INSTRUCTIF

I. DESCRIPTION DE L'APPLICATION ET DE SES REGLES METIERS

INSTRUCT'IF est une application qui connecte des intervenants (étudiants, enseignants ou autres professionnels, comme des retraités) avec des élèves ayant besoin de cours de soutien en visioconférence.

Cette application peut être utilisée par deux types d'utilisateurs : les élèves et les intervenants.

Du point de vue de l'élève : L'utilisateur peut créer un compte ou se connecter s'il en possède déjà un. Pour créer un compte, il doit fournir son nom, son prénom, son adresse mail, sa date de naissance (format yyyy-mm-dd), son mot de passe, le code de son établissement scolaire et sa classe (de 0 à 6, 0 correspondant à la Terminale, 6 à la Sixième). Une fois ces informations fournies et à condition qu'aucun autre compte avec la même adresse mail n'existe déjà, le compte de l'élève est enregistré dans la base de données. Pour se connecter à l'application, il doit entrer son adresse mail et son mot de passe. Une fois connecté, l'élève peut accéder à son profil, où il peut demander un cours. Lorsqu'il souhaite demander un nouveau cours, il accède à la liste de toutes les matières disponibles, choisit une matière et rédige une courte description de sa demande. Un intervenant est alors sélectionné selon sa disponibilité (l'intervenant doit être disponible au moment de la demande), son niveau de qualification et le nombre d'interventions qu'il a déjà réalisées (si plusieurs intervenants sont disponibles on choisit celui qui a effectué le moins de soutien). Si le cours est lancé, l'élève peut quitter la visioconférence et évaluer le cours qu'il vient d'effectuer. L'élève peut quitter l'application à tout moment.

Du point de vue de l'intervenant : Les intervenants bénévoles sont déjà enregistrés dans la base de données de l'application, il n'est donc pas possible de créer un nouveau compte intervenant. Si l'utilisateur est un intervenant bénévole, il peut s'authentifier avec son adresse mail et son mot de passe. Une fois connecté, il a accès à son tableau de bord où il peut consulter l'historique de tous les cours qu'il a donnés. Il a également accès à des statistiques globales du programme de soutien : nombre total de cours donnés, durée moyenne d'un cours, nombre d'établissements scolaires dans chaque ville, et établissements ayant un Indice de Performance Scolaire (IPS) bas. Si un cours lui est proposé, il a la possibilité de lancer la visioconférence. Il peut décider de mettre fin à un cours. Lorsqu'un cours se termine, il doit rédiger un bilan du cours, qui sera ensuite envoyé à l'élève. L'intervenant peut quitter l'application à tout moment.

II. MODELE DU DOMAINE

Voici ci-dessous le diagramme de classe des différents objets constituant l'application INSTRUCT'IF.

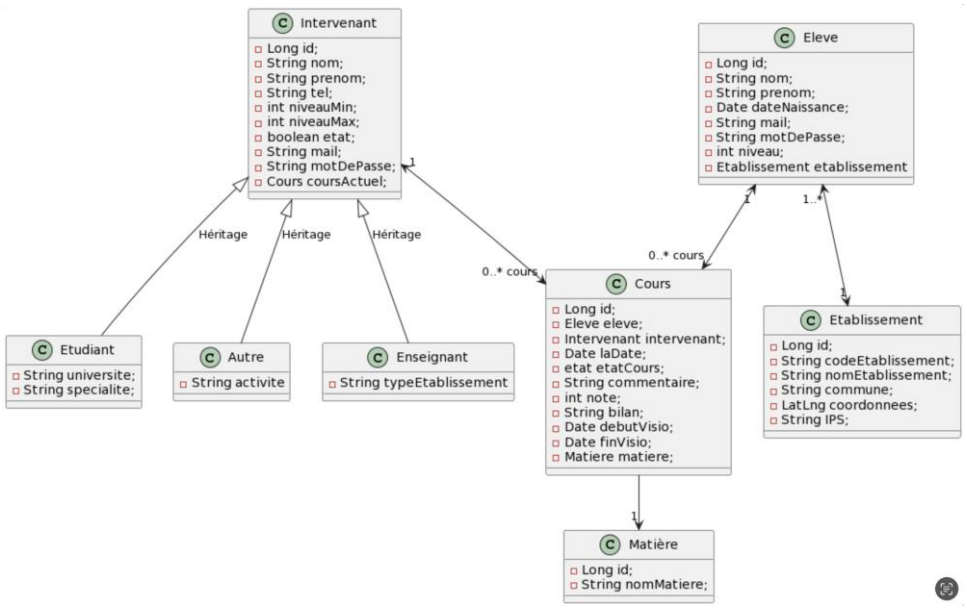

Figure 1: Diagramme de classes de l'application INSTRUCT'IF

III. CONCEPTION DES IHMS ET DES SERVICES

III.1. IHM

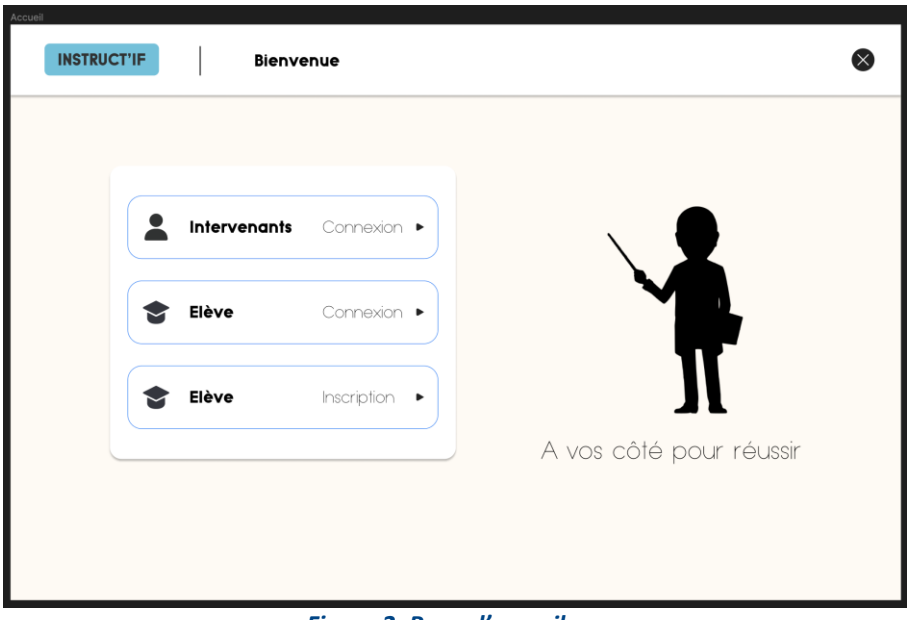

Figure 2: Page d'accueil

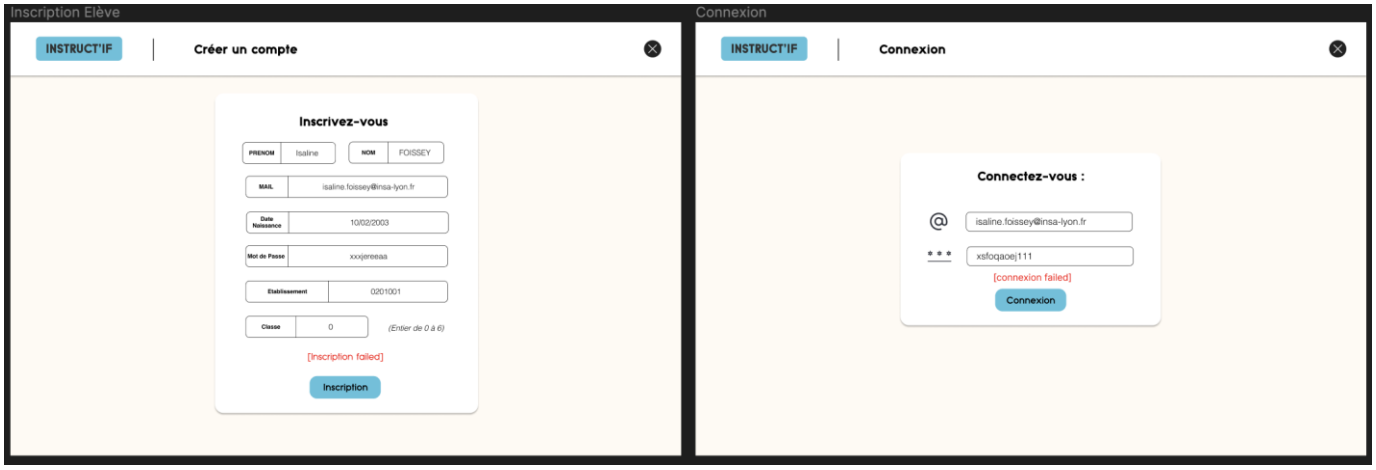

Figure 3 : Page d'inscription élève et page connexion élève

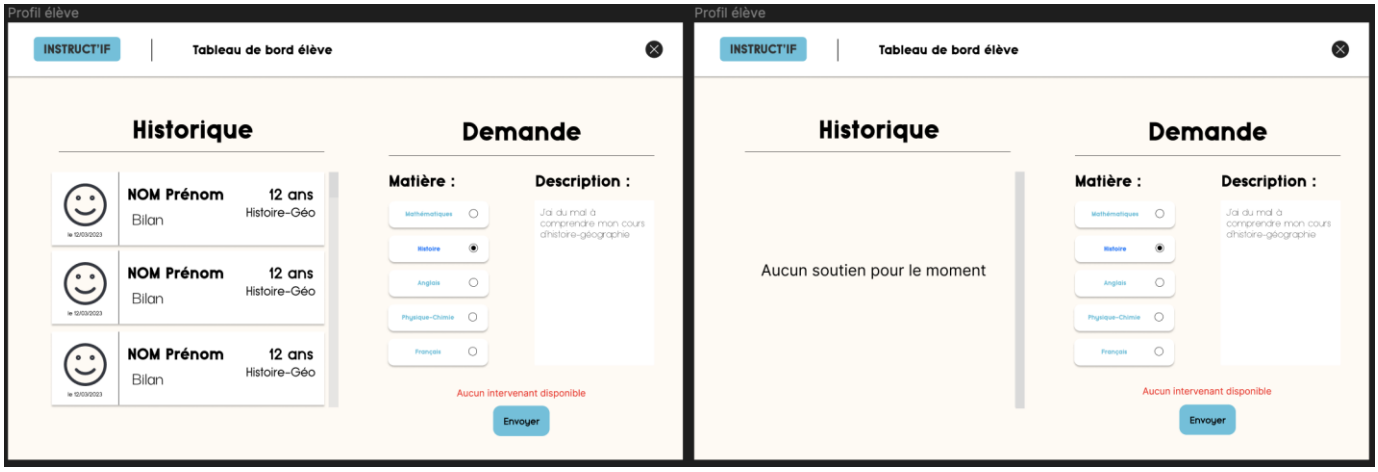

Figure 4 : Page tableau de bord élève

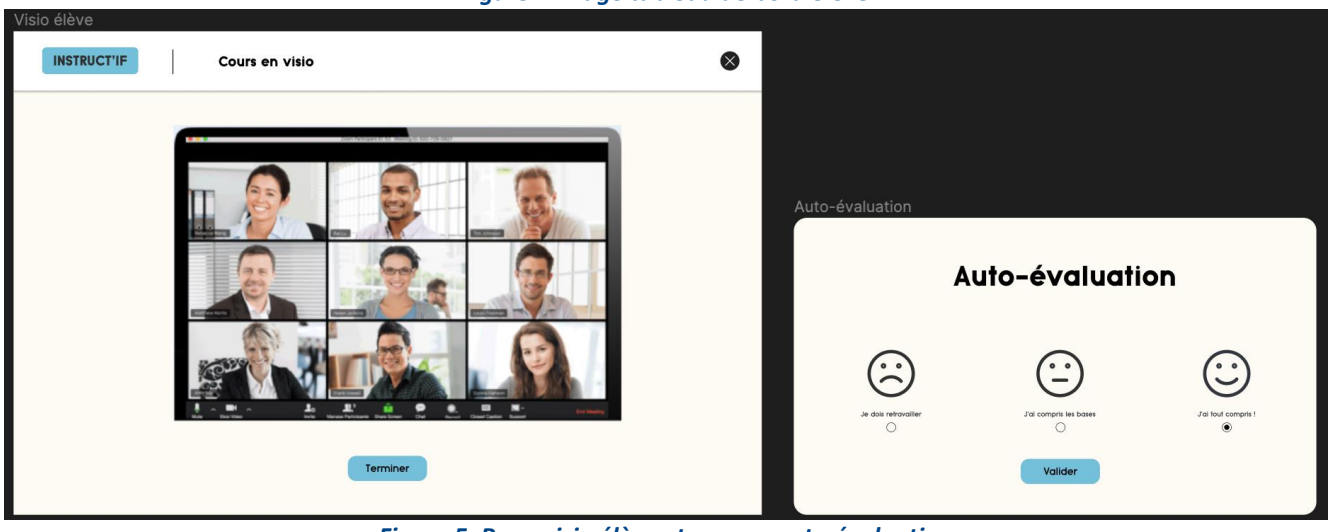

Figure 5: Page visio élève et pop-up auto-évaluation

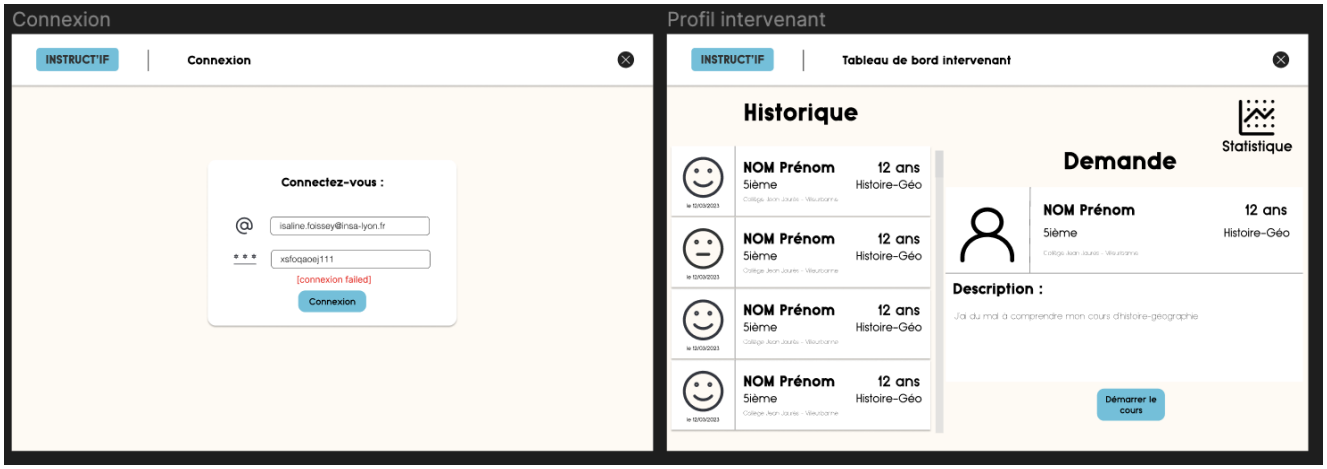

Figure 6: Page connexion intervenant et tableau de bord intervenant 1

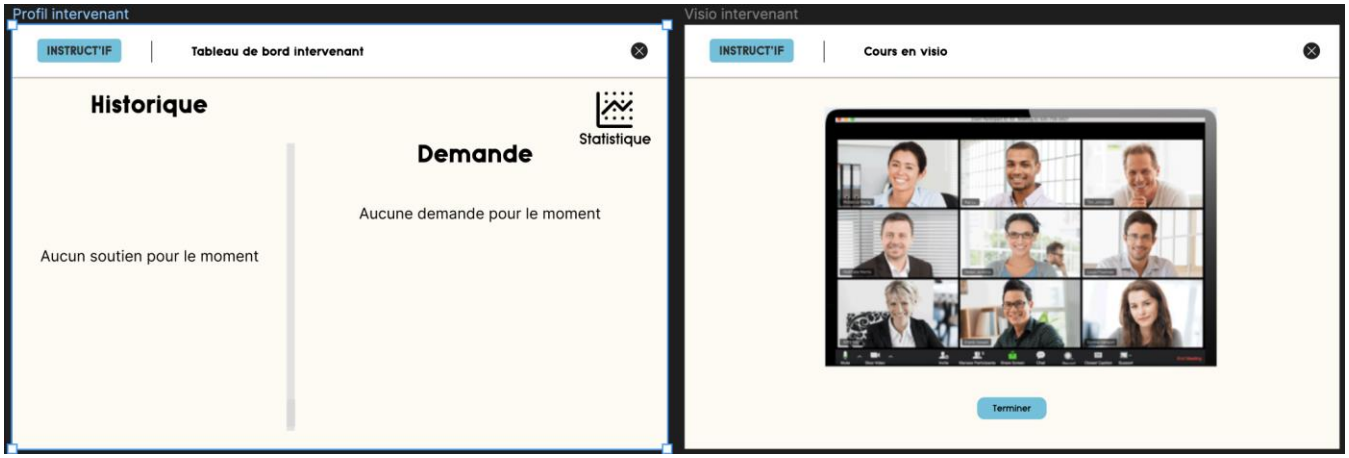

Figure 7: Page tableau de bord intervenant 2 et page visio intervenant

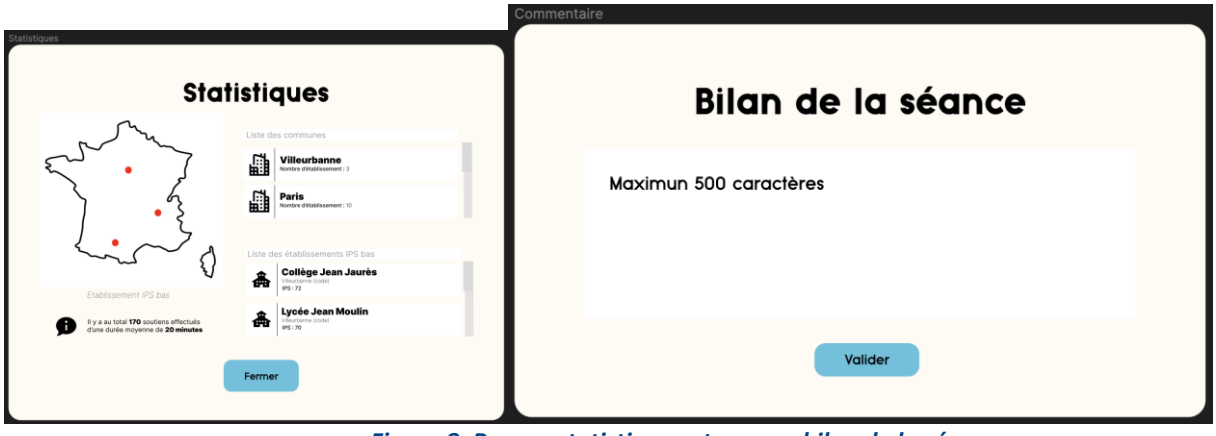

Figure 8: Pop-up statistiques et pop-up bilan de la séance

Pour plus de détails sur l'IHM, vous pouvez consultez le diagramme d'enchainement de fenêtres et également la version interactive de cet IHM disponible sur le lien suivant : [figma]

III.2. ICARS

A. Page d'accueil et Page inscription élève

Intention	Contrôle	Action	Réponse et Services
Appel aux services d'initialisation	-	-	Appel aux services "InitialiserMatiere()" et "InitialiserIntervenant()" pour initialiser la base de données.
Se connecter en tant qu'intervenant	Bouton "Intervenants: Connexion"	Clic	Redirection vers la page de connexion pour les intervenants sans animation de transition.
Se connecter en tant qu'élève	Bouton "Élève: Connexion"	Clic	Redirection vers la page de connexion pour les élèves sans animation de transition.
S'inscrire en tant qu'élève	Bouton "Élève: Inscription"	Clic	Redirection vers la page d'inscription pour les élèves sans animation de transition.
-	-	-	Message de bienvenue affiché sur la page d'accueil.

Intention	Contrôle	Action	Réponse et Services
S'inscrire	Formulaire avec champs de saisie et bouton de validation	Clic	Les champs de saisie sont mis à blanc et disponibles pour être remplis par l'utilisateur.
Valider inscription	Bouton "Inscription"	Clic	Appel au service inscrireEleve(Eleve eleve, String code) avec vérification des champs de saisie non vide et en format adéquat. Si l'inscription est réussie, envoie d'un e-mail de confirmation et redirection vers la page de connexion sans pop-up. En cas d'échec, affichage d'un message d'erreur et envoie d'un e-mail d'information d'échec au client.
Revenir à l'accueil	Bouton INSTRUCT'IF	Clic	Retourne à la page d'accueil de l'application sans fenêtre pop-up ni transition.

Figure 9: Tableau ICARS de la page d'accueil et de la page inscription élève

B. Page connexion élève et Page tableau de bord élève

Intention	Contrôle	Action	Réponse et Services
Se connecter	Formulaire de connexion avec champs de saisie	-	Les champs de saisie pour l'adresse e-mail et le mot de passe sont disponibles pour être remplis par l'utilisateur. Aucune action de mise à blanc n'est mentionnée dans l'interface utilisateur, donc cette partie est omise.
Valider la connexion	Bouton "Connexion"	Clic	Appel au service authentifierEleve(String mail, String motDePasse) pour vérifier que les champs de saisie ne sont pas vides et que les identifiants sont corrects. Si l'authentification est réussie, l'utilisateur est redirigé vers la page menu de l'élève. En cas d'échec, un message d'échec de connexion est affiché sans pop-up.
Revenir à l'accueil	Bouton INSTRUCT'IF	Clic	L'utilisateur est redirigé directement à la page d'accueil de l'application sans pop-up ni transition.

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	Affichage de l'historique en utilisant le service getHistoriqueEleve(Eleve e). Si l'historique est vide, le message "Aucun soutien pour le moment" est affiché. La liste des matières est générée en utilisant le service consulterListeMatieres().
Demander un soutien	Bouton "Envoyer"	Clic	Appel au service demanderCours(Eleve eleve, String nomMatiere, String message). En cas de succès, redirection vers la page de visioconférence de l'élève. En cas d'échec, affichage du message "Aucun intervenant disponible".
Revenir à l'accueil	Bouton INSTRUCT'IF	Clic	L'utilisateur est redirigé directement à la page d'accueil de l'application sans pop-up ni transition.

Figure 10: Tableau ICARS de la page connexion élève et tableau de bord élève

C. Page visio élève

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	La session de visioconférence démarre et l'interface d'attente est affichée jusqu'à ce que l'intervenant rejoigne. Ici, la visio est simulée par une image fixe.
Arrêter la visio	Bouton "Terminer"	Clic	Appel au service terminerVisio(Cours c) pour terminer la session de visioconférence. Ensuite, l'utilisateur est redirigé vers un pop-up pour effectuer une auto-évaluation.

Figure 11: Tableau ICARS de la page visio élève

D. Pop-up auto-évaluation

Intention	Contrôle	Action/Evt	Réponse
Envoyer l'autoévaluation	Bouton "Valider"	Clic	Appel au service finVisioEleve(eleve, evaluation) pour enregistrer l'autoévaluation de l'élève. Si l'opération est réussie, redirection vers le tableau de bord de l'élève. Sinon, affichage d'un message d'erreur.

Figure 12: Tableau ICARS du pop-up auto-évaluation

E. Page connexion intervenant

Intention	Contrôle	Action	Réponse et Services
Se connecter	Formulaire de connexion avec champs de saisie	-	Les champs de saisie pour l'adresse e-mail et le mot de passe sont disponibles pour être remplis par l'utilisateur. Aucune action de mise à blanc n'est mentionnée dans l'interface utilisateur, donc cette partie est omise.
Valider la connexion	Bouton “Connexion”	Clic	Appel au service `authentifierIntervenant(String mail, String motDePasse)` pour vérifier que les champs de saisie ne sont pas vides et que les identifiants sont corrects. Si l'authentification est réussie, l'utilisateur est redirigé vers la page menu de l'intervenant. En cas d'échec, un message d'échec de connexion est affiché sans pop-up.
Revenir à l'accueil	Bouton INSTRUC'IF	Clic	L'utilisateur est redirigé directement à la page d'accueil de l'application sans pop-up ni transition.

Figure 13: Tableau ICARS de la page connexion intervenant

F. Page tableau de bord intervenant 1 et 2

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	Utilisation de consulterHistoriqueIntervenant(Intervenant) pour afficher l'historique des soutiens de l'intervenant. Utilisation de CheckDemandeSoutien(Intervenant) pour vérifier si l'intervenant à un soutien en attente. Si aucun soutien en attente on affiche “Aucune
Voir les statistiques	Bouton "Statistique"	Clic	L'utilisateur est redirigé vers l'overlay des statistiques.
Revenir à l'accueil	Bouton INSTRUC'IF	Clic	L'utilisateur est redirigé directement à la page d'accueil de l'application sans pop-up ni transition.

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	Utilisation de consulterHistoriqueIntervenant(Intervenant) pour afficher l'historique des soutiens de l'intervenant. Utilisation de CheckDemandeSoutien(Intervenant) pour vérifier si l'intervenant à un soutien en attente. Si aucun soutien en attente on affiche “Aucune
Démarrer le cours	Bouton "Démarrer le cours"	Clic	L'utilisateur est redirigé vers la page visio intervenant.
Voir les statistiques	Bouton "Statistique"	Clic	L'utilisateur est redirigé vers l'overlay des statistiques.
Revenir à l'accueil	Bouton INSTRUC'IF	Clic	L'utilisateur est redirigé directement à la page d'accueil de l'application sans pop-up ni transition.

Figure 14: Tableau ICARS de la page tableau de bord intervenant 1 et 2

G. Pop-up statistiques

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	**Utilisation de** `statNbEtablissementParCommune()`, `statEtablissementIPS()`, et `statNbSoutienetDuree()`
Parcourir les établissements ayant un IPS bas	ScrollBar	Scroll	Parcours la liste des établissement ayant un IPS bas
Parcourir les villes dans lesquelles se trouve des établissement bénéficiant du programme de soutien	ScrollBar	Scroll	Parcours la liste des villes bénéficiant dans lesquelles se trouvent des établissements ayant des élèves inscrits.
Fermer	Fermer	Clic	Aller à la page tableau de bord intervenant

Figure 15: Tableau ICARS du pop-up statistiques

H. Page visio intervenant

Intention	Contrôle	Action/Evt	Réponse
Initialisation	-	-	Utilisation de lancerVisio(Cours). Ici la visio est simulée par une image fixe.
Arrêter la visio	Bouton "Terminer"	Clic	Appel au service terminerVisio(Cours c) pour terminer la session de visioconférence. Ensuite, l'utilisateur est redirigé vers la page bilan de la séance.

Figure 16: Tableau ICARS de la page visio intervenant

I. Pop-up bilan de la séance

Intention	Contrôle	Action/Evt	Réponse
Rédiger un bilan	Formulaire de connexion avec champs de saisie	Clic	Les champs de saisie sont mis à blanc et disponibles pour être remplis par l'utilisateur.
Valider	Bouton valider	Clic	Appel au service envoyerBilan(Cours c, String bilan) pour envoyer un bilan de la séance à l'élève. Aller à la page tableau de bord intervenant

Figure 17 : Tableau ICARS du pop-up bilan de la séance

III.3. Services

Chaque service possède également une gestion d'erreur avec un try-catch permettant de roll-back des transactions ouvertes. En cas d'erreur lors de l'opération, l'erreur est enregistrée, ce qui permet de signaler un problème sans affecter la stabilité de la base de données.

1. *Services orientés élève*

Signature du service : inscrireEleve(Eleve eleve, String code) : boolean

Description du service : Permet d'inscrire un nouvel élève en associant un établissement à son profil et en créant une entrée dans la base de données pour lui.

Algorithme :

- *Création d'un DAO pour l'élève (EleveDao) et un autre pour l'établissement (EtablissementDao).*
- *Tentative de récupération de l'établissement par son code à l'aide du EtablissementDao.*
- *Si l'établissement n'est pas trouvé, utilisation de l'API EducNetApi pour obtenir les informations du lycée ou collège par son code.*
- *Obtention des coordonnées géographiques de l'établissement via GeoNetApi en envoyant le nom de l'établissement et la commune.*
- *Création d'un nouvel objet Etablissement avec les informations récupérées.*
- *Association de l'établissement à l'élève.*
- *Ouverture d'une transaction pour l'inscription de l'établissement (si nouvellement créé) et de l'élève dans la base de données.*
- *Validation de la transaction et fermeture du contexte de persistance.*
- *Envoi d'un e-mail de confirmation à l'élève si l'inscription est réussie.*
- *En cas d'exception ou d'erreur, annulation de la transaction, fermeture du contexte de persistance et envoi d'un e-mail d'échec d'inscription à l'élève.*
- *Retour de la valeur booléenne indiquant le succès ou l'échec de l'opération.*

Signature du service : authentifierEleve(String mail, String motDePasse) : Eleve

Description du service : Authentifie un élève en vérifiant ses identifiants de connexion. Le service renvoie l'objet Eleve si l'authentification est réussie, sinon null.

Signature du service : consulterListeEleves() : List<Eleve>

Description du service : Récupère la liste de tous les élèves enregistrés dans la base de données. Le service renvoie la liste des élèves récupérée, ou null en cas d'erreur.

Signature du service : demanderCours(Eleve eleve, String nomMatiere, String message) : Cours

Description du service : Permet à un élève de demander un cours dans une matière donnée, avec un message spécifique à transmettre à l'intervenant. Le service cherche un intervenant disponible dans la base de données. Si plusieurs intervenants sont disponibles, il prend le premier intervenant qui a effectué le moins de soutien. Le service renvoie le cours créé si la demande a été traitée avec succès, sinon null en cas d'exception.

Signature du service : consulterListeMatieres() : List<Matiere>

Description du service : Récupère et retourne la liste complète des matières disponibles dans la base de données. Le service retourne la liste des matières obtenue ou null si une exception est survenue pendant l'opération.

Signature du service : terminerVisio(Cours c) : Cours

Description du service : Permet de marquer la fin d'une session de visioconférence pour un cours donné, enregistrant l'heure de fin et changeant l'état du cours à terminé. Le service renvoie l'instance de Cours mise à jour si la session de visioconférence a été correctement terminée, sinon null en cas d'échec.

Signature du service : getHistoriqueEleve(Eleve e) : List<Cours>

Description du service : Récupère et retourne l'historique complet des sessions de soutien auxquelles un élève donné a participé. Le service renvoie une liste de Cours qui représente l'historique complet des sessions de soutien terminés de l'élève.

2. Services orientés intervenant

Signature du service : lancerVisio(Cours cours) : Cours

Description du service : Ce service lance une session de visioconférence pour un cours en attente attribué à un intervenant spécifique. Le service renvoie l'instance de Cours mise à jour si la session de visioconférence a été lancée avec succès, sinon null en cas d'échec.

Signature du service : terminerVisio(Cours cours) : Cours

Description du service : Permet de marquer la fin d'une session de visioconférence pour un cours donné, enregistrant l'heure de fin et changeant l'état du cours à terminé. Le service renvoie l'instance de Cours mise à jour si la session de visioconférence a été correctement terminée, sinon null en cas d'échec.

Signature du service : consulterListeIntervenants() : List<Intervenant>

Description du service : Fournit une liste de tous les intervenants inscrits dans la base de données. La méthode renvoie la liste des intervenants si l'opération est réussie, ou null si une exception est levée durant le processus.

Signature du service : EnvoyerBilanCours(Cours c, String bilan) : void

Description du service : Ce service est responsable d'envoyer par mail le bilan d'un cours à l'élève concerné. Le bilan récapitule les points clés et les recommandations suite à une session de soutien. En plus de l'envoi du bilan, le service met à jour le statut de l'intervenant pour le rendre disponible pour de futures sessions et enregistre le bilan dans l'historique du cours. Algorithme :

- *Identification et récupération des détails du cours et de l'intervenant concerné via leurs DAO respectifs.*
- *Composition d'un message de bilan contenant les détails pertinents du cours et les observations personnalisées.*
- *Envoi du message de bilan à l'élève via un service de messagerie électronique.*
- *Mise à jour du cours avec le bilan fourni et réinitialisation de l'état de l'intervenant (disponible pour de nouvelles sessions).*
- *Enregistrement des modifications dans la base de données en validant la transaction.*
- *Affichage d'un message de confirmation de l'envoi du bilan et de la mise à jour du statut de l'intervenant.*

Signature du service : checkDemandeSoutien(Intervenant intervenant) : Object[]

Description du service : Vérifie l'existence d'une demande de soutien actuellement assignée à un intervenant et récupère les informations pertinentes pour cette demande. Le service retourne un tableau contenant le cours actuel et l'élève associé, ou null si aucune demande n'est en cours ou en cas d'erreur.

Signature du service : getHistoriqueIntervenant(Intervenant intervenant) : List<Cours>

Description du service : Fournit une liste détaillée de tous les cours dispensés par un intervenant spécifique, permettant un aperçu complet de son activité au sein du programme de soutien. Le service renvoie une liste de Cours qui représente l'historique complet des sessions de soutien terminés de l'intervenant.

Signature du service : statNbEtablissementParCommune() : List<Object[]>

Description du service : Calcule et retourne le nombre d'établissements par commune, fournissant une vue d'ensemble utile pour l'analyse de la distribution géographique des établissements bénéficiant du soutien. Le service retourne une liste d'objets, chaque objet représentant une commune et le nombre d'établissements correspondants, ou null en cas d'erreur.

Signature du service : statEtablissementIPS() : List<Etablissement>

Description du service : Identifie les établissements où au moins un soutien a été effectué avec un Indice de Position Sociale bas, offrant une perspective sur les zones potentiellement prioritaires pour le soutien. Le service renvoie une liste des établissements ayant un IPS bas, ou null en cas d'échec de l'opération.

Signature du service : statNbSoutienetDuree() : Object[]

Description du service : Fournit des statistiques sur le nombre total de soutiens réalisés et la durée moyenne des sessions, permettant une évaluation de l'efficacité et de la portée du programme de soutien. Le service renvoie un tableau d'objets contenant le nombre total de soutiens et la durée moyenne des sessions, ou null en cas d'échec de l'opération.

3. Services d'initialisation

Signature du service : initialiserIntervenant() : boolean

Description du service : Ce service est conçu pour initialiser la base de données avec un ensemble d'intervenants prédéfinis. Le service renvoie true si l'initialisation s'est déroulée sans erreur, false si une exception a été levée et la transaction annulée.

Signature du service : initialiserMatiere() : boolean

Description du service : Initialise la base de données avec une liste prédéfinie de matières scolaires. Le service renvoie true si l'initialisation des matières a été réussie, false si une exception a été levée et que la transaction a été annulée.