AUTIER Agathe, ROUQUETTE Joseph, B3330

PREDICTIF

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

PREDICT'IF est une application qui permet à des clients de prendre des rendez-vous avec des médiums incarnés par des employés de l'entreprise. Cette application peut être exécutée par deux types d'utilisateur : les employés et les clients.

>> Du point de vue client :

Si l'utilisateur est un client, alors il peut se créer un compte ou bien s'identifier. Pour créer un compte, il doit renseigner son nom, son prénom, son mail qui lui sert d'identifiant, son numéro de téléphone, sa date de naissance selon le format yyyy-mm-dd, son mot de passe, son genre (seulement 'H' pour homme et 'F' pour femme) et son adresse postale. Une fois que ces informations sont renseignées et à condition qu'un autre compte ayant la même adresse mail n'existe pas déjà, le compte du client est enregistré dans la base de données. Pour pouvoir s'authentifier à l'application, l'utilisateur doit avoir déjà créé un compte, dans le cas contraire, la connexion serait refusée. Si c'est bien le cas, alors il suffit de renseigner son mail et son mot de passe. Il peut alors avoir accès à son profil astral qui comprend son signe du zodiaque, son animal totem, sa couleur fétiche et son signe chinois. Il peut aussi consulter l'historique de ses consultations. Quand il veut demander une nouvelle consultation, il a d'abord accès à la liste de tous les médiums de l'entreprise avec leurs informations, c'est-à-dire une petite présentation et leur genre (+ son support pour si c'est un spirite, sa formation et sa promotion si c'est astrologue). Puis, il doit entrer l'id du médium avec qui il souhaite s'entretenir pour valider le rendez-vous. Un message est alors envoyé à l'employé qui est sélectionné pour incarner ce médium. Le choix de l'employé est en accord avec le genre du médium demandé et est fonction du nombre de consultations qu'il a déjà eu : c'est l'employé en ligne qui a le moins de rendez-vous à son actif qui est choisi. Si aucun employé n'est disponible pour incarner le médium, la demande de consultation est immédiatement refusée. Le client peut quitter l'application quand il veut.

>> Du point de vue Employé :

Les employés de l'entreprise sont déjà présents dans la base de données de l'application. Il n'est donc pas possible de créer un compte avec un employé. Si l'utilisateur est un employé de l'entreprise (donc déjà enregistré dans la base de données), il peut s'authentifier grâce à son mail et à son mot de passe de la même manière que pour le client. Il peut voir tous les clients présents dans la base de données et obtenir le détails de leurs informations (profil astral, historique…) en rentrant leur id. Il a accès au TOP 5 des médiums qui génèrent le plus de consultations, à la répartition des consultations par employé et à la répartition des consultations par médiums. Il peut lancer une consultation à condition qu'elle lui ait été attribuée lorsque qu'un client à demander à joindre un médium et que cette consultation soit en cours (une consultation en cours ne comporte aucun commentaire). Une fois la consultation lancée, il peut obtenir des prédictions automatiques pour son client dans les catégories amour, santé et travail, (en attribuant une note entre 1 et 4 dans chaque catégorie). Lorsqu'il met fin à la consultation, il peut lui mettre un commentaire. Il peut consulter l'historique des consultations qu'il a déjà réalisé avec leur détail, la consultation en cours apparait également dans l'historique. Il peut quitter l'application quand il veut.

II/MODELE DU DOMAINE

Predict'IF est une application de services permettant de mettre en relation des clients avec des médiums incarnés par des employés.

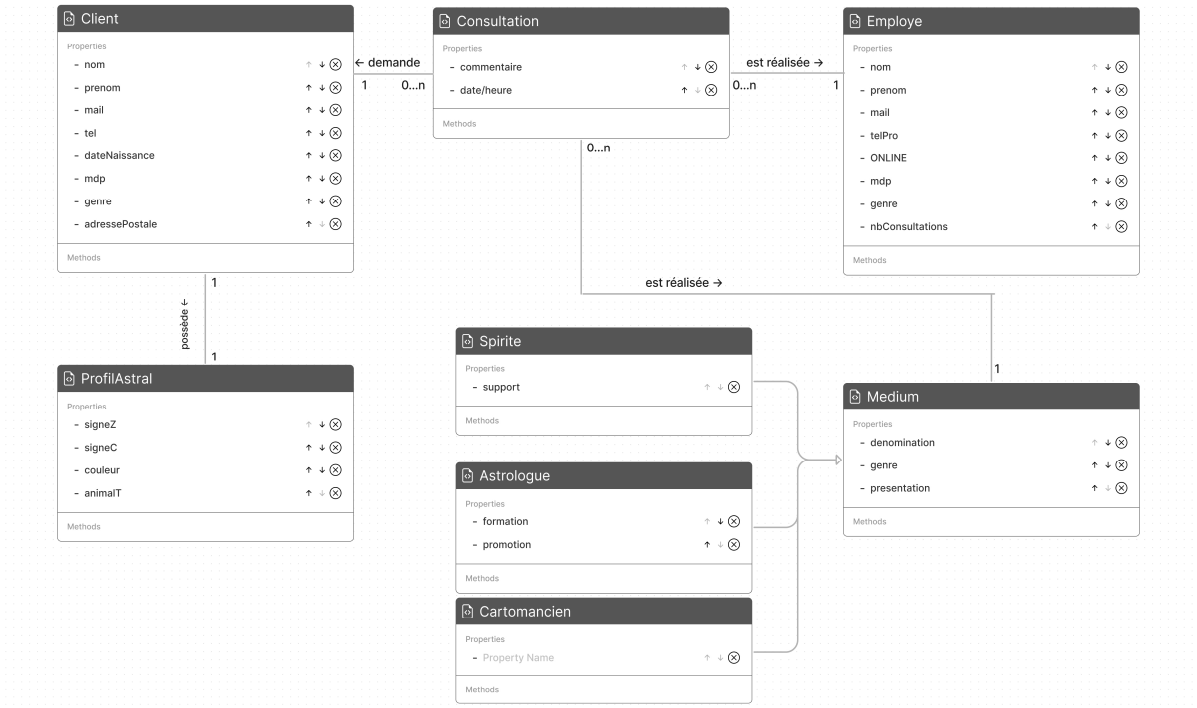


Diagramme de classe de l'application Predict'IF

La classe Client représente chaque client de PREDICT'IF. Un Client a un nom, un prénom, une adresse mail pour l'identifier, un numéro de téléphone, une date de naissance, un mot de passe pour accéder à son compte, un genre (H ou F) et une adresse postale. Chaque Client possède un seul profil astral créé à partir de sa date de naissance. Un Client peut avoir assister à plusieurs consultations comme à aucune. La classe Employé représente chaque employé de PRÉDICT'IF. Un Employé a un nom, un prénom, une adresse mail pour l'identifier, un numéro de téléphone, un mot de passe pour accéder à son compte, un genre (H ou F), un nombre de consultation à son actif et un indicateur de statut pour savoir s'il est en ligne. Un Employé peut réaliser plusieurs consultations comme aucune.

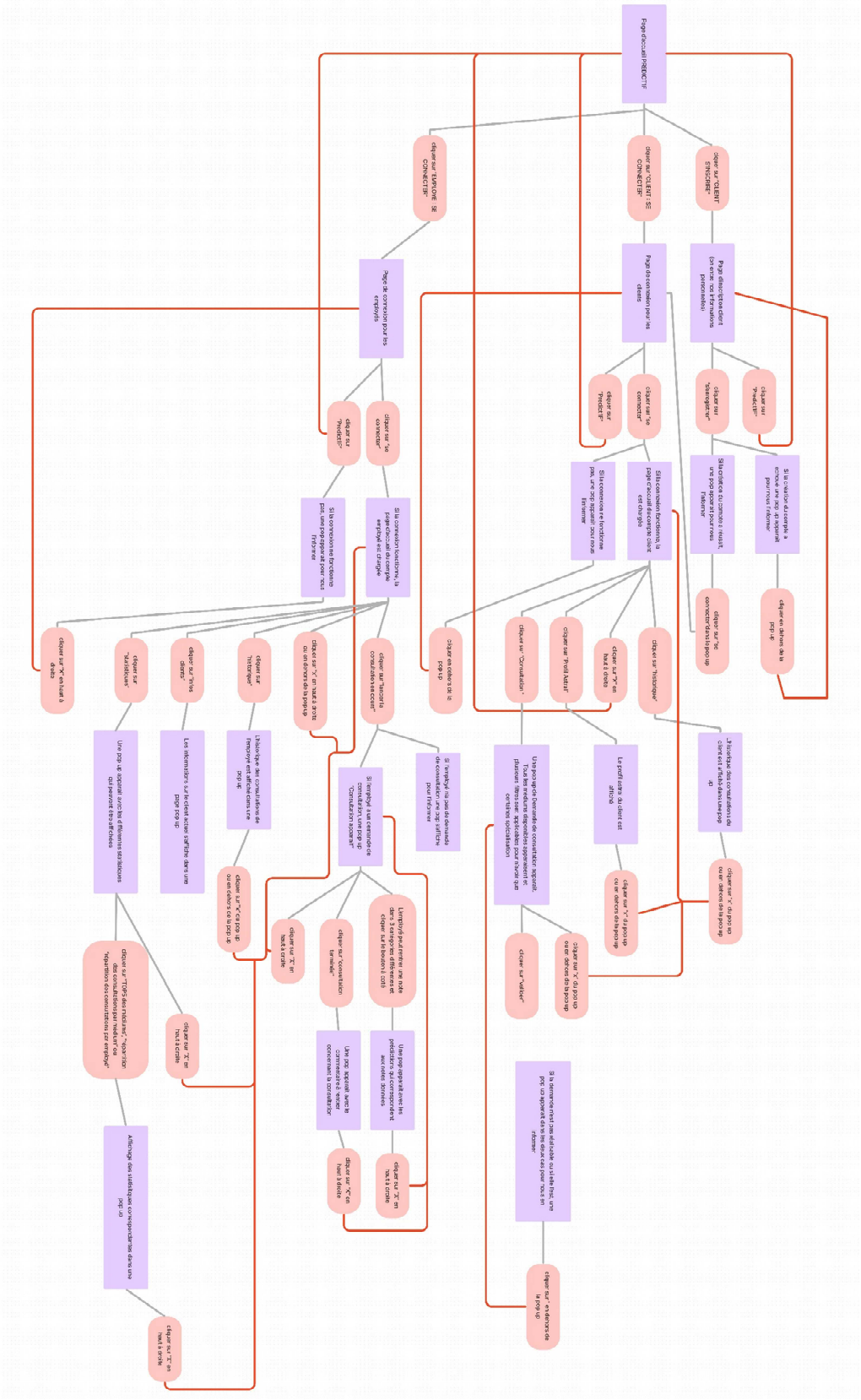
La classe médium représente les médiums qui sont incarnés par les employés lors des consultations. Un médium est caractérisé par une dénomination, un genre et une petite présentation. Un médium peut être incarné par plusieurs consultations comme aucun. La classe Spirite représente les Médiums spécialisés en spiritisme. Un Spirite est caractérisé par un support qu'il utilise comme une Boule de Cristal. La carte Cartomancien représente les Médiums spécialisés en cartomancie. Un Cartomancien ne possède pas d'attributs supplémentaires. La classe Astrologue représente les Médiums spécialisés en astrologie. Un Astrologue est caractérisé par une formation et une promotion. Un Astrologue, un cartomancien et un spirite sont des types de Médium i.e. leurs classes respectives héritent de la classe médium.

La classe Consultation représente chaque consultation réalisée dans le cadre de PREDICT'IF. Une consultation est caractérisée par un commentaire (écrit par l'employé en charge de cette consultation à la fin de celle-ci) et une date. Une consultation peut être réalisé par un seul médium et donc un seul employé. Elle est demandée par un seul client.

La classe ProfilAstral représente le profile astral de chaque Client qui est créé dès son inscription. Un Profil Astral contient un signe du zodiaque, un signe chinois, une couleur fétiche et un animal totem. Chaque ProfilAstral n'est associé qu'à un seul client.

III/IHM

>> Diagramme d'enchaînement de fenêtres, disponible également ici.



>>IHM

Figure 1-page d'accueil

Figure-2-inscription client

Figure 3- page connexion client

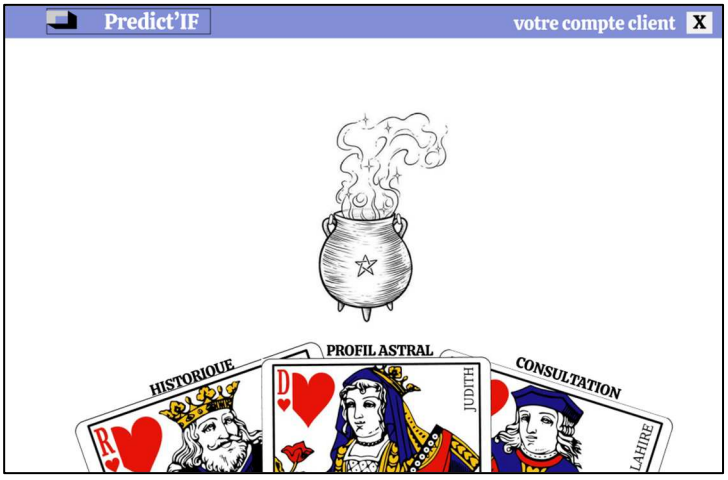


Figure 4- page compte client

Pop-up 1

Pop-up 2

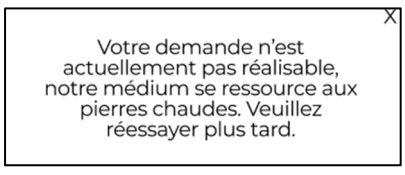
Pop-up 3

Pop-up 4 erreur de création de compte

Pop-up 5 erreur d'authentification

Pop-up 6 compte créé

Pop-up 7



Pop-up 8

Figure 5- page connexion employé

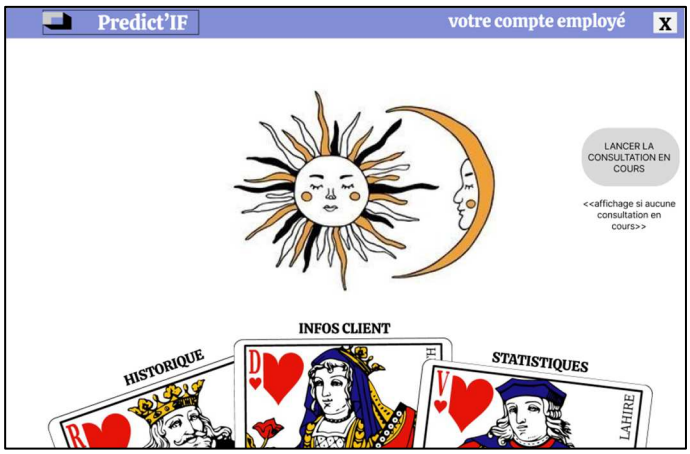
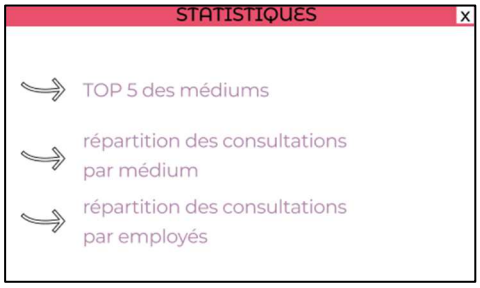


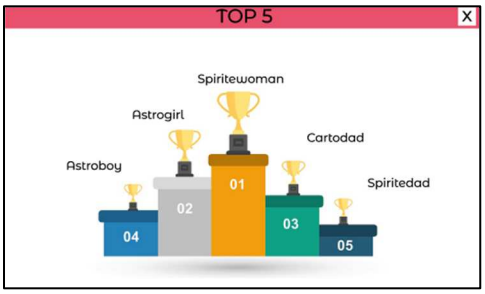
Figure 6 - menu employé

Pop-up 9

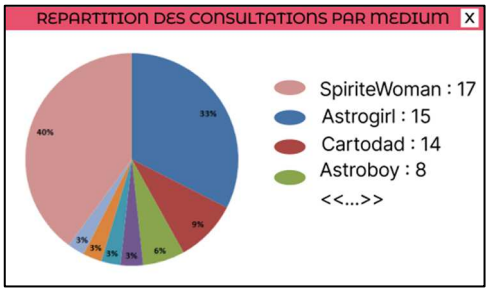
Pop-up 10



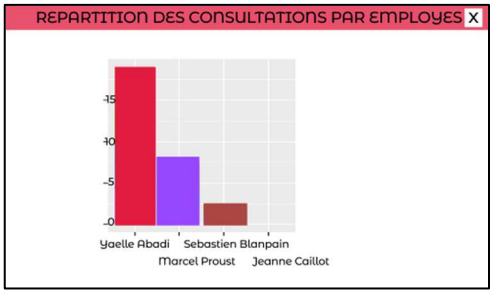
Pop-up 11



Pop-up 12



Pop-up 13



Pop-up 14

Pop-up 15

Pop-up 16 - obtention de la prédiction

Pop-up 17- nouveau commentaire

Pop-up 18-aucune consultation en cours

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 :
https://www.figma.com

III/CONCEPTION DES IHMS ET DES SERVICES

>> Tableau ICARS

Page d'accueil

Intention	Contrôle	Action	Réponse et Services
Initialisation de la base de données			Appel aux services "InitialiserMediums()" et "InitialiserEmployés()"pour initialiser la base de données
Initialisation de l'affichage			Affichage du top 5 médium : la dénomination des 5 médiums les plus populaires grâce à l'appel au service "obtenirStat1()" + affichage de l'image d'un génie, et des 3 boutons cités ci-dessous
Connexion Employé	Bouton "Employé: Se connecter"	Clic	Redirection vers la page « Connexion Employé » avec une transition style fumée
Connexion Client	Bouton "Client: Se connecter"	Clic	Redirection vers la page « Connexion Client» avec une transition style fumée
Inscription Client	Bouton "Client : S'inscrire"	Clic	Redirection vers la page « Inscription Client» avec une transition style fumée
Consulter le génie	Image de lampe à génie dans le coin en bas à droite	Passage par dessus avec la souris	Affichage d'une image animée de génie prononçant un tweet d'une célébrité ventant les mérites de ce site, si possible les yeux du génie doivent alors suivre la souris.

Page Inscription Client :

Intention	Contrôle	Action	Réponse et Services
S'inscrire	Formulaire avec champs de saisie et bouton de validation	Clic	Mise à blanc des champs de saisie, ceux si sont modifiables
Valider inscription	Bouton "S'enregistrer"	Clic	Appel au service "inscriptionClient(client)" avec vérification des champs de saisie non vide et dans un format adéquat (pour date yyyy-mm-dd et pour genre 'H' ou 'F') Si inscription OK → envoie d'un mail de confirmation au client et affichage d'un pop-up avec un message de confirmation proposant une redirection vers la page "Connexion Client" Sinon → envoie d'un mail d'infirmation au client et affichage d'un pop-up d'erreur
Revenir à l'accueil	Bouton "Predict'IF"	Clic	Ferme la fenêtre actuelle et revient à la page d'accueil de l'application

Page Connexion Client :

Intention	Contrôle	Action	Réponse et Services
Se connecter	Formulaire avec champs de saisi	Clic	Mise à blanc des champs de saisie, ceux si sont modifiables
Valider la connexion	Bouton "Se connecter"		Appel au service "authentifierClient(Client client)" avec vérification des champs de saisie non vide Si authentification OK (càd le service renvoie un client non null) → redirection vers la page menu client. Sinon Affichage d'un pop-up avertissant l'utilisateur de l'échec
Revenir à l'accueil	Bouton "Predict'IF"	Clic	Ferme la fenêtre actuelle et revient à la page d'acceuil de l'application

Page Compte Client :

Intention	Contrôle	Action	Réponse et Services
Consulter son historique	Carte "Historique"	Clic	Remplacement de l'image centrale (ou de la fenêtre centrale actuelle) par une fenêtre contenant une liste des des dernières consultations triées par date, avec leurs détails. La transition se fera de manière fluide comme si on dégainait une carte de sa main.
Quitter l'historique	Bouton "X"	Clic	Ferme la page Historique
Lire son profil astral	Carte "Profil Astral"	Clic	Remplacement de l'image centrale (ou de la fenêtre centrale actuelle) par une fenêtre contenant les 4 thèmes du profil astral grâce à l'appel à la méthode "getProfilAstral()". Ces thèmes seront illustrés par une image caractéristique en plus d'être écrits. La transition se fera de manière fluide comme si on dégainait une carte de sa main.
Quitter le profil astral	Bouton "X"	Clic	Ferme la page Profil Astral
Demande de consultation	Carte "Consultation"	Clic	Remplacement de l'image centrale (ou de la fenêtre centrale actuelle) par une fenêtre contenant une liste des médiums. La transition se fera de manière fluide comme si on dégainait une carte de sa main. Cette fenêtre possédera aussi des cases à cocher permettant de filtrer les médiums affichés par type, et un bouton pour valider la demande. L'ensemble des médiums est obtenu en faisant appel au service « listerTousMedium() ».
Filtrer les médiums lors du choix pour une demande	Cases à cocher apparues dans la fenêtre « Demande de consultation »	Clic	Ajouter/Enlever les médiums du type coché à la liste de médiums affichée
Choisir un médium	Puce associée à chaque médium apparue dans la partie « Demande de consultation »	Clic	Sélection d'un médium dans l'objectif de le choisir pour demander une consultation
Valider la demande de consultation	Bouton « Valider » apparu dans la partie « Demande de consultation »	Clic	Appel au service « demanderConsultation(Client client, String idMedium) » en vérificatiant qu'un médium a bien été sélectionné Si succès de la demande (càd le service renvoie une consultation non null) apparition d'un pop-up disant que la consultation a été prise en compte et envoi d'une notification à l'employé qui va s'en charger Sinon appartition d'un pop up indiquant que le médium n'est pas disponible
Quitter la demande de consultation	Bouton "X"	Clic	Fermeture de la page demander consultation
Quitter le compte	Bouton "X"	Clic	Ferme la fenêtre actuelle et revient à la page d'accueil de l'application

Page Connexion Employé

Intention	Contrôle	Action	Réponse et Services
Se connecter	Formulaire avec champs de saisi	Clic	Mise à blanc des champs de saisie, ceux si sont modifiables
Valider la connexion	Bouton "Se connecter"	Clic	Appel au service "authentifierEmployé(Employe employe)" avec vérification des champs de saisie non vide Si authentification OK (càd le service renvoie un employé non null) → redirection vers la fenêtre menu employé, avec transmission de l'employé obtenu Sinon Affichage d'un pop-up avertissant l'utilisateur de l'échec
Revenir à l'accueil	Bouton "Predict'IF"	Clic	Ferme la fenêtre actuelle et revient à la fenêtre d'accueil de l'application

Page Compte Employé

Intention	Contrôle	Action	Réponse et Services
Consulter l'historique	Carte "Historique"	Clic	Appel au service "obtenirHistoriqueE(Employe employe)", ouverture d'une fenêtre avec le string résultant de ce service affiché
Quitter l'historique	Bouton "X"	Clic	Fermeture de la page demander historique
Consulter les infos Client	Carte "Infos Client"	Clic	Affichage d'une fenêtre avec la liste de tous les clients avec l'appel au service "listerTousClients()"
Quitter les infos clients	Bouton "X"	Clic	Fermeture de la page infos clients
Consulter les statistiques	Carte "Statistiques"	Clic	Ouverture du la page Statistiques (voir le détails ci dessous)
Quitter consulter Statistiques	Bouton "X"	Clic	Fermeture de la page statistiques
Lancer la consultation en cours	Bouton "Lancer la consultation en cours"	Clic	Appel au service "trouverConsultationCourante(Employe employe)" Ouverture Page Consultation -Affichage des questions par thème -Affichage bouton Finie
Quitter Lancer la consultation en cours	Bouton "X"	Clic	Fermeture de la fenêtreLancer la consultation en cours
Quitter le compte	Bouton "X"	Clic	Ferme la fenêtre actuelle et revient à la page d'accueil de l'application

Page Statistiques

Intention	Contrôle	Action	Réponse et Services
Consulter le statistique le top 5 médiums	texte "Top 5 des médiums"	Clic	Appel au service "obtenirStat1()" pour afficher une fenêtre contenant le String résultat
Quitter Top 5	Bouton "X"	Clic	Fermeture de la fenêtre Top 5
Consulter la statistique la répartition des consultations par médiums	texte "Répartition des consultations par médium"	Clic	Appel au service "obtenirStat2()" pour afficher une fenêtrecontenant le String résultat + affichage d'un diagramme camembert associé à ces données
Quitter la répartition des consultations par médiums	Bouton "X"	Clic	Fermeture de la fenêtre répartition des consultations par médiums
Consulter la statistique la répartition des consultations par employé	texte "Répartition des consultations par employés"	Clic	Appel au service "obtenirStat3()" pour afficher une fenêtrecontenant le String résultat
Quitter la répartition des consultations par employés	Bouton "X"	Clic	Fermeture de la fenêtre répartition des consultations par employés

Page infos clients

Intention	Contrôle	Action	Réponse et Services
Consulter les informations détaillées d'un client	La puce associée à un client	Clic	Appel du service "trouverClientEnCours(Long idEmploye)" pour trouver le client associé à la consultation en cours de l'employé. On affiche ensuite le nom, prenom, profil astral et l'historique du client renvoyé
Quitter la répartition les informations détaillées	Bouton "X"	Clic	Fermeture de la fenêtre informations détaillées du client

Page lancer Consultation en Cours

Intention	Contrôle	Action	Réponse et Services
Saisir des notes dans les 3 catégories	Formulaire avec champs de saisie	Clic	Mise à blanc des champs de saisie, ceux si sont modifiables
Obtenir une prédiction automatique	Bouton rond	Clic	Appel au service "obtenirPrediction(Client client, String nA, String nS, String nT)" et affichage du résultat du service dans un pop-up
Terminer la consultation	Bouton "Consultation terminée ?"	Clic	Affichage d'un pop-up "Commenter la consultation"
Commenter la consultation	Formulaire avec champs de saisie dans le pop-up "Commenter Consultation"	Clic	Mise à blanc des champs de saisie, ceux si sont modifiables Appel au service "commenterConsultation(String commentaire, Consultation c)"
Quitter la répartition Lancer la consultation en cours	Bouton "X"	Clic	Fermeture de la fenêtre Lancer la consultation en cours

>> Services
Chaque service possède également une gestion d'erreur avec un try-catch permettant de roll-back des transactions ouvertes par exemple.

SERVICE public void initialiserEmployes()

>> description : permet de créer dans la base de données 15 employés différents
>> algo :

1. Création de 15 employés avec le constructeur de la classe Employé
2. Création d'une DAO Employé et du contexte de persistance puis ouverture de la transaction
3. Appel à la fonction create de la DAO Employé pour persister les 15 employés dans la BD
4. Validation de la transaction et fermeture du contexte de persistance

SERVICE public Employe trouverEmployeParId(Long idToFind)

>> description : permet de trouver un Employé dans la base de données à partir de son ID
>> algo :

1. Création d'une DAO Employé et du contexte de persistance
2. Appel à la méthode find de la DAO pour renvoyer l'employé correspondant à l'id
3. Fermeture du contexte de persistance, on return l'employé trouvé (null si aucun ne correspond)

SERVICE public String listerTousEmployes()

>> description : permet de lister tous les employés présents dans la base de données par ordre alphabétique des prénoms
>> algo :

1. Création d'une DAO Employé et du contexte de persistance
2. Appel à la méthode liste de la DAO, elle effectue une requête JPQL qui liste tous les employés dans l'ordre alphabétique
3. Parcours de la liste des employés et stockage des infos des employés dans un string result
4. Fermeture du contexte de persistance et on return result

SERVICE public Employe authentifierEmploye(String mail, String motdepasse)

>> description : permet grâce à un mail et à un mot de passe d'authentifier l'employé concerné
>> algo :

1. Création d'une DAO Employé et du contexte de persistance

2. Appel à la méthode findThanksToMail() de la DAO avec le mail en paramètre. Celle-ci renvoie l'employé associé au mail s'il existe et null sinon
3. Fermeture du contexte de persistance
4. Si le mot de passe en paramètre est différent du mot de passe associé à l'employé trouvé avec la DAO, on return null, sinon on retourne l'employé trouvé avec la DAO

SERVICE public String consulterHistoriqueE(Employe employe)

>> description : permet à un employé de voir toutes ses consultations à un instant t, de la plus récente à la plus ancienne

>> algo :

1. Création d'une DAO Consultation et du contexte de persistance
2. Appel de la méthode listeConsultationEmploye() de la DAO, elle retourne une liste des consultations triées par date
3. Parcours de la liste et on remplit un string result de retour avec les infos des consultations
4. Fermeture du contexte de persistance et on return result

SERVICE public String consulterHistoriqueCpourE(Client client)

>> description : permet à un employé de consulter l'historique d'un client avec to

>> algo : même principe que consulterHistoriqueE(), sauf qu'on appelle la méthode listeConsultationClient() au lieu de listeConsultationEmploye()

SERVICE public void commenterConsultation(String com, Consultation c)

>> description : permet à un employé, une fois la consultation terminée de la commenter

>> algo

1. Création d'une DAO Consultation et Employe puis du contexte de persistance
2. Si la consultation passée en paramètre n'est pas null, ouverture de la transaction
3. On set le nouveau commentaire de la consultation puis appel de la méthode maj de la DAO Consultation, pour mettre à jour le commentaire dans la base de données
4. On set le statut online de l'employé à true car sa consultation est terminée après qu'il ait mis un commentaire puis appel de la méthode maj de la DAO Employe, pour mettre à jour le statut dans la base de données
5. Validation de la transaction et fermeture du contexte de persistance

SERVICE public Consultation trouverConsultationCourante(Employe employe)

>> description : permet pour un employé donnée de trouver la consultation en cours

>> algo

1. Création d'une DAO Consultation puis du contexte de persistance, déclaration d'une consultation c
2. Appel à la méthode de la DAO findConsultationRunning() avec l'id de l'employé en paramètre, cette méthode retourne dans c la dernière consultation de l'employé ou null si il n'en a pas
3. Si c est non null et que son commentaire est null
   Alors un message de lancement est envoyé au client associé à la consultation
   Sinon si c est non null et que son commentaire est non null
   c = null, car une consultation commentée est considérée comme terminée
4. Fermeture du contexte de persistance et on return c

SERVICE public String obtenirPrediction(String nA, String nS, String nT, Client client)

>> description : permet pour un client donné et pour des notes dans 3 catégories, d'obtenir une prédiction pour chaque catégorie

>> algo

1. Création d'une instance d'astroApi, astroApi
2. Récupération des notes données dans les 3 catégories et conversion en int
3. On return un string avec les prédictions amour travail et santé obtenues grâce à la méthode getPredictions() de astroApi

SERVICE public String trouverClientEnCours(Long idEmploye)

>> description : permet de trouver le client associé à la consultation en cours d'un employé
>> algo

1. Création d'une DAO Consultation puis du contexte de persistance
2. Appel à la méthode findConsultationRunning() de la DAO, celle-ci renvoie une consultation
3. Si la consultation retournée n'est pas null, on récupère le client associé
4. Fermeture du contexte de persistance et on return le client associé ou null si il n'a pas été trouvé

SERVICE public String obtenirStat1()

>> description : permet d'accéder au TOP5 des médiums
>>algo

1. Création d'une DAO Consultation puis du contexte de persistance
2. On initialise le String result à null
3. Création d'une liste de Médiums grâce la méthode top5 de la DAO qui retourne une liste des 5 Médiums avec le plus de consultations
4. Parcours de cette liste, la dénomination des médiums et leur rang est placée dans le String result
5. Fermeture du contexte de persistance et on return le result

SERVICE public String obtenirStat2()

>> description : permet d'accéder à la répartition des consultations par médiums
>> algo

1. Création d'une DAO Consultation puis du contexte de persistance
2. On initialise le String result à null
3. On crée une liste d'objets grâce à la DAO qui contient à la fois les médiums et le nombre de consultations de chacun.
4. On parcourt cette liste et on met dans le string result, la dénomination de chaque médium et son nombre de consultation
5. Fermeture du contexte de persistance et on return le result

SERVICE public String obtenirStat3()

>> description : permet d'accéder à la répartition des consultations par employés
>> algo : même principe que obtenirStat2() mais avec des employés à la place des médiums

SERVICE public void inscriptionClient(Client C1)

>> description : permet de persister un client donné
>> algo

1. Création d'une instance astroNetAPi et d'une instance ProfilAstral grâce au prénom et à la date de naissance du client passé en paramètre
2. On set le Profil Astral du Client C avec le profil Astral nouvellement créé
3. Création d'une DAO Client et ProfilAstral puis du contexte de persistance
4. Ouverture de la transaction
5. Appel des méthodes create() respectives des 2 DAO pour persister le client et le profil astral dans la BD
6. Validation de la transaction puis envoi d'un mail de confirmation au client
7. Fermeture du contexte de persistance

SERVICE public Client trouverClientParId(Long idToFind)

>> description : permet de trouver le client associé à un Id
>> algo même principe que trouverEmployeParId()

SERVICE public String listerTousClients()

>> description : permet de lister tous les clients présents dans la base de données, triés par ordre alphabétique
>> algo même principe que listerTousEmployes()

SERVICE public Client authentifierClient(String mail, String motdepasse)

>> description : permet d'authentifier un client donné grâce à son mail et à son mot de passe
>> algo même principe qu'authentifierEmploye()

SERVICE public Consultation demanderConsultation(Client client, String id)

>> description : permet à client de demander une consultation avec un médium en particulier
le client demandant et l'id du médium associé, la consultation, renvoie null si aucun employé adéquat n'est disponible pour incarner le médium oui si le médium n'existe pas
>> algo

1. Création d'une DAO Consultation, Employe, Medium et du contexte de persistance
2. Déclaration d'une consultation c, initialisée à null
3. Appel à la méthode find de la DAO de médium pour retrouver le médium choisi par le client
4. Si le medium retrouvé est non null
    a. Grâce au genre du médium, on en déduit l'appellation qu'il faudra utiliser dans le mail (Mr ou Mme)
    b. Appel à la méthode findEmployeAdapte() de la DAO Employé, celle-ci retourne un employé du même genre que le médium, disponible et ayant le moins de consultation parmi ceux respectant les deux premiers critères
    c. si l'employé retourné n'est pas null
        i. Création d'une instance de consultation à partir du client et de la date actuelle
        ii. Ouverture de la transaction
        iii. Set de l'employé et du médium associé à la consultation
        iv. Appel à la méthode create de la DAO Consultation pour la faire persister
        v. Incrémentation du nombre de consultation de l'employé et set son booléen online à false
        vi. Appel à la méthode maj de la DAO Employe pour faire persister les deux modifications
        vii. Validation de la transaction et envoi d'un message à l'employé concerné
    d. Sinon c = null
5. Sinon c = null
6. Fermeture du contexte de persistance et on return la consultation c

SERVICE public String consulterHistorique(Client client)

>> description : permet à un client de consulter l'historique de ses consultations à un instant, de la plus récente à la plus ancienne, il peut voir la date et le médium associé à chaque consultation
>> algo même principe que consulterHistoriqueE()

SERVICE public void initialiserMediums()

>> description : permet de créer dans la base de données 15 employés
>> algo même principe qu'initialiserEmployes()

SERVICE public Medium trouverMediumParId(Long idToFind)

>> description : permet à partir d'un identifiant de trouver le médium associé
>> algo même principe que trouverEmployeParId()

SERVICE public String listerTousMediums()

>> description : permet de lister tous les médiums de la base de données, triés par id croissants
>> algo même principe que listerTousEmployes