n-n entre la table projet et la table experience.

- **ID_projet (BIGINT)** : identifiant du projet, clé étrangère vers la table projet.
- **ID_experience (BIGINT)** : identifiant de l'expérience, clé étrangère vers la table experience.
- **Clé primaire composée** : (ID_projet, ID_experience).

Elle permet ainsi de relier plusieurs expériences à un même projet. Elle représente une autre relation n-n entre la table Projet et la Table Expérience.

Table salle_materiel

Cette table contient les informations relatives aux salles disponibles pour la réalisation d'expériences. La clé primaire de cette table est **ID_salle (BIGINT)**, un entier unique identifiant chaque salle.

- **Salle (VARCHAR(20))** : nom de la salle.

- **Materiel (VARCHAR(50))** : description du matériel disponible dans cette salle (ex : microscope, centrifugeuse, etc.).
- **Nombre (INT(11))** : Nombre d'unité du matériel présent dans la salle

Table salle_experience

Cette table établit la relation entre les salles et les expériences. Elle correspond à une relation n-n entre la table salle et la table experience.

- **ID_experience (BIGINT)** : identifiant de l'expérience, clé étrangère vers la table experience.
- **ID_salle (BIGINT)** : identifiant de la salle, clé étrangère vers la table salle_materiel.
- **Clé primaire composée** : (ID_experience, ID_salle).

Elle permet de gérer les expériences qui peuvent avoir lieu dans plusieurs salles, ou les salles partagées entre plusieurs expériences.

Table notification_experience

La table notification_experience regroupe toutes les notifications liées aux expériences. Elle permet de tracer et transmettre aux utilisateurs les demandes, validations ou informations concernant une expérience. La clé primaire est **ID_notification_experience (BIGINT)**.

- **ID_compte_envoyeur (BIGINT)** : identifiant de l'utilisateur qui envoie la notification. Il s'agit d'une clé étrangère référencée vers la table compte.
- **ID_compte_receveur (BIGINT)** : identifiant de l'utilisateur qui reçoit la notification. Il s'agit également d'une clé étrangère vers la table compte.
- **ID_experience (BIGINT)** : identifiant de l'expérience concernée par la notification. Cette valeur est une clé étrangère liée à la table experience.
- **Type_notif (INT)** : type ou catégorie de notification (ex : demande de participation, validation, modification, etc.).
- **Date_envoi (DATETIME)** : date et heure d'envoi de la notification, automatiquement définie à la création (DEFAULT CURRENT_TIMESTAMP).

Grâce à ces champs, cette table permet de gérer toutes les interactions concernant les expériences entre les différents utilisateurs.

Table notification_projet

La table notification_projet contient les notifications relatives aux projets. Elle permet d'informer les utilisateurs des mises à jour, demandes de collaboration, validations ou autres actions portant sur un projet. La clé primaire est **ID_notification_projet (BIGINT)**.

- **ID_compte_envoyeur (BIGINT)** : identifiant de l'utilisateur ayant envoyé la notification, clé étrangère vers la table compte.
- **ID_compte_receveur (BIGINT)** : identifiant du destinataire de la notification, également clé étrangère vers la table compte.
- **ID_projet (BIGINT)** : identifiant du projet concerné par la notification, clé étrangère vers la table projet.
- **Type_notif (INT)** : type ou nature de la notification (ex : ajout au projet, validation, modification, etc.).
- **Date_envoi (DATETIME)** : horodatage de l'envoi de la notification, défini automatiquement (DEFAULT CURRENT_TIMESTAMP).

Cette table permet de gérer la communication et les interactions autour des projets entre les différents utilisateurs de la plateforme.

Tables des fonctions

Nom	Input	Output	Fonctionnalité
Fonctions générales aux différentes pages			
connectBDD()	Aucun	Objet PDO ou message d'erreur	Procédure qui établit la connexion à la base MySQL via PDO. Retourne l'objet de connexion si réussite, sinon un message d'erreur.
afficher_bandeau_haut()	\$bdd, \$userID	Header affiché	Procédure qui affiche le bandeau supérieur commun aux pages, ainsi que les notifications de l'utilisateur concerné.
afficher_bandeau_bas()	Aucun	Footer affiché	Procédure qui affiche le pied de page sur les pages
get_last_notif()	\$bdd, \$IDuser, \$limit=4	Tableau des notifs	Affiche les dernières notifications à l'utilisateur concerné sous forme d'un array.
notification()	\$bdd, \$ID_compte_envoyeur, \$ID_compte_receveur, \$type_notif, \$objet		Procédure qui insère une nouvelle notification dans la table de la bdd pour un utilisateur donné et un objet donné.
verifier_taille_champs()	\$texte	Booléen	fonction qui vérifie que la longueur du texte ne dépasse pas celle définie
gerer_notification()	\$id_notifcation		procédure permettant de mettre à jour l'état d'une notification (valider ou supprimer)
verification_connexion()	\$id_user	lance layout_erreur()	Vérifie si la personne essayant d'accéder à une page est bien connectée, auquel cas l'accès à la page est bloqué.
verification_accès()	\$id_user, \$id_objet, \$type=0	Booléen (True si il a accès)	Fonction qui vérifie que l'utilisateur a les droits d'accès à l'expérience(\$type=1) ou le projet(\$type=0) (collaborateurs, gestionnaires, expérimentateur). En fonction du résultat, layout_erreur() est lancé
layout_erreur()	Aucun	Aucun	Affiche une mise-en-page spécifique avec un message

			d'erreur et lance une fonction exit() pour empêcher tout autre affichage sur la page
compte_est_admin()	\$bdd, \$email	Booléen (True si admin)	Fonction qui vérifie si l'utilisateur qui se connecte est un admin (état =3)
verifier_confidentialite()	\$ID_utilisateur, \$ID_projet	Booléen (True si il a accès)	Fonction qui vérifie les droits d'accès d'un utilisateur à un projet (gestionnaire, collaborateur, expérimentateur).
shortage()	\$texte, \$modalite	Texte tronqué	Fonction qui renvoie un texte tronqué suivie de "..." si jamais il est trop long.
get_projets()	\$bdd, \$user=0	Listes les différents projets	Fonction qui récupère les projets stockés dans la bdd et les affiche sur plusieurs pages. Si l'user =1 alors affiche les projets de l'utilisateur.
filtrer_pro_exp()	\$projet_exp=0		Cela prend en compte si l'utilisateur ne veut chercher que des projets(\$projet_exp=0), des expériences(\$projet_exp=1) ou les deux(\$projet_exp=2). En fonction du choix, il appelle get_Projets() et/ou get_Experiences()
filtrer_projets()	\$liste_projets, \$texte=None, \$confidentialité = None, \$statut = None	Liste filtrée des projets	Fonction qui filtre projets selon des critères (confidentialité, statut, texte).
trier_projets()	\$liste_projets, \$ordre=0, \$modalité=1	Liste triée des projets	Permet de trier les résultats par ordre alphabétique (\$modalité=0), par date de création(\$modalité=1) ou par date de modification(\$modalité=2) que ce soit croissant (\$ordre=0) ou décroissant (\$ordre=1)
get_experiences()	\$bdd, \$user=0	Listes les différentes expériences	Fonction qui récupère les expériences stockées dans la bdd et les affiche sur plusieurs pages. Si l'user =1 alors affiche les expériences de l'utilisateur.
filtrer_experiences()	\$liste_experiences, \$texte=None, \$, \$statut = None	Liste filtrée des expériences	Fonction qui filtre les expériences selon des critères tel que du texte qui prend en compte les

			expérimentateur ou le titre ou le statut qui permet de savoir si l'expérience est fini
create_page()	\$item, \$items_per_page	Nombre de page	Fonction qui récupère un ensemble d'item (projet ou expérience) et qui retourne le nombre de page/onglet que l'on doit créer
afficher_projets_pages() ()	\$projets, \$page_actuelle \$items_per_page		Procédure qui permet d'afficher les projets, en créant le nombre de pages nécessaire.
afficher_experience_pages() ()	\$experiences, \$page_actuelle, \$items_per_page		Procédure qui permet d'afficher les expériences, en créant le nombre de pages nécessaire.
trier_experiences()	\$liste_projets, \$ordre=0, \$ date_réservation=Non e	Liste triée des projets	Permet de trier les résultats par ordre croissant (\$ordre=0), par date de réservation de la salle d'expérience
gerer_experience()	\$Id_experience, \$description=NULL, \$matériel=NULL, \$salle=NULL,\$date=N ULL,\$heure_deb=NUL L,\$heure_fin=NULL		Procédure qui permet de modifier les informations relatives à une expérience selon les paramètres qui ne sont pas NULL
gerer_projet()	\$Id_projet,\$description =NULL, \$statut=NULL, \$gestionnaire=NULL,\$ collaborateur=NULL		Procédure qui permet de modifier les informations relatives à un projet selon les paramètres qui ne sont pas NULL
supprimer_experience()	\$Id_experience		Procédure qui permet de supprimer une expérience
supprimer_projet()	\$Id_projet		Procédure qui permet de supprimer un projet
Page d'inscription			
verifier_mdp()	\$mdp	Tableau d'erreurs ou vide	Fonction qui vérifie la complexité du mot de passe : longueur, majuscule, minuscule, chiffre, caractère spécial
email_existe()	\$bdd, \$email	Booléen (True si l'email existe)	Fonction qui vérifie si l'adresse email est déjà utilisée dans la base
mot_de_passe_identiqu	\$mdp1, \$mdp2	Booléen	fonction qui compare le mot de

e()		(true si identiques)	passee et sa confirmation
verifier_email_axoulab()	\$email	Booléen (true si format valide)	fonction qui vérifie si l'email respecte le format <code>prenom.nom@axoulab.fr</code>
inserer_utilisateur()	\$nom, \$prenom, \$date_de_naissance, \$email, \$mdp_hash	Booléen (true si insertion réussie)	procédure qui insère un nouvel utilisateur dans la table compte
Page de connexion			
mot_de_passe_correct()	\$bdd, \$email, \$mdp	Booléen (true si mot de passe correspond)	Fonction qui vérifie si le mot de passe fourni correspond à celui enregistré dans la bdd
connexion_valide()	\$bdd, \$email, \$mdo	Booléen (true si connexion possible)	Fonction qui combine email_existe() et mot_de_passe_correct() pour valider la connexion d'un utilisateur
recuperer_id_compte()	\$bdd, \$email	ID du compte (int) ou null	Fonction qui permet de récupérer l'identifiant unique associé à une adresse email donnée
en_cours_validation()	\$bdd, \$email	Booléen (True si validée)	Fonction qui vérifie si le compte a été validé par l'admin
Page mot de passe oublié			
email_existe()	\$bdd, \$email	Booléen (true si existe)	Fonction qui vérifie si l'adresse mail existe dans la bdd. Si c'est le cas, alors un mail de réinitialisation du mot de passe est envoyé
Page explorer			
Les fonctions de cette page se trouvent dans les fonctions communes, et seront utilisées pour la page "Mes projets"			
Page projet			
get_info_projet()	\$ID_projet	Données du projet	Fonction qui récupère les informations du projet depuis la bdd et les affiche si l'utilisateur a

			accès
barre_progression()	\$ID_projet	Affichage barre de progression	Fonction qui calcule et montre l'avancement du projet selon le nombre d'expériences terminées
afficher_projet()	\$info_projet	Affichage des infos du projet	Fonction qui affiche les infos complètes d'un projet.
Page mes projets			
Les fonctions de cette page se trouvent dans les fonctions communes, et seront utilisées pour la page "Explorer"			
Page création de projet			
creer_projet()	\$nom_projet, \$description, \$confidentialité		Procédure qui insère un projet dans la BDD avec nom du projet, une description, sa confidentialité, les gestionnaires et les collaborateurs
ajouter_participant()	\$bdd,\$id_projet, \$gestionnaires, \$collaborateur		Procédure qui permet de relier un projet à des utilisateurs et quels sont leur rôles sur le projet
Page experience			
get_infos_experience()	\$Id_experience	Données de l'expérience	Fonction qui récupère les informations de l'expérience depuis la BDD si l'utilisateur a accès
afficher_experience()	\$info_experience	Affichage des infos de l'expérience	Fonction qui affiche les infos de l'expérience.
Page création d'expérience			
creer_experience()	\$nom_experience, \$description, \$matériel, \$salle,\$date,\$heure_d eb,\$heure_fin		Procédure qui crée une nouvelle expérience dans la BDD avec nom de l'expérience, la description, le matériel et la salle et vérifie la disponibilité via verifier_Disponibilite()
verifier_disponibilite()	\$salle, \$matériel, \$date	Booléen	Fonction qui vérifie la disponibilité pour la création d'une expérience en fonctions des salles et matériels disponibles à la date demandée
Page mes expériences			

Les fonctions de cette page se trouvent dans les fonctions communes, et seront utilisées pour la page "Explorer"

Pages admin

get_utilisateurs()	\$bdd	Liste d'utilisateur	Fonction qui récupère tous les utilisateurs et affiche leur informations
get_materiel_salle()	\$bdd	Liste du matériel ainsi que des salles	Fonction qui récupère la liste de l'ensemble du matériel et affiche leur informations
modifier_utilisateur()	\$user		Procédure qui permet de changer les informations de l'utilisateur ou de supprimer l'utilisateur
gerer_materiel_salle()	\$materiel, \$salle, \$suppression		Procédure qui ajoute ou supprime du matériel ou une salle

Page mon profil

modifier_mdp()	\$mdp, \$user_ID		Procédure qui met à jour le mot de passe de l'utilisateur
modifier_photo()	\$photo, \$user_ID		Procédure qui met à jour la photo de l'utilisateur

Table des notifications :

Type de notification	Texte de notification	Destinataire	Possibilité
1	{Nom_envoyeur} {Prenom_envoyeur} vous a proposé de créer l'expérience {Nom_experience}	Gestionnaire(s)	Valider, Rejeter, Demande à être modifiée
2	{Nom_envoyeur} {Prenom_envoyeur} a validé l'expérience {Nom_experience}	Utilisateur connecté	Valider
3	{Nom_envoyeur} {Prenom_envoyeur} a refusé l'expérience {Nom_experience}	Utilisateur connecté	Valider
4	{Nom_envoyeur} {Prenom_envoyeur} vous a invité à modifier l'expérience {Nom_experience}	Utilisateur connecté	Valider

5	{Nom_experience} a été modifiée par {Nom_envoyeur} {Prenom_envoyeur}	Expérimentateur	Valider
11	{Nom_envoyeur} {Prenom_envoyeur} vous a proposé de créer le projet {Nom_projet}	Chercheur	Valider, Rejeter, Demande à être modifiée
12	{Nom_envoyeur} {Prenom_envoyeur} a validé le projet {Nom_projet}	Etudiant	Valider
13	{Nom_envoyeur} {Prenom_envoyeur} a refusé le projet {Nom_projet}	Etudiant	Valider
14	{Nom_envoyeur} {Prenom_envoyeur} vous a invité à modifier le projet {Nom_projet}	Etudiant	Valider
15	{Nom_projet} a été modifiée par {Nom_envoyeur} {Prenom_envoyeur}	Etudiant	Valider

Diagrammes UML

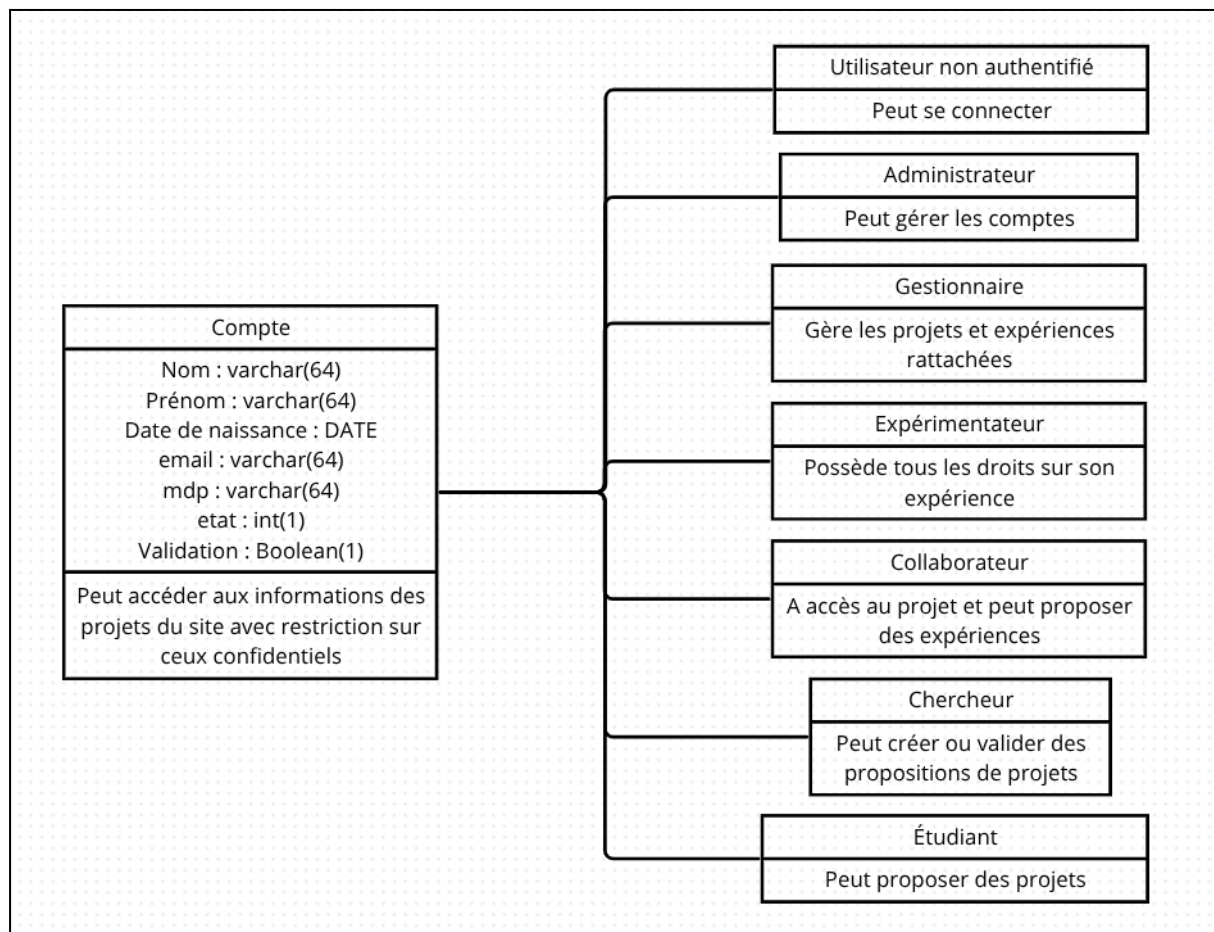


Diagramme de classe des différents types de comptes

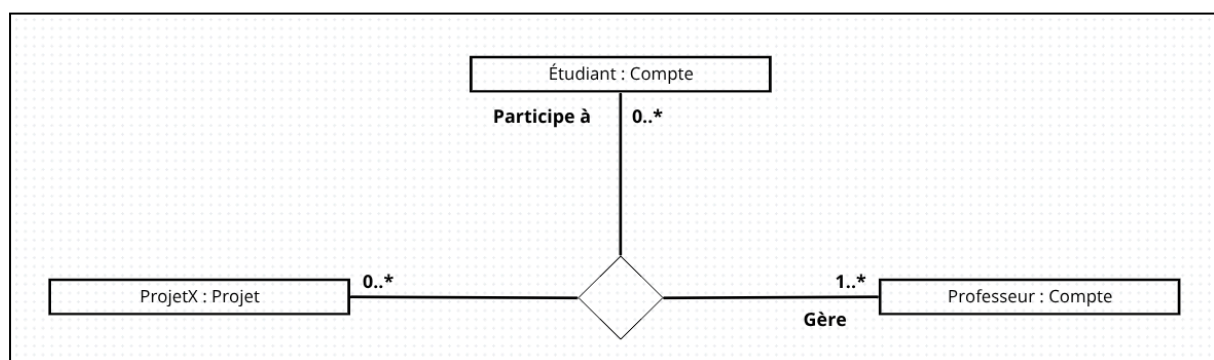


Diagramme d'objets pour la gestion des projets entre étudiants et professeurs

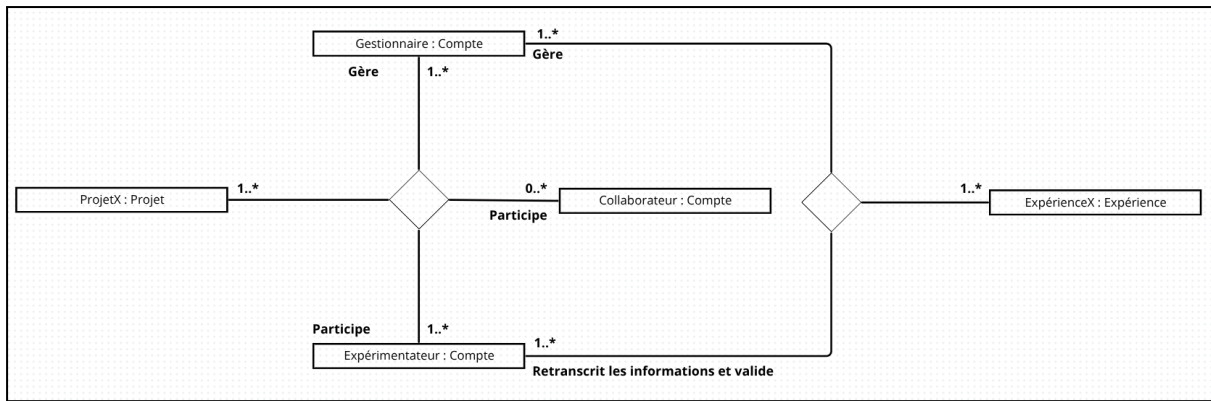
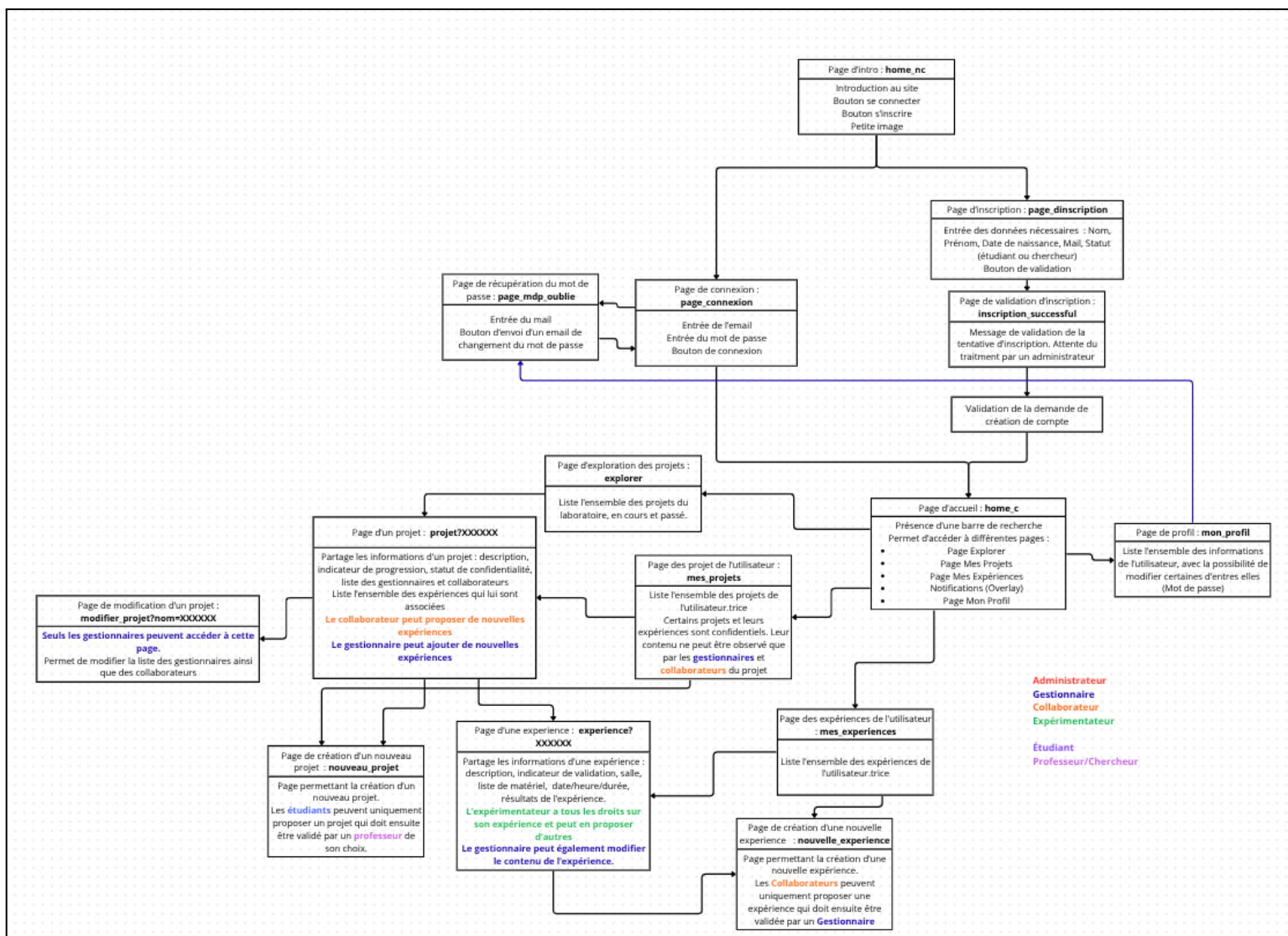
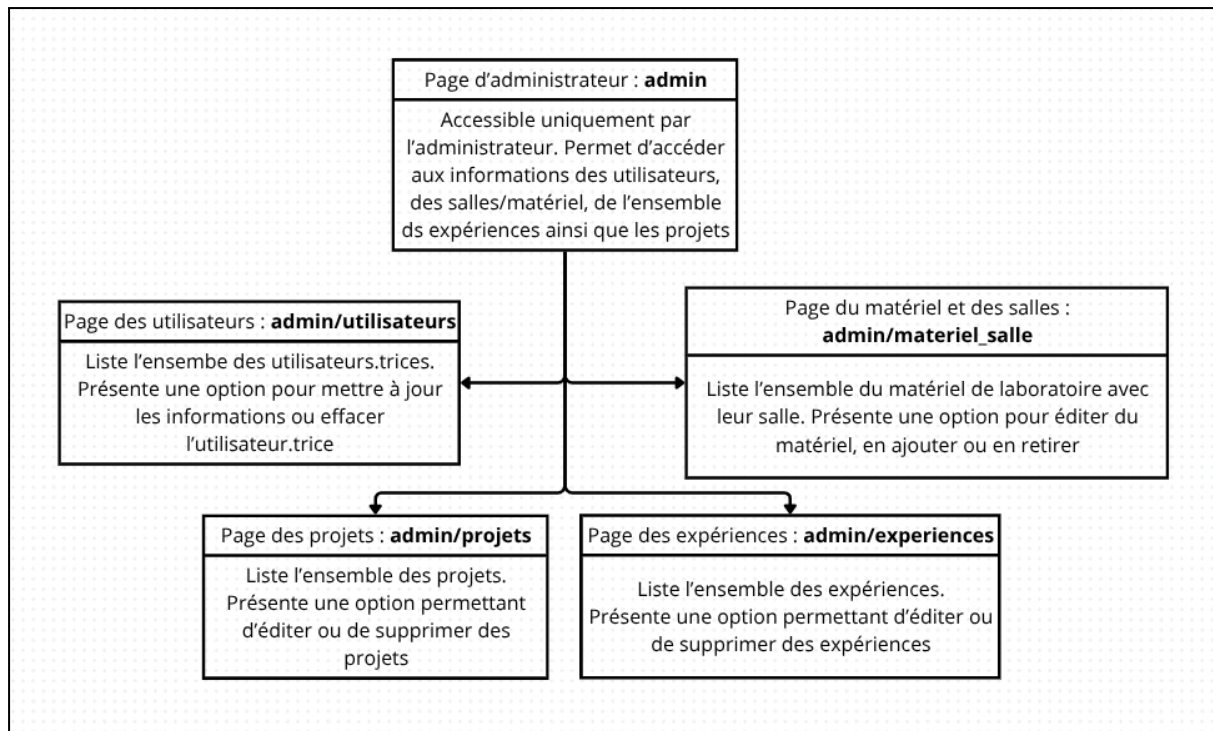


Diagramme d'objets pour la gestion des projets et expériences entre gestionnaire, collaborateurs et expérimentateur



Cheminement des différentes pages du site web et fonction de leur accessibilité.



Cheminement des différentes pages accessibles uniquement à l'administrateur

Conclusion

Pour conclure, ce rapport intermédiaire représente une étape cruciale de la mise en place de notre projet de site web. Ainsi, nous y avons présenté la structure générale avec les différents objets, notamment étroitement liés aux différents utilisateurs de notre site web. Nous avons réalisé également un schéma du cheminement des différentes pages du site. De plus, le schéma de notre base de données a été mis en place, exposant les différentes tables et variables qui seront en communication constante avec le site web. Enfin, nous vous avons énuméré l'ensemble des fonctions qui composeront notre code.

Toutes ces étapes indispensables représenteront un gain de temps, d'efficacité et d'organisation considérable pour la création du script, en facilitant notamment le travail de coopération entre chacun des membres du groupe.