Chloé CHABAL, Widad FARHAT
B3124

TP DASI

Introduction

L'objectif de ce TP est de développer une application, INSTRUCT'IF, permettant d'aider les élèves de collège et de lycée à faire leur devoir. Pour réaliser cela, il a fallu spécifier le modèle, les services et les IHM.

1. Description de l'application

L'application met à disposition des élèves des séances de soutien scolaire en visio avec des intervenants. Les élèves peuvent faire des demandes de soutien sur l'application dans n'importe quelle matière.

Les élèves peuvent s'inscrire sur l'application. Pour cela, les élèves ont un formulaire à remplir en ligne et doivent fournir les informations suivantes : leur nom, leur prénom, leur date de naissance, le code de l'établissement, leur classe ainsi qu'une adresse mail et un mot de passe. Une fois l'inscription faite, les élèves peuvent s'authentifier avec leur adresse mail et leur mot de passe. Quand ils sont connectés, les élèves peuvent créer une demande et consulter l'historique des demandes.

Les intervenants sont des étudiants, des enseignants ou d'autres personnes souhaitant faire du soutien. Ils peuvent s'authentifier sur l'application avec leur login et leur mot de passe. Une fois connectés, les intervenants peuvent voir l'historique des interventions et consulter les statistiques.

Un élève peut faire une demande de soutien, quand il le souhaite. Il sélectionne la matière et décrit sa demande, puis il valide sa demande. L'application va essayer de trouver un intervenant disponible dont le niveau correspond à celui de l'élève et son expérience. Une fois l'intervenant choisi, il peut consulter le profil de l'élève et les détails de la demande avant de lancer la visio.

L'intervenant va apporter de l'aide à l'élève pendant la visio, jusqu'à ce que l'élève se sente rassuré. A la fin de la visio, l'élève est invité à s'auto-évaluer par rapport à la demande.

2. Règles métier

Pour nous permettre de développer notre application, nous avons dû définir différentes règles métier. Par rapport aux intervenants,

- Tous les intervenants peuvent prendre en charge toutes les matières scolaires.
- Les intervenants sont déjà inscrits, on ne traite pas le cas de l'inscription d'un intervenant. Ils peuvent seulement s'authentifier.
- Un intervenant est considéré comme occupé dès qu'une demande lui est attribué.

Par rapport aux demandes,

- Les demandes de soutien sont traitées immédiatement. Si aucun intervenant est trouvé, la demande est annulée. L'élève voit le résultat de sa demande dans une fenêtre pop-up.
- La visio est simulée par l'affichage d'une image dans la page Web. L'élève peut mettre fin à la visio uniquement.
- Une demande possède une note, donnée à la fin de la visio, qui est comprise entre 1 et 5.

Par rapport aux élèves,

- Les adresses mails des élèves sont supposées correctes et uniques. L'inscription d'un élève échoue donc s'il existe un autre élève avec la même adresse mail.
- Les élèves sont inscrits dans des établissements français.
- Un élève ne fait qu'une seule demande à la fois.

3. Modèle du domaine

Nous avons développé les objets métier de l'application qui sont Eleve, Intervenant, Etudiant, Enseignant, Autre, Etablissement, Demande et Matiere (Figure 1).

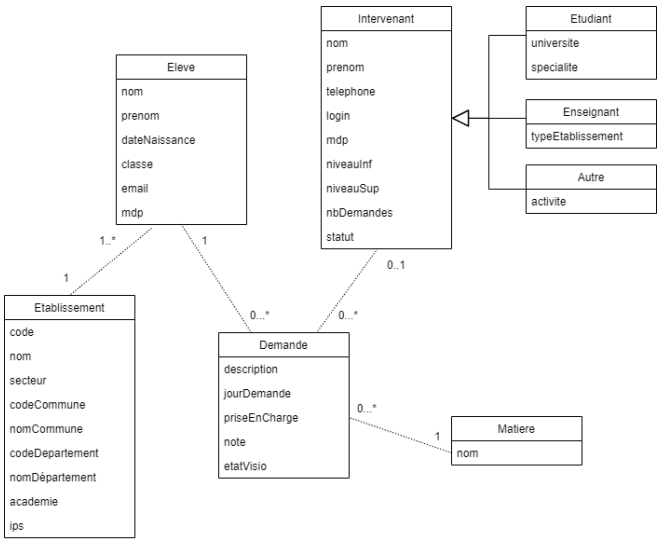

Figure 1 : Modèle conceptuel de données de l'application

- Classes Matiere, Eleve, Demande et Etablissement

Nous avons une classe Eleve qui permet de modéliser un élève. Celui-ci est persisté dans la base de données si son inscription réussit.

Ensuite, nous avons une classe Etablissement, qui permet de modéliser un établissement scolaire français. Un établissement est entré dans la base de données lorsqu'un élève s'inscrit, si son établissement n'est pas déjà dans la base.

La classe Demande permet de représenter chaque demande, avec son jour de création, sa description et la note la caractérisant. L'historique des demandes d'un élève prend en compte uniquement celles qui ont été réalisées. L'attribut priseEnCharge permet donc de savoir si la demande a fini d'être traitée (true), c'est-à-dire si la visio est finie, il vaut false jusqu'à ce que la visio soit finie. De plus, l'attribut etatVisio permet de savoir si la visio a été lancée (true) ou non (false) par l'intervenant.

La classe Matiere est quant à elle, une classe avec un seul attribut son nom.

- Classe Intervenant

Nous avons choisi de modéliser les différents intervenants, Enseignant Etudiant ou Autre, en faisant un héritage. Ainsi, même si chaque type d'intervenant a sa particularité aux niveaux des attributs, leur fonction globale reste commune et facilite l'implémentation. Ils partagent le plus grand nombre des attributs.

Nous avons également fait le choix d'ajouter l'attribut statut dans l'objet Intervenant pour indiquer si ce dernier est libre (true) ou non (false). Un intervenant devient occupé dès qu'une demande lui est attribuée et est libre dès qu'il a fini la visio avec l'élève.

- Ajout d'une classe non-métier

Nous avons choisi d'ajouter une classe utile Classe qui permet de faire le lien entre les classes en string et leur correspondance en Integer. Ainsi, la comparaison entre Integer est plus simple à implémenter. Les classes vont de 0 à 6 en nombre entier.

4. Conception des services métiers

Grâce à l'User Story, nous avons conçu un diagramme Use Case qui nous a permis de déterminer les différents services à développer.

- Service gérant l'inscription de l'élève : boolean inscrireEleve(Eleve eleve, String code)

Grâce à ce service, un élève peut créer un compte sur l'application INSTRUCT'IF. Le service reçoit en entrée un code d'établissement et un objet Eleve qui se caractérise par un nom, prénom, date de naissance, classe. Au début, l'attribut concernant son établissement est « null ».
Un élève pourra être entré dans la base de données si son adresse mail est unique. Sinon, l'inscription échouera. Un mail sera envoyé en fonction du résultat.
Un établissement est entré dans la base de données lorsqu'un élève s'inscrit, si son établissement n'est pas déjà dans la base.
Le service renvoie un booléen permettant de savoir si l'inscription a fonctionné ou non.

Pseudo-code :

```
Fonction inscrireEleve(Elve eleve, String code)-> boolean
        Initialiser la variable sortie
        Créer un contexte de persistance
        Rechercher si le code correspond à un établissement
        Si l'établissement n'existe pas
                Créer un nouvel établissement
        Ouvrir une transaction
        Faire persister l'élève et l'établissement s'il n'existait pas
        Définir l'établissement de l'élève
        Valider la transaction
        Associer l'établissement à l'élève
        Si Transaction validée
                Envoyer un mail de bienvenue à l'élève
                Mettre la variable sortie à true
        Sinon
                Envoyer un mail d'échec à l'élève
                Annuler la transaction
                Mettre la variable sortie à false
        Fermer le contexte de persistance
        Retourner la variable sortie
```

- Service gérant l'authentification d'un élève : Eleve authentifierEleve(String mail, String motDePasse)

Ce service permet à un élève de s'authentifier pour pouvoir accéder aux différentes fonctionnalités de l'application. Le service va vérifier que l'adresse mail et le mot de passe passés en paramètre sont présents dans la base de données. Si c'est le cas, l'élève authentifié est renvoyé sinon on renvoie null.

- Service gérant l'authentification d'un Intervenant : Intervenant authentifierIntervenant(String login, Strong motDePasse)

Ce service permet à un intervenant de s'authentifier pour pouvoir accéder aux différentes fonctionnalités de l'application. Le service va vérifier que le login et le mot de passe passés en paramètre sont présent dans la base de données. Si c'est le cas, l'intervenant authentifié est renvoyé sinon on renvoie null. L'algorithme de ce service est semblable au précédent en remplaçant Eleve par Intervenant.

- Service gérant la création de demande : boolean creerDemande(Matiere matiere, String description, Eleve eleve)

Ce service permet à un élève de créer une demande. Pour cela, l'élève va choisir une matière dans le menu déroulant et fournir une description concernant sa demande. En entrée, le service prend donc une matière, une description et un élève.

Une demande sera entrée dans la base de données uniquement si un intervenant est trouvé en fonction de son expérience, de sa disponibilité et du niveau de l'élève. On prend l'intervenant avec le moins de demandes réalisées. Une fois l'intervenant trouvé, on change son statut en occupé (true), on augmente d'un son nombre de demandes et on associe l'intervenant à la demande. On persiste alors la demande dans la base de données.

On renvoie un booléen : true si la demande a été créée et false si on n'a pas trouvé d'intervenant.

Pseudo-code :

```
Fonction creerDemande(Eleve eleve, String descriptif, Matiere matiere)-> boolean
        Initialiser la variable sortie à false
        Créer un objet demande avec les attributs : matière, description, élève et jourDemande
        Trouver un intervenant libre avec le niveau compatible avec l'élève et avec le moins de demandes
        Si un intervenant a été trouvé
                Associer l'intervenant à la demande
                Augmenter la note de l'intervenant
                Définir le statut de l'intervenant a false (pris)
                Envoyer une notification à l'intervenant
                Ouvrir une transaction
                Faire persister la demande et mettre à jour l'intervenant
                Valider la transaction
                Mettre la variable sortie à true
        Si Transaction non valide
                Annuler la transaction
        Fermer le contexte de persistance
        Renvoyer la variable sortie
```

- Service gérant la gestion des demandes du point de vue de l'intervenant : Demande consulterDemandeEnCoursIntervenant(Intervenant intervenant)

Ce service permet de trouver la demande que doit traitée un intervenant. Pour cela, le service cherche dans la base de données la demande liée à l'intervenant où la prise en charge est toujours à false.

Le service retourne la demande en cours, ce qui permet à l'intervenant de regarder le descriptif de la demande et le profil de l'élève.

- Service gérant la demande en cours du point de vue de l'élève : Demande consulterDemandeEnCoursEleve(Eleve eleve)

Ce service permet de trouver la demande en cours d'un élève. Pour cela, le service cherche dans la base de données la demande liée à l'élève où la prise en charge est toujours à false.

Le service retourne la demande en cours. Ce service va être utile lorsque nous voudrons savoir si l'intervenant a lancé la visio (etatVisio == true).

- Service gérant le début d'une visio : boolean lancerVisioIntervenant(Demande demande)

Après avoir vu le profil de l'élève et le descriptif de la demande, l'intervenant va lancer la visio en appuyant sur un bouton. Ce service sera appelé pour mettre à jour l'état de la visio et pour lancer le processus gérant la visio. Le service renvoie true si la modification de la demande a bien été prise en compte.

- Service gérant la fin d'une visio : boolean arreterVisio(Demande demande)

Lorsque l'élève met fin à la visio, il va appuyer sur le bouton raccrocher. Ainsi, ce service va être appelé. Le statut de l'intervenant va être modifié ainsi que l'état de la visio et la prise en charge de la demande. Cela signifie que l'intervenant peut avoir une nouvelle demande car celle-ci est terminée. Le service renvoie true si la modification a réussi.

- Service gérant l'évaluation d'une demande : boolean evaluerDemande(Demande demande, Integer note)

Lorsque l'élève a fini la visio, il doit s'autoévaluer. Pour cela, une fenêtre avec des smileys s'affiche. Lorsque l'élève va cliquer sur un des smileys, ce service va être appelé. La demande va être modifiée dans la base en fonction de la note choisie par l'élève. Le service renvoie true si la modification a réussi.

- Service gérant la consultation des demandes réalisées par l'élève : List<Demande> genererHistoriqueDemandesEleve(Eleve eleve)

Ce service permet de récupérer tout l'historique des demandes réalisées par l'élève. Le service va parcourir la base de données et récupérer uniquement les demandes faites par l'élève dans l'ordre de date décroissant. Si l'élève n'a pas de demande, le service renvoie null.

- Service gérant la consultation des demandes faites par l'intervenant : List<Demande> genererHistoriqueDemandesIntervenant(Intervenant intervenant)

Ce service permet de récupérer tout l'historique des demandes réalisées par l'intervenant. Le service va parcourir la base de données et récupérer uniquement les demandes faites par l'intervenant dans l'ordre de date décroissant. Si l'intervenant n'a pas réalisé de demande, le service renvoie null.

- Service gérant la liste des matières : List<Matiere> genererListeMatieres()

Ce service permet de trouver toutes les matières se trouvant dans la base de données pour permettre l'affichage du menu déroulant.

- Services gérant les statistiques : int genererNombreDemandesTotales(), List<Object[]> genererStatistiqueDemandeParClasse(), List<Object[]> genererStatistique-DemandeParMatiere(), List<Object[]> genererStatistiqueDemandeParCodeIPS() et List<Object[]> genererStatistiqueDemandeParDepartement()

L'application veut permettre aux intervenants de voir des statistiques. Pour cela, nous avons donc plusieurs services en fonction des différentes statistiques.

- Services permettant de créer la base de données : boolean creerBDIntervenant() et boolean creerBDMatiere()

Ces services permettant d'initialiser notre base de données avec des intervenants déjà inscrits et des matières prédéfinies.

- Services permettant de retrouver les objets métier grâce à leur Id : Matiere trouverMatiereParID(Long idMatiere), Demande trouverDemandeParId(Long idDemande), Intervenant trouverIntervenantParId(Long idIntervenant) et Eleve trouverEleveParId(Long idEleve)

Ces services permettent de retrouver un objet métier dans la base de données à partir de leur id.

5. Conception des IHM

Dans ce qui suit, la page d'accueil définit la première page que l'on voit en ouvrant l'application, celle où l'on peut s'authentifier ou s'inscrire si on est un élève. La page principale est la page affichée une fois qu'on a réussi l'authentification. Nous pouvons accéder aux différentes fonctionnalités propres à chaque type. (Annexe 1 : Description de l'IHM)

- Inscription des élèves (Figure 2)

Intention	Contrôle	Action	Réponse
Initialisation			Page d'accueil - Affichage des champs à modifier de l'authentification de l'élève et de l'intervenant et passage de ces champs à modifiable - Activation du bouton Inscription, des boutons CONNEXION élève et intervenant et bouton «cacher mdp»
Inscription	Bouton inscription	Clic	Affichage de la page d'inscription d'un élève - Affichage des champs à modifier - Passage des champs à modifiable - Activation du bouton Valider, bouton pour avoir la liste des classes «cacher mdp»
Voir liste classe	Bouton « v »	Clic	Afficher la liste des classes
Voir/cacher mdp	Bouton «cacher mdp»	Clic	Rendre le mot de passe invisible (ou visible)
Valider l'inscription	Bouton valider	Clic	Service inscriptionEleve(eleve, code) Si Inscrire OK alors Retour sur la page d'accueil pour l'authentification Sinon Gestion de l'erreur (envoi d'un mail) Retour sur la page d'accueil pour recommencer l'inscription

- Authentification des élèves (Figure 2)

Intention	Contrôle	Action	Réponse
Initialisation			Page d'accueil - Affichage des champs à modifier de l'authentification de l'élève et de l'intervenant et passage de ces champs à modifiable - Activation du bouton Inscription, des boutons CONNEXION élève et intervenant et bouton «cacher mdp»
Voir/Cacher mdp	Bouton «cacher mdp»	Clic	Rendre le mot de passe invisible (ou visible)
Authentifica-tion	Bouton CONNEXION Eleve	Clic	Service authentificationEleve(email, mdp) Si Authentification OK alors Affichage de la page principale Sinon Gestion erreur (affichage mdp/email incorrects) Retour sur la page d'accueil

- Authentification des intervenants (Figure 2)

Intention	Contrôle	Action	Réponse
Initialisation			Page d'accueil - Affichage des champs à modifier de l'authentification de l'élève et de l'intervenant et passage de ces champs à modifiable - Activation du bouton Inscription, des boutons CONNEXION élève et intervenant et bouton «cacher mdp»
Voir/cacher mdp	Bouton «cacher mdp »	Clic	Rendre le mot de passe invisible (ou visible)
Authentifica-tion	Bouton CONNEXION Intervenant	Clic	Service authentificationIntervenant(login, mdp) Si Authentification OK alors Affichage de la page principale Sinon Gestion erreur (affichage mdp/login incorrects) Retour sur la page d'accueil

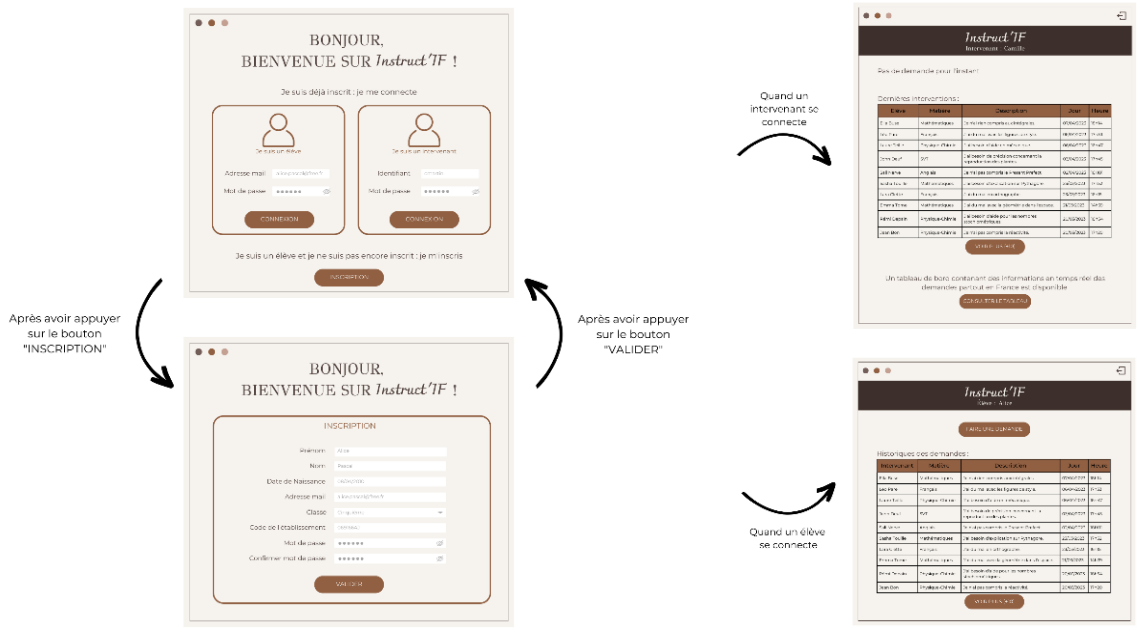

Figure 2 : Inscription et authentification des élèves / Authentification des intervenants

- Création d'une demande (Figure 3)

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'Eleve - Affichage de 10 dernières demandes service genererHistoriqueDemandesEleve(eleve) - Activer bouton Demande et bouton voir + - Service consulterDemandeEnCoursEleve(eleve) - Si il y a une demande en cours alors Affichage de la description Si etatVisio ==true Affichage de l'image de visio pour l'élève et l'intervenant Sinon Afficher pas de demande en cours

Créer demande	Bouton Demande	Clic	Affichage de la page Création Demande - Affichage des champs à modifier pour créer une demande - Passage des champs à modifiables - Service genererListeMatieres() - Activer bouton valider
Valider demande	Bouton valider	Clic	Service creerDemande(matiere, description, eleve) Si Création OK alors Affichage de la page principale avec une fenêtre pop-up « La demande a réussi » Activation Bouton X Sinon Affichage de la page principale avec une fenêtre pop-up « Echec de la demande » Activation Bouton X
Fermer fenêtre pop-up	Bouton X	Clic	Retour sur la page principale
Se déconnecter	Bouton Quitter	Clic	Retour sur la page d'accueil

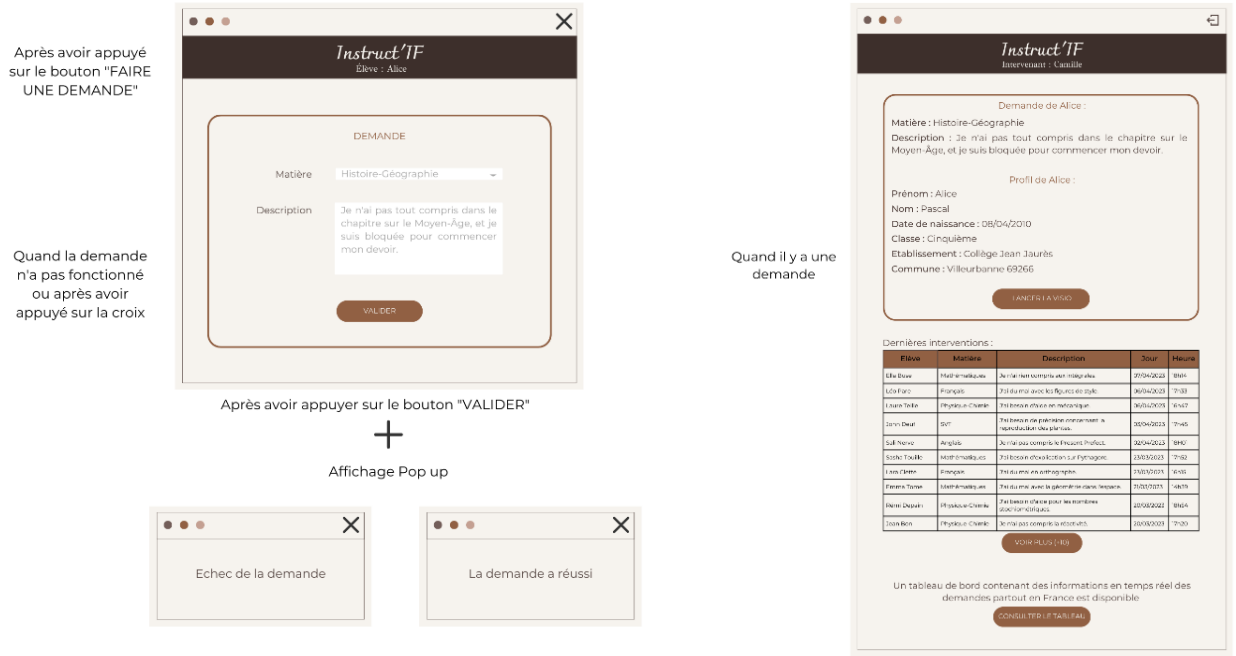

Figure 3 : Création d'une demande par un élève/Consultation de la demande par l'intervenant

- Gestion des informations d'une demande (Figure 3)

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'Intervenant - Affichage de 10 dernières demandes genererHistoriqueDemandesIntervenant(intervenant) - Activer bouton Consulter le tableau et bouton voir + - Service ConsulterDemandeEnCoursIntervenant(intervenant) - Activation Bouton Quitter - Si il y a une demande alors Affichage des caractéristiques de la demande Activer Bouton Lancer Visio Sinon Afficher « pas de demande en cours »
Se déconnecter	Bouton Quitter	Clic	Retour sur la page d'accueil

- Lancement de la visio (côté Intervenant) (Figure 3)

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'intervenant - Affichage de 10 dernières demandes genererHistoriqueDemandesIntervenant(intervenant) - Activer bouton Consulter le tableau et bouton voir + - Service ConsulterDemandeEnCoursIntervenant(intervenant) - Si il y a une demande alors Affichage des caractéristiques de la demande Activer Bouton Lancer Visio Sinon Afficher « pas de demande en cours »
Lancer la visio	Bouton Lancer Visio	Clic	Service lancerVisioIntervenant(demande)

- Arrêt de la visio

Intention	Contrôle	Action	Réponse
Quitter la visio	Bouton raccrocher	Clic	Si mode Eleve alors Service arreterVisio(demande) met à jour le statut de l'intervenant, l'état de la visio et classe la demande comme terminée Affichage de la fenêtre d'évaluation Si mode Intervenant Si etatVisio == false Affichage de la fenêtre principale

- Evaluation de la visio (Figure 4)

Intention	Contrôle	Action	Réponse
Initialisation			Fenêtre d'évaluation - Activation des boutons smileys
Evaluer la demande	Bouton Smiley correspondant à une note	Clic	Service evaluerDemande(demande, note) met à jour la note de la demande Affichage de la fenêtre principale

Figure 4 : Evaluation d'une demande

- Consultation de l'historique des demandes d'un élève

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'élève - Affichage de 10 dernières demandes service genererHistoriqueDemandesEleve(eleve) - Activer bouton Demande et bouton voir + - Service consulterDemandeEnCoursEleve(eleve) - Si il y a une demande en cours alors Affichage de la description Si etatVisio ==true Affichage de l'image de visio pour l'élève et l'intervenant Sinon Afficher pas de demande en cours
Consulter l'historique	Bouton voir +	Clic	Affichage de la page avec les demandes Service genererHistoriqueDemandesEleve(eleve) Activation Bouton Revenir à mon profil
Revenir au profil	Bouton Revenir à mon profil	Clic	Retour page principale
Se déconnecter	Bouton Quitter	Clic	Retour page d'accueil

- Consultation de l'historique des demandes d'un élève (Figure 5)

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'intervenant - Affichage de 10 dernières demandes genererHistoriqueDemandeIntervenant(intervenant) - Activer bouton Consulter le tableau et bouton voir + - Service ConsulterDemandeEnCoursIntervenant(intervenant) - Si il y a une demande alors Affichage des caractéristiques de la demande Activer Bouton Lancer Visio Sinon Afficher « pas de demande en cours »
Consulter l'historique	Bouton voir +	Clic	Affichage de la page avec les interventions Service genererHistoriqueDemandeIntervenant(intervenant) Activation Bouton revenir à mon profil

Revenir à mon profil Se déconnecter	Bouton Revenir à mon profil Bouton quitter	Clic Clic	Affichage de la page principale Affichage de la page d'accueil

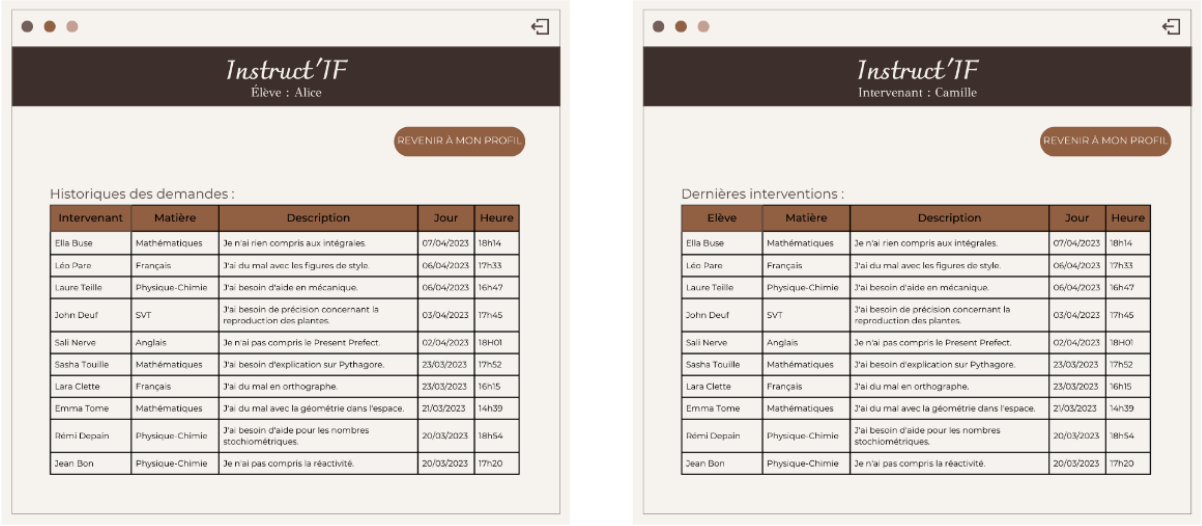

Figure 5 : Historique des demandes Eleve/Intervenant

- Consultation des statistiques (Figure 6)

Intention	Contrôle	Action	Réponse
Initialisation			Page principale de l'intervenant - Affichage de 10 dernières demandes genererHistoriqueDemandeIntervenant(intervenant) - Activer bouton Consulter le tableau et bouton voir + - Service ConsulterDemandeEnCoursIntervenant(intervenant) - Si il y a une demande alors Affichage des caractéristiques de la demande Activer Bouton Lancer Visio Sinon Afficher « pas de demande en cours »
Consulter les statistiques	Bouton Consulter le tableau	Clic	Affichage de la fenêtre des statistiques Activation bouton Revenir à mon profil Service genererNombreDemandesTotales(), genererStatistiqueDemandeParClasse(), genererStatistiqueDemandeParMatiere(), genererStatistiqueDemandeParCodeIPS() et genererStatistiqueDemandeParDepartement()
Revenir à mon profil	Bouton revenir à mon profil	Clic	Affichage de la page principale
Se déconnecter	Bouton Quitter	Clic	Affichage de la page d'accueil

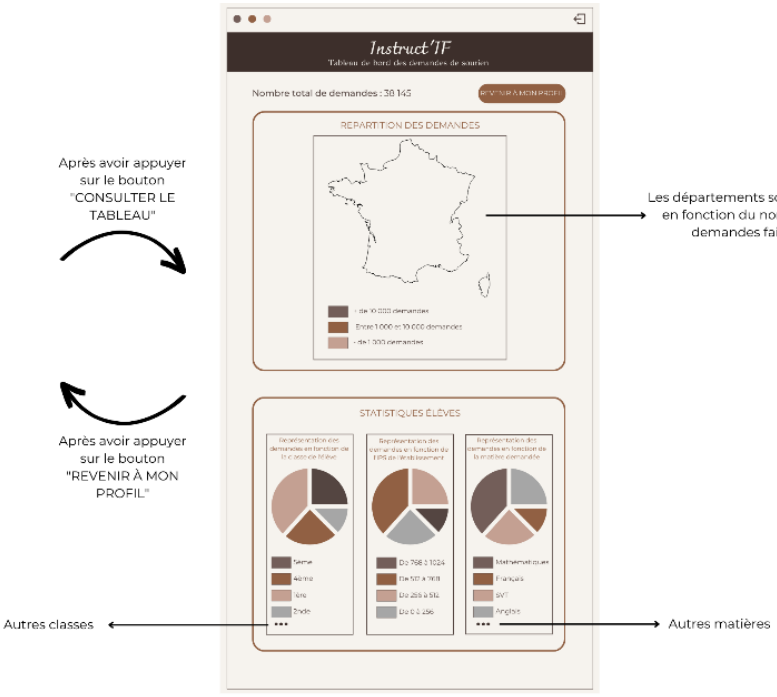

Figure 6 : Consultation du tableau de bord

Conclusion

INSTRUCT'IF est une application pour aider les élèves à faire leur devoir en dehors de cours. Il a fallu concevoir et analyser le problème. Après cette phase, nous avons développé les services permettant d'implémenter l'IHM que nous avons conçue.

Annexe 1

Annexe 1 : Description de l'IHM