



Cahier des charge Application HomeSkolar

I. Spécification Fonctionnelles

1. Authentification

Création de compte :

En tant qu'utilisateur (élève/tuteur), je veux pouvoir créer un compte.

Critère 1 : Accéder à la page

- **Étant donné** que l'utilisateur souhaite accéder à la page de création de compte.
- **Quand** il clique sur le bouton « S'inscrire »,
- **Alors** il est redirigé sur la page de création de compte

Critère 2 : Possibilité de crée un compte

- **Étant donné** que l'utilisateur ce trouve sur la page de création de compte.
- **Quand** il arrive sur cette même page,
- **Alors** un formulaire de création apparait (login, password, bool : Je suis tuteur).

Critère 3 : Création de compte invalide

- **Étant donné** que l'utilisateur rempli le formulaire de création de compte.
- **Quand** il clique sur « Validé » avec un couple login/password non conforme ou existant,
- **Alors** il est redirigé sur la même page
- **Et** un message d'erreur s'affiche.

Critère 4 : Création de compte valide

- **Étant donné** que l'utilisateur rempli le formulaire de création de compte.
- **Quand** il clique sur le bouton « Validé » et que le couple login/password est conforme
- **Alors** son compte est créé,
- **Et** il est redirigé sur le Dashboard de l'application.

Connexion :

En tant qu'utilisateur (élève/tuteur), je veux pouvoir me connecter afin d'accéder au Dashboard.

Critère 1 : Accéder à la page

- **Étant donné** que l'utilisateur souhaite accéder à la page de connexion.
- **Quand** il clique sur le bouton « Se connecter »,
- **Alors** il est redirigé sur la page de connexion.

Critère 2 : Possibilité de se connecter

- **Étant donné** que l'utilisateur ce trouve sur la page de connexion,
- **Quand** il accède à la page,
- **Alors** un formulaire de connexion apparait (login/password).

Critère 3 : Connexion invalide

- **Étant donné** que l'utilisateur rempli le formulaire de connexion,
- **Quand** il clique sur « Se connecter » avec des informations du formulaire non valide (login, password),
- **Alors** il est redirigé sur la même page
- **Et** un message d'erreur s'affiche.

Critère 4 : Connexion valide

- **Étant donné** que l'utilisateur rempli le formulaire de connexion,

- **Quand** il clique sur le bouton "Se connecter",
- **Alors** la connexion est réussie, et l'utilisateur est redirigé vers le Dashboard de l'application.

Déconnexion :

En tant qu'utilisateur (élève/tuteur), je veux pouvoir me déconnecter.

Critère 1 : Déconnexion valide

- **Étant donné** qu'un utilisateur est connecté,
- **Quand** il clique sur le bouton "Déconnecter",
- **Alors** il est redirigé vers la page d'accueil
- **Et** sa session est détruite.

Modification profile :

En tant qu'utilisateur, je veux pouvoir mettre à jours les données de mon profil.

Critère 1 : Accès à la page de modification du profil

- **Étant donné** que l'utilisateur est connecté à son compte,
- **Quand il souhaite modifier ses informations de profil,**
- **Alors** il doit pouvoir accéder à la page de modification du profil à partir de son tableau de bord ou du menu utilisateur,
- **Et** il ne doit pas pouvoir accéder à la page de modification du profil si l'utilisateur n'est pas connecté (le système doit rediriger l'utilisateur non authentifié vers la page de connexion).

Critère 2 : Possibilité de modifier les informations de mon compte

- **Étant donné** que l'utilisateur est sur la page de modification du profil,
- **Quand** il souhaite mettre à jour ses informations personnelles telles que son username, nom et prénom,
- **Alors** il doit pouvoir modifier ces informations via des champs de formulaire appropriés,
- **Et** les modifications doivent être sauvegardées correctement dans la base de données lorsqu'il soumet le formulaire,
- **Et** l'utilisateur doit recevoir une confirmation que les modifications ont été enregistrées avec succès.

Modification mot de passe :

En tant qu'utilisateur, je veux pouvoir mettre à jours mon mot de passe.

Critère 1 : Affichage du formulaire de changement de mot de passe

- **Étant donné** que l'utilisateur se trouve sur la page de gestion de son profil.
- **Quand** il clique sur le lien « Changer mot de passe »,
- **Alors** un formulaire de changement de mot de passe doit apparaître, comprenant des champs pour l'ancien mot de passe, le nouveau mot de passe, et la confirmation du nouveau mot de passe.

Critère 2 : Changement valides

- **Étant donné** que l'utilisateur remplit correctement les champs de modification de mot de passe
- **Quand** il entre l'ancien mot de passe, un nouveau mot de passe valide (respectant les exigences de sécurité comme la longueur minimale ou les caractères requis), et qu'il confirme correctement le nouveau mot de passe,
- **Alors** le système doit valider les données, mettre à jour le mot de passe dans la base de données et afficher un message de confirmation indiquant que la modification a été effectuée avec succès.

Critère 3 : Modifications invalides

- **Étant donné** que l'utilisateur remplit incorrectement les champs de modification de mot de passe,
- **Quand** il entre un ancien mot de passe incorrect, un nouveau mot de passe qui ne respecte pas les critères de sécurité, ou que la confirmation du nouveau mot de passe ne correspond pas,
- **Alors** le système doit détecter les erreurs, empêcher l'enregistrement des modifications, et afficher un message d'erreur invitant l'utilisateur à corriger les informations.

2. Communication

Voir messages :

En tant qu'élève/tuteur, je veux pouvoir accéder à ma liste de message.

Critère 1 : Possibilité de voir la liste des messages.

- **Étant donné** qu'un élève ou tuteur souhaite envoyer un message,
- **Quand** il clique sur l'icône du Dashboard « messagerie »,
- **Alors** l'utilisateur est redirigé dans la messagerie
- **Et** accède à la liste des messages.

Envoi de messages :

En tant qu'élève/tuteur, je veux pouvoir envoyer des messages pour communiquer avec mon interlocuteur.

Critère 1 : Possibilité de créer un message

- **Étant donné** qu'un utilisateur se trouve sur l'interface de messagerie,
- **Quand** il clique sur le bouton « Nouveau message »,
- **Alors** une fenêtre s'ouvre demandant de renseigner le destinataire et le contenu.

Critère 2 : Envoi valide

- **Étant donné** qu'un élève ou un tuteur est connecté à la plateforme et qu'il se trouve dans la fenêtre de création du message,
- **Quand** il saisit un message dans le champ prévu à cet effet, possède un destinataire et clique sur le bouton « Envoyer »,
- **Alors** le message doit être envoyé à son destinataire
- **Et** apparaître dans l'historique de la conversation en temps réel.

Critère 3 : Envoi non valide

- **Étant donné** qu'un élève ou un tuteur est connecté à la plateforme et qu'il se trouve sur l'interface de messagerie,
- **Quand** le message créé ne possède pas de destinataire et qu'il clique sur le bouton « Envoyer »,
- **Alors** le message n'est pas envoyé
- **Et** un message d'erreur s'affiche.

Réception de messages :

En tant qu'élève/tuteur, je veux recevoir des messages pour rester informé.

Critère 1 : Réception valide

- **Étant donné** qu'un élève ou tuteur est connecté à la plateforme,
- **Quand** un expéditeur envoie un message,
- **Alors** le message doit apparaître directement dans l'interface de l'utilisateur.

Modification d'un message :

En tant qu'élève/tuteur, je veux pouvoir modifier un message envoyé pour corriger une erreur.

Critère 1 : Possibilité de modifier un message envoyé

- **Étant donné** que l'utilisateur a envoyé un message,
- **Quand** il réalise qu'il y a une erreur ou un besoin de clarification dans ce message,
- **Alors** il doit avoir la possibilité de cliquer sur une option « Modifier » à côté du message pour corriger le contenu.

Critère 2 : Modification valide

- **Étant donné** que l'utilisateur a cliqué sur le bouton « Modifier » et qu'il a changé le contenu du message,
- **Quand** il clique sur le bouton « Valider modification »,
- **Alors** le message est modifié.

Critère 3 : Modification invalide

- **Étant donné** que l'utilisateur a cliqué sur le bouton « Modifier » et qu'il ne souhaite plus modifier le message,
- **Quand** il clique sur le bouton « Annulé »,
- **Alors** la modification s'annule.

Suppression d'un message :

En tant qu'élève/tuteur, je veux pouvoir supprimer un message envoyé pour retirer une communication.

Critère 1 : Possibilité de supprimer un message envoyé

- **Étant donné** que l'utilisateur a envoyé un message,
- **Quand** il réalise qu'il n'a plus besoin de ce message ou qu'il ne souhaite plus l'afficher,
- **Alors** il doit avoir la possibilité de cliquer sur une option « supprimer » à côté du message à supprimer.

Critère 2 : Suppression valide

- **Étant donné** que l'utilisateur a cliqué sur l'option « supprimer »,
- **Quand** il clique sur le bouton « Validé » ;
- **Alors** une confirmation est demandée par « Oui/Non » et si « Oui » le message est supprimé de l'interface.

Critère 3 : Suppression invalide

- **Étant donné** que l'utilisateur a cliqué sur l'option « supprimer »,
- **Quand** il clique sur le bouton « Annulé »,
- **Alors** la suppression du message s'annule.

Marquage des messages comme lus :

En tant qu'élève/tuteur, je veux pouvoir marquer un message comme lu pour différencier les messages consultés des autres.

Critère 1 : Possibilité de marquage manuel comme lu

- **Étant donné** que l'utilisateur consulte sa messagerie,
- **Quand** il souhaite marquer manuellement un message comme lu,

- **Alors** il doit avoir la possibilité de cliquer sur une option "Marquer comme lu" à côté du message pour changer son état.
- **Et** le boolean « is_pinned » doit être mis à jour à true.

Critère 2 : Possibilité de marquage automatique comme lu

- **Étant donné** que l'utilisateur a reçu un message,
- **Quand** il ouvre le message.
- **Alors** le message passera automatiquement en lu
- **Et** le booléen « is_pinned » doit être mis à jour à true.

Marquage des messages comme non lus :

En tant qu'élève/tuteur, je veux pouvoir marquer un message comme non lu.

Critère 1 : Possibilité de marquage manuel comme non lu

- **Étant donné** que l'utilisateur consulte sa messagerie,
- **Quand** il souhaite marquer manuellement un message comme non lu,
- **Alors** il doit avoir la possibilité de cliquer sur une option « Marquer comme non lu » à côté du message pour changer son état.
- **Et** le booléen « is_pinned » doit être mis à jour à false.

Critère 2 : Indicateur visuel des messages non lus

- **Étant donné que** l'utilisateur parcourt une conversation,
- **Quand** il y a des messages non lus,
- **Alors** un indicateur visuel (comme une couleur différente, un badge "Non lu", ou une icône) doit apparaître à côté des messages non lus.

Épingler des messages importants :

En tant qu'élève/tuteur, je veux pouvoir épingler des messages importants pour y accéder rapidement.

Critère 1 : Possibilité d'épingler des messages

- **Étant donné** qu'un utilisateur se trouve sur l'interface de messagerie et peut consulter ces messages,
- **Quand** il sélectionne et accède aux options disponibles,
- **Alors** l'option « Épingler » doit être visible et accessible pour permettre à l'utilisateur de marquer le message comme important.

Critère 2 : Epingler message valide

- **Étant donné** qu'un utilisateur consulte un message,
- **Quand** il clique sur le bouton « épingler »,
- **Alors** le message reste accessible en priorité dans sa messagerie,
- **Et** le bool « is_pinned » doit être mis à jour à true.

Critère 3 : Gestion des messages épinglés

- **Étant donné** que l'utilisateur a épinglé un message,
- **Quand** il n'a plus besoin de ce message comme référence,
- **Alors** il doit avoir la possibilité de cliquer sur une option "Déépingler" pour retirer le message de la section des messages *épinglés*.

3. Gestion des rendez-vous



Voir rendez-vous :

En tant qu'élève/tuteur, je souhaite voir la liste de mes rendez-vous.

Critère 1 : Accéder à la page

- **Étant donné** qu'un utilisateur est connecté,
- **Quand** il clique sur l'icône « rendez-vous »,
- **Alors** il est redirigé sur la page de rendez-vous.

Critère 2 : Possibilité de voir les éléments de la page rendez-vous

- **Étant donné** qu'un élève ou tuteur souhaite consulter ses rendez-vous à venir,
- **Quand** il accède à la page des rendez-vous,
- **Alors** seuls les rendez-vous à venir (non passés) sont affichés, triés par date et heure.

Création de rendez-vous :

En tant que tuteur je veux pouvoir créer un rendez-vous avec mon élève afin de le rencontrer.

Critère 1 : Possibilité de planifier un rendez-vous

- **Étant donné** que l'utilisateur se trouve sur la page des rendez-vous,
- **Quand** il clique sur le bouton « Créer rendez-vous »,
- **Alors** un formulaire de création de rendez-vous s'affiche permettant d'entrer la date, l'heure, de choisir un élève les détails du rendez-vous.

Critère 2 : Création de rendez-vous invalide

- **Étant donné** qu'un tuteur remplit le formulaire de création de rendez-vous,
- **Quand** il essaie de valider le formulaire sans renseigner tous les champs obligatoires (date, heure, élève, détails),
- **Alors** un message d'erreur s'affiche sous chaque champ manquant pour informer le tuteur de compléter le formulaire.

Critère 3 : Création de rendez-vous valide

- **Étant donné** que le tuteur a rempli correctement le formulaire,
- **Quand** il clique sur le bouton "Confirmer",
- **Alors** le rendez-vous est enregistré dans le système,
- **Et** le booléen « on_hold » doit être égal à true.

Modification de rendez-vous :

En tant qu'utilisateur je veux pouvoir modifier mes rendez-vous.

Critère 1 : Possibilité de modification du rendez-vous

- **Étant donné** que l'utilisateur a un rendez-vous planifié,
- **Quand** il souhaite modifier la date, l'heure, ou les détails,
- **Alors** il doit avoir la possibilité de cliquer sur une option "Modifier" à côté du rendez-vous pour ajuster les informations nécessaires (date, heure, etc.)

Critère 2 : Modification valide

- **Étant donné** que l'utilisateur modifie un rendez-vous,
- **Quand** il clique sur le bouton « Validé »,
- **Alors** ces modifications doivent être enregistrées et appliquées à la fiche du rendez-vous.
- **Et** l'utilisateur doit de nouveau accepter le rendez-vous.

Critère 3 : Modification invalide

- **Étant donné** que l'utilisateur modifie un rendez-vous,
- **Quand** il tente de valider les modifications sans avoir rempli tous les champs obligatoires (date, heure, lieu, etc.) ou avec des données incorrectes,
- **Alors** un message d'erreur doit s'afficher sous chaque champ incorrect ou manquant, empêchant la validation tant que les informations ne sont pas correctement renseignées.

Suppression de rendez-vous :

En tant que tuteur je veux pouvoir supprimer mes rendez-vous avec un élève.

Critère 1 : Possibilité de supprimer un rendez-vous

- **Étant donné** qu'un tuteur a planifié un rendez-vous avec un élève,
- **Quand** il souhaite supprimer le rendez-vous
- **Alors** il doit avoir la possibilité de cliquer sur une option « Supprimer »

Critère 2 : Validation de la suppression

- **Étant donné** qu'un tuteur choisi de supprimer un rendez-vous avec un élèves,
- **Quand** il clique sur le bouton « Supprimer »
- **Alors** une fenêtre de confirmation doit apparaitre pour s'assurer qu'il souhaite bien annuler définitivement le rendez-vous.

Critère 3 : Confirmation de la suppression valide

- **Étant donné** qu'une fenêtre de confirmation de suppression définitive apparait,
- **Quand** il clique sur « Validé »,
- **Alors** le rendez-vous doit être supprimé.

Critère 4 : Confirmation de la suppression non valide

- **Étant donné** qu'une fenêtre de confirmation de suppression définitive apparait,
- **Quand** il clique sur « Annulé »,
- **Alors** la suppression du rendez-vous est annulé et la fenêtre de confirmation ce détruit.

Accepter rendez-vous :

En tant qu'élève ou tuteur, je veux pouvoir accepter un rendez-vous pour confirmer ma participation.

Critère 1 : Possibilité d'accepter un rendez-vous

- **Étant donné** qu'un utilisateur reçoit une invitation à un rendez-vous,
- **Quand** il souhaite confirmer sa participation,
- **Alors** il doit avoir la possibilité de cliquer sur une option "Accepter" à côté de l'invitation pour valider sa disponibilité.

Critère 2 : Acceptation valide

- **Étant donné** qu'un utilisateur clique sur le bouton « Accepter » pour un rendez-vous,
- **Quand** il confirme son acceptation dans l'interface,
- **Alors** le rendez-vous doit être marqué comme accepté dans le système, et une notification de confirmation doit être envoyée à l'autre participant.

Critère 3 : Mise à jour du statut du rendez-vous

- **Étant donné** que l'utilisateur accepte un rendez-vous,
- **Quand** l'acceptation est confirmée,
- **Alors** le statut du rendez-vous doit être mis à jour pour refléter qu'il est "Confirmé" ou "Accepté" dans la liste des rendez-vous de l'utilisateur.

Refus de rendez-vous :

En tant qu'élève ou tuteur, je veux pouvoir refuser un rendez-vous.

Critère 1 : Possibilité de refuser un rendez-vous

- **Étant donné** qu'un élève ou un tuteur reçoit une invitation à un rendez-vous,
- **Quand** il accède à la notification ou à l'interface de gestion des rendez-vous,
- **Alors** il doit avoir la possibilité de cliquer sur un bouton « Refuser » ou « Annuler » pour signaler qu'il ne souhaite pas participer au rendez-vous.

Critère 2 : Refus valide

- **Étant donné** qu'un élève ou un tuteur clique sur le bouton « Refuser » pour un rendez-vous,
- **Quand** il confirme son refus dans l'interface de gestion des rendez-vous,
- **Alors** le rendez-vous doit être marqué comme « Refusé » ou « Annulé » dans le système, et une notification de refus doit être envoyée à l'autre participant pour l'informer de l'annulation du rendez-vous.

Critère 3 : Mise à jour du statut du rendez-vous

- **Étant donné** que l'utilisateur refuse un rendez-vous,
- **Quand** le refus est confirmé,
- **Alors** le statut du rendez-vous doit être mis à jour pour refléter qu'il est "Non-confirmé" ou "Refusé" dans la liste des rendez-vous de l'utilisateur.

4. Gestion des tâches

Voir tâche memo :

En tant qu'élève/tuteur, je souhaite voir la liste de mes tâches.

Critère 1 : Accès à la page de création

- **Étant donné** qu'un utilisateur est connecté,
- **Quand** il clique sur l'icône « tâche »,
- **Alors** il doit être redirigé vers la page des tâches/mémos,

Crée tâche mémo :

En tant qu'utilisateur, je veux pouvoir créer des tâches ou mémos pour organiser mes activités et rappels.

Critère 1 : Possibilité de créer une tâche

- **Étant donné** que l'utilisateur se trouve sur l'interface de gestion des tâches/mémos,
- **Quand** il clique sur le bouton « Créer une tâche » ou « Nouveau mémo »,
- **Alors** un formulaire de création de tâche ou mémo s'affiche, permettant à l'utilisateur d'entrer les détails requis comme le titre, la description, la date d'échéance (si applicable), et toute autre information pertinente.

Critère 2 : Création de tâche ou mémo valide

- **Étant donné** que l'utilisateur remplit le formulaire de création de tâche ou mémo,
- **Quand** il clique sur le bouton « Enregistrer » ou « Ajouter » après avoir rempli tous les champs requis,
- **Alors** la tâche ou le mémo doit être enregistré dans le système, et un message de confirmation doit s'afficher, indiquant que la tâche ou le mémo a été créé avec succès
- **Et** l'état du bool « is_memo » doit être mis à jour à true..

Critère 3 : Création de tâche ou mémo invalide

- **Étant donné** que l'utilisateur remplit le formulaire de création de tâche ou mémo,
- **Quand** il essaie de valider le formulaire sans renseigner tous les champs obligatoires (comme le titre, la description, ou d'autres informations requises),
- **Alors** un message d'erreur doit s'afficher sous chaque champ manquant pour informer l'utilisateur de compléter le formulaire avant de pouvoir enregistrer la tâche ou le mémo.

Modifier tâche memo :

En tant qu'utilisateur (tuteur ou élève), je veux pouvoir modifier des tâches ou mémos existants pour mettre à jour les informations.

Critère 1 : Possibilité de modifier une tâche ou un mémo

- **Étant donné** qu'un utilisateur est connecté et a accès à l'interface de gestion des tâches/mémos,

- **Quand** il visualise une tâche ou un mémoire existant et souhaite le modifier,
- **Alors** il doit voir une option ou un bouton « Modifier » à côté de la tâche ou du mémoire pour accéder au formulaire de modification.

Critère 2 : Modification de tâche ou mémoire valide

- **Étant donné** que l'utilisateur (tuteur ou élève) accède au formulaire de modification d'une tâche ou mémoire,
- **Quand** il modifie les détails requis (comme le titre, la description, la date d'échéance, etc.) et clique sur le bouton « Enregistrer » ou « Mettre à jour »,
- **Alors** les modifications doivent être enregistrées dans le système,
- **Et** un message de confirmation doit s'afficher, indiquant que la tâche ou le mémoire a été mis à jour avec succès.

Critère 3 : Modification de tâche ou mémoire invalide

- **Étant donné** que l'utilisateur (tuteur ou élève) accède au formulaire de modification d'une tâche ou mémoire,
- **Quand** il essaie de valider le formulaire sans renseigner tous les champs obligatoires ou avec des informations invalides,
- **Alors** un message d'erreur doit s'afficher sous chaque champ manquant ou incorrect pour informer l'utilisateur de compléter ou corriger les informations avant de pouvoir enregistrer les modifications.

Critère 4 : Modification par un tuteur des tâches d'un élève

- **Étant donné** qu'un tuteur est connecté et a accès aux tâches des élèves,
- **Quand** il accède à la tâche d'un élève et clique sur l'option « Modifier »,
- **Alors** il doit pouvoir modifier les détails de la tâche de l'élève et enregistrer les modifications,
- **Et** ces modifications doivent être reflétées dans l'interface de gestion des tâches de l'élève.

Supprimer tâche memo :

En tant qu'utilisateur, je veux pouvoir supprimer des tâches ou des mémoires

Critère 1 : Possibilité de supprimer une tâche ou un mémoire

- **Étant donné** qu'un utilisateur est connecté et se trouve sur l'interface de gestion des tâches/mémoires,
- **Quand** il visualise une tâche ou un mémoire existant,
- **Alors** il doit voir une option ou un bouton « Supprimer » à côté de la tâche ou du mémoire pour accéder à la fonction de suppression.

Critère 2 : Suppression de tâche ou mémoire valide

- **Étant donné que l'utilisateur clique sur le bouton « Supprimer » à côté d'une tâche ou d'un mémoire,**
- **Quand** il confirme la suppression dans la boîte de dialogue de confirmation,
- **Alors** la tâche ou le mémoire doit être définitivement supprimé du système, et un message de confirmation doit s'afficher pour indiquer que la suppression a été effectuée avec succès.

Critère 3 : Suppression de tâche invalide

- **Étant donné** que l'utilisateur clique sur le bouton « Supprimer » à côté d'une tâche ou d'un mémoire,
- **Quand** il clique sur « annuler » dans la boîte de dialogue de confirmation,
- **Alors** la tâche n'est pas supprimé de la base de donnée.

Attribuer tâche memos :

En tant que tuteur, je veux pouvoir attribuer une tâche à un élève.

Critère 1 : Possibilité d'attribuer une tâche ou un mémoire

- **Étant donné** qu'un tuteur est connecté en tant que tuteur et se trouve sur l'interface de gestion des tâches/mémoires,
- **Quand** il clique sur l'option pour créer ou modifier une tâche ou un mémoire,
- **Alors** il doit voir une option pour attribuer cette tâche ou ce mémoire à un ou plusieurs élèves.

Critère 2 : Attribution de tâche ou mémoire valide

- **Étant donné** que le tuteur remplit le formulaire de création ou de modification d'une tâche ou d'un mémoire,
- **Quand** il sélectionne un ou plusieurs élèves à qui attribuer la tâche ou le mémoire et clique sur le bouton « Attribuer »,
- **Alors** la tâche ou le mémoire doit être attribué aux élèves sélectionnés, avec l'attribut « is_memo » égal true,
- **Et** un message de confirmation doit s'afficher pour indiquer que l'attribution a été effectuée avec succès.

Critère 3 : Attribution de tâche ou mémoire invalide

- **Étant donné** que le tuteur remplit le formulaire de création ou de modification d'une tâche ou d'un mémoire,
- **Quand** il tente d'attribuer la tâche ou le mémoire sans sélectionner d'élèves ou avec des informations invalides,
- **Alors** un message d'erreur doit s'afficher pour informer le tuteur de compléter les informations requises avant de pouvoir attribuer la tâche ou le mémoire.

Critère 4 : Accès à l'option d'attribution

- **Étant donné** qu'un utilisateur est connecté en tant qu'élève,
- **Quand** il accède à l'interface de gestion des tâches/mémoires,
- **Alors** il ne doit pas voir l'option pour attribuer des tâches, car cette fonctionnalité est réservée aux tuteurs.

Marquer compléter :

En tant que utilisateur je veux pouvoir marquer une tâche memo comme complété.

Critère 1 : Possibilité de marquer une tâche mémo comme complété

- **Étant donné** qu'un utilisateur est connecté et se trouve sur l'interface de gestion des tâches/mémos,
- **Quand** il visualise une tâche mémo,
- **Alors** il doit voir une option ou un bouton « Marquer comme complété » à côté de du mémo pour effectuer cette action.

Critère 2 : Marquage memo comme complété valide

- **Étant donné** qu'un utilisateur clique sur le bouton « Marquer comme complété » et que le bool « is_memo » est égal à true,
- **Quand** il confirme son action dans la boîte de dialogue de confirmation
- **Alors** la tâche ou le mémo doit être marqué comme complété dans le système, et l'état du bool « completed » doit être égal à true.

Critère 3 : Marquage tâche assigné a élève

- **Étant donné** qu'un tuteur est connecté et se trouve sur l'interface de gestion des tâches,
- **Quand** il visualise une tâche qu'il a assignée à un élève,
- **Alors** il doit voir une option ou un bouton « Marquer comme complété » à côté de cette tâche pour effectuer cette action au nom de l'élève.

Critère 4 : Marquage de tâche assignée à un élève valide

- **Étant donné** qu'un tuteur clique sur le bouton « Marquer comme complété » pour une tâche assignée à un élève,
- **Quand** il confirme son action dans la boîte de dialogue de confirmation,
- **Alors** la tâche doit être marquée comme complétée dans le système, avec la mise à jour « is_completed » égal à true,
- **Et** un message de confirmation doit s'afficher indiquant que la tâche a bien été complétée pour l'élève.

5. Notification :

Envoyer notification :

En tant qu'utilisateur, je veux être notifié des événements importants tels que les messages, les rendez-vous et les tâches.

Critère 1 : Possibilité d'envoyer une notification type message

- **Étant donné** qu'un message est reçu par un utilisateur,
- **Quand** le système détecte la réception du message,
- **Alors** une notification doit être générée et envoyée à l'utilisateur pour l'informer de ce nouveau message.

Critère 2 : Notification valide pour un message

- **Étant donné** qu'un message a été envoyé à un utilisateur,
- **Quand** la notification pour ce message est envoyée,
- **Alors** l'utilisateur doit recevoir la notification contenant les détails du message et doit pouvoir voir un aperçu ou l'intégralité du message, selon les paramètres de la notification.
- **Et** le booléen « is_sent » doit être mis à jour à true pour indiquer que la notification a été envoyée.

Critère 3 : Possibilité d'envoyer une notification type rendez-vous

- **Étant donné** qu'un rendez-vous est planifié, modifié, accepté ou supprimé,
- **Quand** le système reçoit une de ces informations,
- **Alors** une notification doit être générée et envoyée à l'utilisateur pour l'informer du rendez-vous prévu.

Critère 4 : Notification valide pour un rendez-vous

- **Étant donné** qu'un rendez-vous est planifié, modifié, accepté ou supprimé,
- **Quand** la notification pour ce rendez-vous est envoyée,
- **Alors** l'utilisateur doit recevoir la notification contenant les détails du rendez-vous, comme la date, les participants, l'heure et l'url de la session.
- **Et** le booléen « is_sent » doit être mis à jour à true pour indiquer que la notification a été envoyée.

Critère 5 : Possibilité d'envoyer une notification type tâche

- **Étant donné** qu'une tâche est assignée, modifiée ou supprimée,
- **Quand** le système enregistre la tâche ou la retire,
- **Alors** une notification doit être générée et envoyée à l'utilisateur pour l'informer de la tâche assignée, mise à jour ou supprimer.

Critère 6 : Notification valide pour une tâche

- **Étant donné** qu'une tâche est assignée, modifiée ou supprimée,
- **Quand** la notification pour cette tâche est envoyée,
- **Alors** l'utilisateur doit recevoir la notification contenant les détails de la tâche, tels que le titre, la description et la date limite.
- **Et** le booléen « is_sent » doit être mis à jour à true pour indiquer que la notification a été envoyée.

Marquer notification comme lu :

En tant qu'utilisateur je veux pouvoir marquer une notification comme « lu ».

Critère 1 : Accès à la Page des Notifications

- **Étant donné** que l'utilisateur est connecté à son compte,
- **Quand** il clique sur l'icône ou le lien des notifications depuis le tableau de bord ou le menu utilisateur,
- **Alors** il doit être redirigé vers la page des notifications où il pourra consulter toutes ses notifications récentes.

Critère 2 : Possibilité de Marquer une Notification Comme « Lu »

- **Étant donné** que l'utilisateur se trouve sur la page des notifications,
- **Quand** il clique sur une notification non lue,
- **Alors** le système doit marquer cette notification comme « lu » en la distinguant visuellement (par exemple, en la grisant ou en supprimant l'indicateur de non-lu) et en mettant à jour son statut dans la base de données.

Critère 3 : Marquer comme « lu » valide

- **Étant donné** que l'utilisateur clique sur une notification non lue,
- **Quand** le système la marque correctement comme « lu », que l'indicateur visuel change et que le booléen « is_read » est mis à true dans la base de données,
- Alors l'utilisateur doit voir la notification différenciée des autres non lues, et la base de données doit refléter ce changement en ayant le champ « is_read » de la notification à true.

II. Veille technologique

1. Authentification

1.1. Création de compte et Connexion/Déconnexion

Technologies Clés :

→ **Django pour la gestion des utilisateurs** : Utilisation du framework Django, qui offre une gestion intégrée des utilisateurs, y compris la création de comptes, l'authentification, et la gestion des sessions.

Sources :

- **Django Documentation** : <https://docs.djangoproject.com/en/5.1/> ;
- **Real Python** : <https://realpython.com/> ;
- **Stack Overflow** : <https://stackoverflow.com/> ;

Justification :

- **Django Documentation** est la source la plus fiable pour comprendre les capacités natives du framework ;
- **Real Python** est reconnu pour ses tutoriels pratiques et à jour, ce qui est crucial pour l'implémentation ;
- **Stack Overflow** est essentiel pour obtenir des solutions aux problèmes pratiques rencontrés lors du développement ;

1.2. Sécurité des mots de passe

Technologies Clés :

→ **Hashage sécurisé (PBKDF2, Argon2, bcrypt)** : Django utilise PBKDF2 par défaut, mais peut être configuré pour utiliser d'autres algorithmes de hash.

Sources :

- **OWASP** : <https://owasp.org/> ;
- **Django Security Documentation** : <https://docs.djangoproject.com/en/5.1/topics/security/> ;

Justification :

- **OWASP** est la référence mondiale en matière de sécurité des applications web ;
- **Django Security Documentation** est indispensable pour une implémentation conforme aux meilleures pratiques ;

2. Communication

2.1. Messagerie

Technologies Clés :

→ **Django channels** : Les messages sont envoyés et reçus en temps réel via WebSockets grâce à Django Channels.

Sources :

- **Django channels documentation** : <https://channels.readthedocs.io/en/latest/> ;

Justification :

→ **Django channels documentation** : Fournit les informations nécessaires pour implémenter la messagerie en temps réel via WebSockets, assurant une communication instantanée et fluide dans l'application.

3. Notifications

4.1. Notifications de tâches et de messages

Technologies Clés :

→ **Python** : Pour la gestion des notifications côté serveur, y compris la création, la mise à jour, et la gestion des notifications dans la base de données.

→ **JavaScript** : Pour l'affichage dynamique des notifications dans l'interface utilisateur, y compris la gestion des actions utilisateur comme marquer les notifications comme lues.

Sources :

→ **Python.org** : <https://www.python.org/doc/> ;

→ **Real Python** : <https://realpython.com/> ;

→ **JavaScript_info** : <https://javascript.info/> ;

Justification :

→ **Real Python** est reconnu pour ses tutoriels pratiques et à jour, ce qui est crucial pour l'implémentation ;

→ **JavaScript** offre une mise à jour dynamique de l'interface utilisateur, améliorant l'interaction avec les notifications.

4. base de donnée

5.1. Gestion de la base de données avec Django

Technologies Clés :

→ **PostgreSQL** : Bases de données Open-source relationnelles compatibles avec Django.

Sources :

→ **PostgreSQL Documentation**: <https://www.postgresql.org/> ;

Justification :

→ **PostgreSQL** documentation est la documentation officiel permettant de rester à jours des dernières version.

Autre sources pertinente au projet : <https://stackoverflow.com/> ; <https://www.reddit.com/> ;

III. Spécifications Techniques

Front end

Technologies : Développement en **HTML**, **CSS** et **JavaScript** avec un **framework** moderne tel que **React**.

Justification :

→ **HTML**, **CSS**, et **JavaScript** sont les bases pour créer des sites web. **React** facilite la création d'interfaces utilisateur dynamiques qui réagissent rapidement sans rechargement complet de la page, améliorant ainsi l'expérience utilisateur.

Responsive Design : Conception responsive pour garantir une utilisation fluide sur **desktop** et **mobile**.

Justification :

→ Assure que l'application s'adapte automatiquement à la taille de l'écran, offrant une expérience cohérente sur tous les appareils.

Pages Dédiées : Les tâches, rendez-vous et messages auront chacun leur propre page.

Justification :

→ Permet une gestion et une visualisation claires et organisées des différentes fonctionnalités, améliorant l'ergonomie et la facilité d'utilisation de l'application.

Back end

Technologies : Développement en Python avec un framework tel que Django.

Justification :

→ Python est un langage puissant pour le développement, et Django simplifie la création d'applications web robustes avec des fonctionnalités prêtes à l'emploi.

Base de données : Utilisation d'une base de données PostgreSQL.

Justification :

→ PostgreSQL est une base de données fiable et extensible, idéale pour gérer de grandes quantités de données en toute sécurité.

Gestion des Notifications : Implémentation des notifications en temps réel en utilisant **Django Channels** pour le backend (Python) et JavaScript pour le frontend.

Justification :

→ Permet l'envoi et la réception instantanés des notifications grâce à l'intégration des WebSockets, offrant une réactivité élevée et une expérience utilisateur fluide.

IV. Diagramme UML

