****

**Dossier de Conception**

**~ Prédictif ~**

****

**Milène Daugan et Romain Chikirou mai – juin 2020**

Table des matières

[I Modèle du domaine 3](#__RefHeading___Toc791_342855041)

[1) Modèle conceptuel de données 3](#__RefHeading___Toc2226_576320575)

[2) Navigation du client 4](#__RefHeading___Toc2306_576320575)

[3) Navigation de l’employé 5](#__RefHeading___Toc2308_576320575)

[II Maquettes IHM, ICARS et services 6](#__RefHeading___Toc789_342855041)

[Fenêtre principale connexion 6](#__RefHeading___Toc2236_576320575)

[Fenêtre pop-up inscription 7](#__RefHeading___Toc2238_576320575)

[Fenêtre profil + historique 9](#__RefHeading___Toc2294_576320575)

[Fenêtre choix des médiums 11](#__RefHeading___Toc2296_576320575)

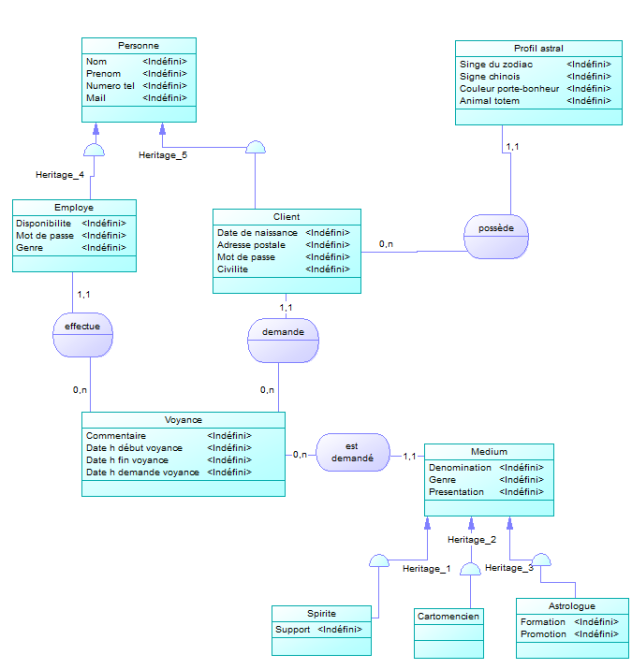
[Fenêtre connexion employé 13](#__RefHeading___Toc2298_576320575)

[Fenêtre Employé 14](#__RefHeading___Toc2300_576320575)

[Fenêtre aide consultation 16](#__RefHeading___Toc2302_576320575)

## I Modèle du domaine

### 1) Modèle conceptuel de données



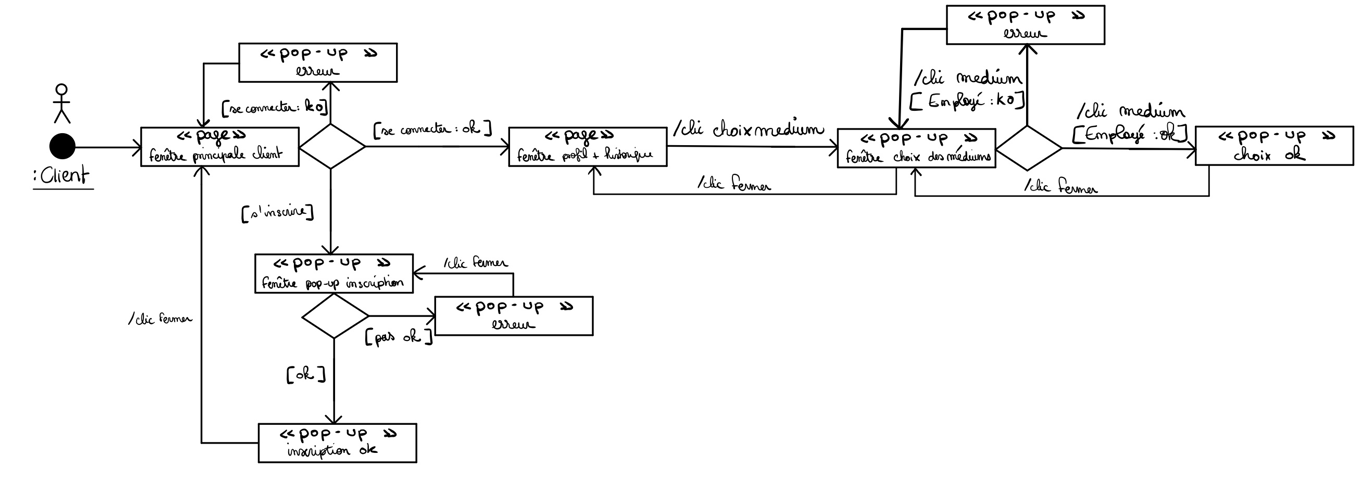
Nous avons deux situations d’héritage dans notre diagramme pour les classes:

- Medium (un spirite, un cartomancien et un astrologue sont des médiums)

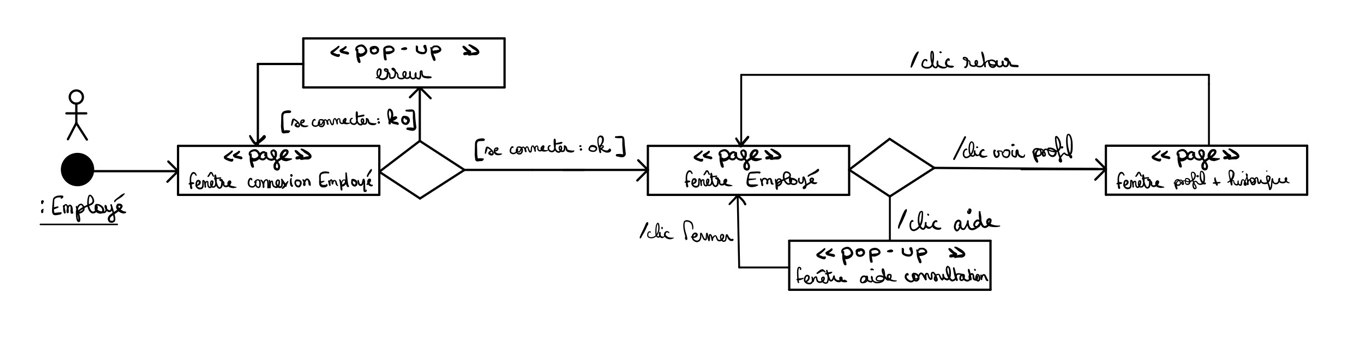
- Personne (un employé et un client sont des personnes)

Les spécifications annoncées ici sont approfondies mais ne pourront pas être toutes implémentées dans la programmation de l’application par contrainte de temps.

### 2) Navigation du client

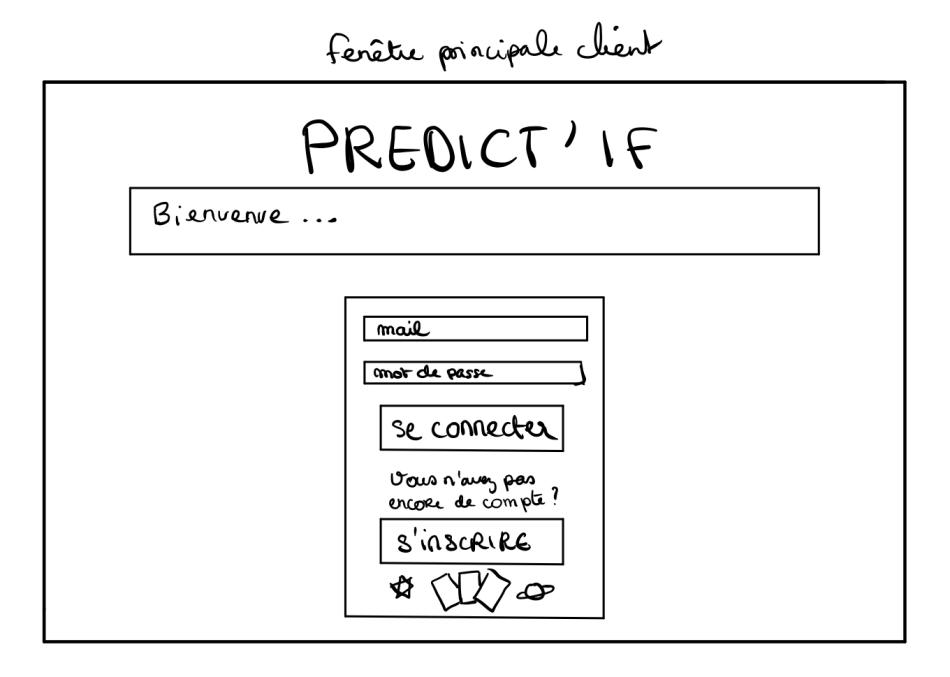


### 3) Navigation de l’employé



## II Maquettes IHM, ICARS et services

#### Fenêtre principale connexion



|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Affichage de la fenêtre |
| Connexion à un compte | Bouton : «Se connecter » | Clic | Service AuthenticateClient(mail, password) :  - Si ok : Connexion et redirection vers la fenêtre profil+ historique  Sinon : Affichage de pop-up erreur |
| Création d’un compte | Bouton « S’inscrire » | Clic | Affichage de la pop-up inscription |

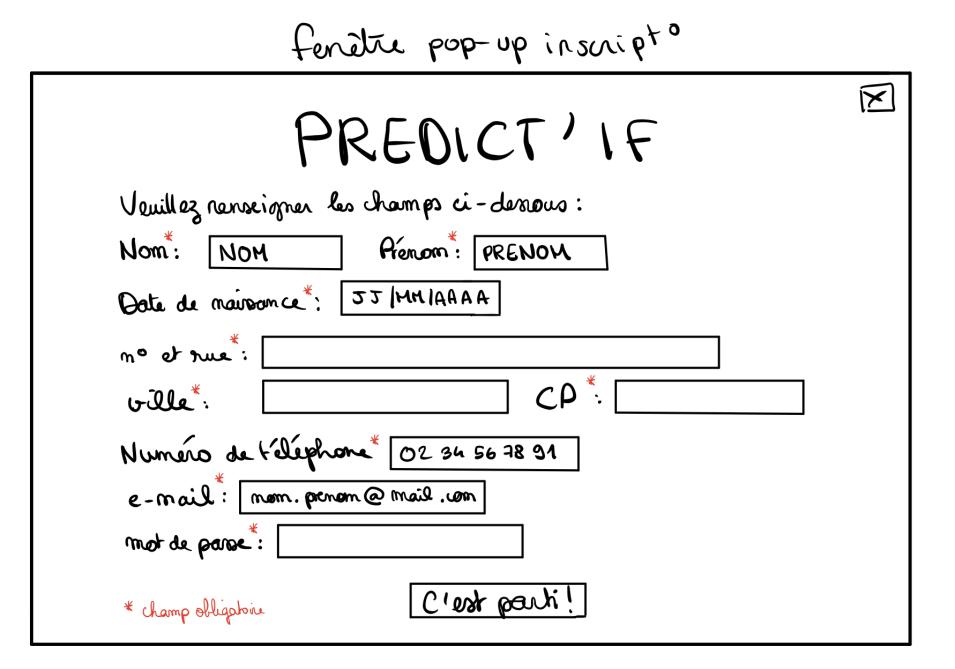
Liste des services :

Client AuthenticateClient(String mail, String password)

Prend le mail et le mot de passe entrés par l’utilisateur en paramètre.

S’ils correspondent à un tuple mail associé à son mot de passe dans la BD SQL, alors le service renvoie une instance Client, sinon NULL.

#### Fenêtre pop-up inscription

****

|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Affichage de la fenêtre |
| Valider la création d’un compte | Bouton : «C’est parti !» | Clic | Service CheckClientInputs(...) :  - **Si ok :** RegisterClient(...) et redirection vers fenêtre principale Client, affichage de pop-up de création de compte ok  - Si ok : envoi mail de confirmation  - Sinon : envoi mail d’échec  avec SendMail(client,message)  - **Sinon :** affichage pop-up d’erreur et retour à l’affichage de pop-up inscription |

Liste des services :

Boolean CheckClientInputs(String lastName, String firstName, Date birthDate, String street, String city, Int cp, String phone, String email, String password)

Prend les champs de textes remplis par l’utilisateur en paramètre.

Renvoie True si tous les champs ont été remplis et sont conformes par rapport aux types de valeurs attendues, sinon renvoie False.

**Client RegisterClient(String lastName, String firstName, Date birthDate, String street, String city, Int cp, String phone, String email, String password)**

Prend les champs de textes remplis par l’utilisateur en paramètre.

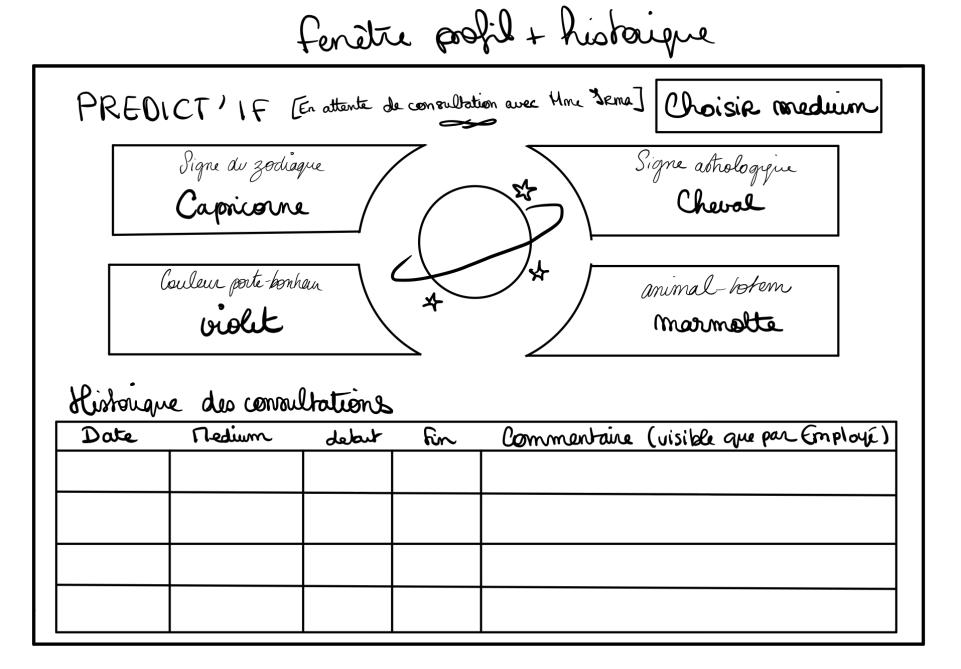
Créer une instance Client à partir de ces informations. Si cette instance est bien créée, elle est renvoyée en retour, sinon NULL.

Void SendMail(Client client, String message)

Prend une instance Client et un string message en paramètre.

Envoi un message à l’adresse mail du client.

#### Fenêtre profil + historique

****

|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Affichage de la fenêtre en fonction du bon mode utilisateur  Affichages des données client avec ClientProfile(client) |
| Si mode = client (et aucune consultation en cours ?) : Demander une consultation | Bouton : «Choisir un médium» | Clic | Affichage de pop-up choix des médiums |

|  |  |
| --- | --- |
| Modes | |
| **Mode client**  - Affichage du bouton « choisir medium »  - Colonne « commentaire » du tableau « historique des consultations » non visible | **Mode employé**  - Bouton « choisir medium » non visible  - Colonne « commentaire » du tableau « historique des consultations » visible |

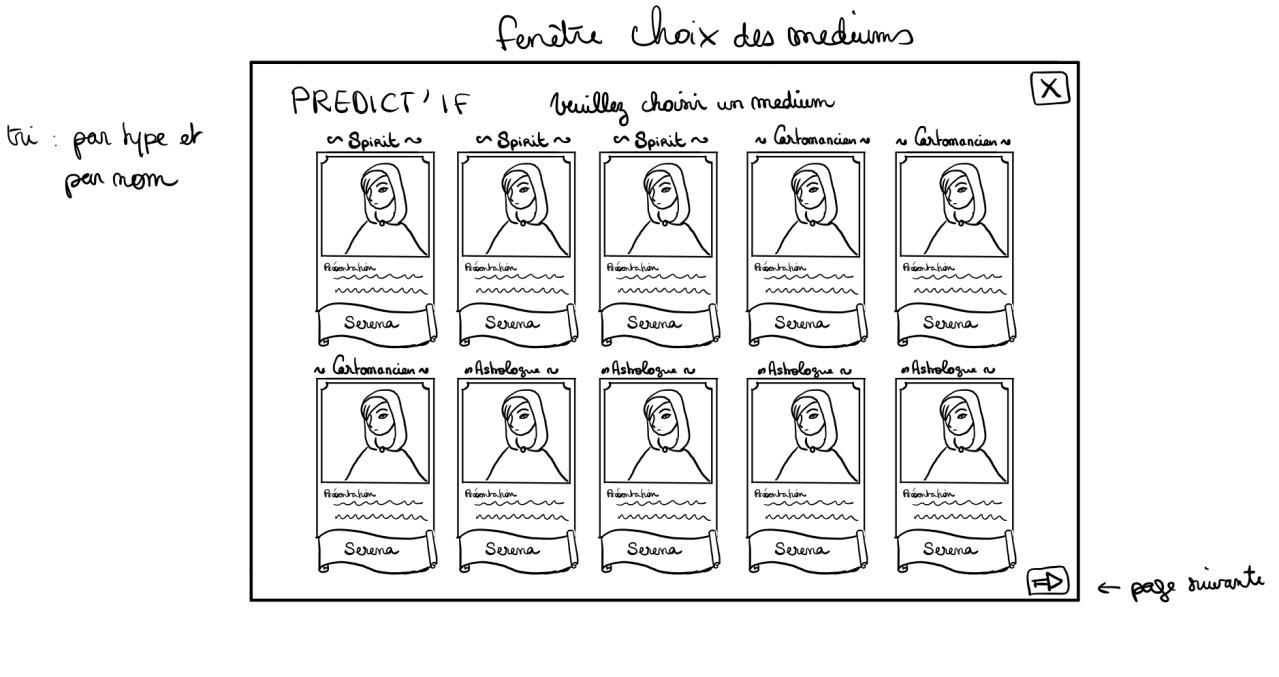
* + - 1. Liste des services :

XXX ClientProfile(Client client)

Prend une instance client en paramètre.

Renvoie le profil astral du client et son historique de consultations.

#### Fenêtre choix des médiums



|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Service ConsultMediumList()  (trie des médiums par type et par nom) |
| Sélectionner un médium | Icône du médium | Clic | Affichage de pop-up de confirmation de choix |
| Valider un médium | Bouton « Valider » de la pop-up | Clic | Service EmployeeReady(medium)  Si ok : Service AskConsultation (medium, client)  Sinon : Affichage de pop-up erreur |
| Visualiser plus de médiums | Bouton flèche directionnelle | Clic | Affiche d’autres icônes de médiums |
| Quitter la pop-up | Bouton « X » | Clic | Redirection vers la fenêtre profil +  historique |

Liste des services :

List <Medium> ConsultMediumList()

Renvoie la liste des mediums de l’agence

Boolean EmployeeReady(Medium medium)

Prend une instance medium en paramètre.

Si un employé de même genre que le medium est disponible, renvoie True, sinon False.

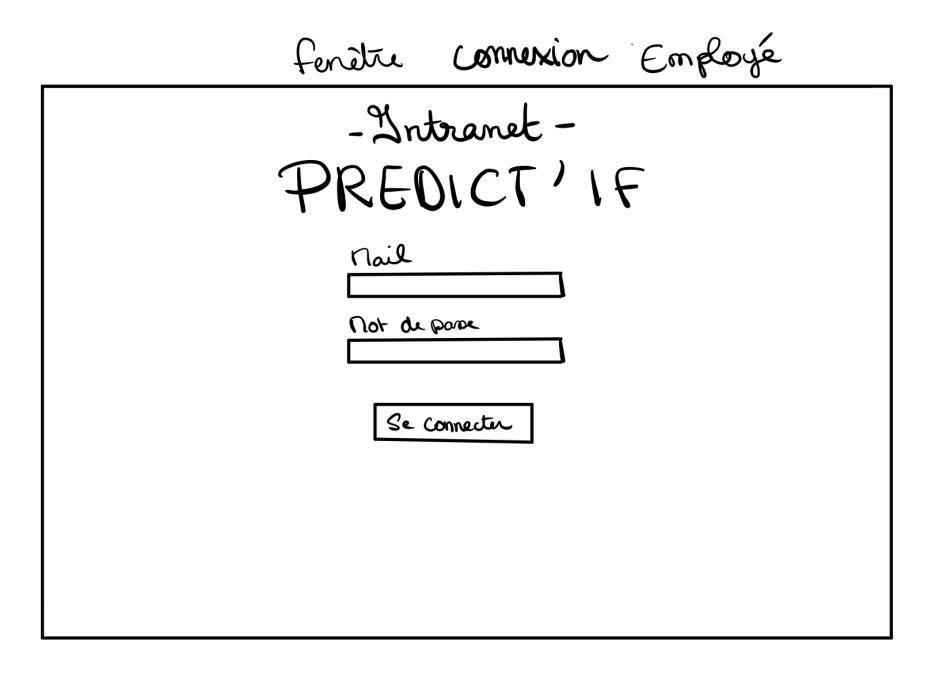
Boolean AskConsultation(Medium medium, Client client)

Prend des instances medium et client en paramètre.

Crée une instance Consultation et met à jour son champ « Date Heure Demande Consultation».

Si elle s’est bien créée, renvoie True, sinon false.

#### Fenêtre connexion employé



|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Affichage de la fenêtre |
| Connexion à un compte employé | Bouton « Se connecter » | Clic | Service AuthenticateEmployee(mail, password) :  - Si ok : Connexion et redirection vers la fenêtre Employé  - Sinon : affichage pop-up d’erreur |

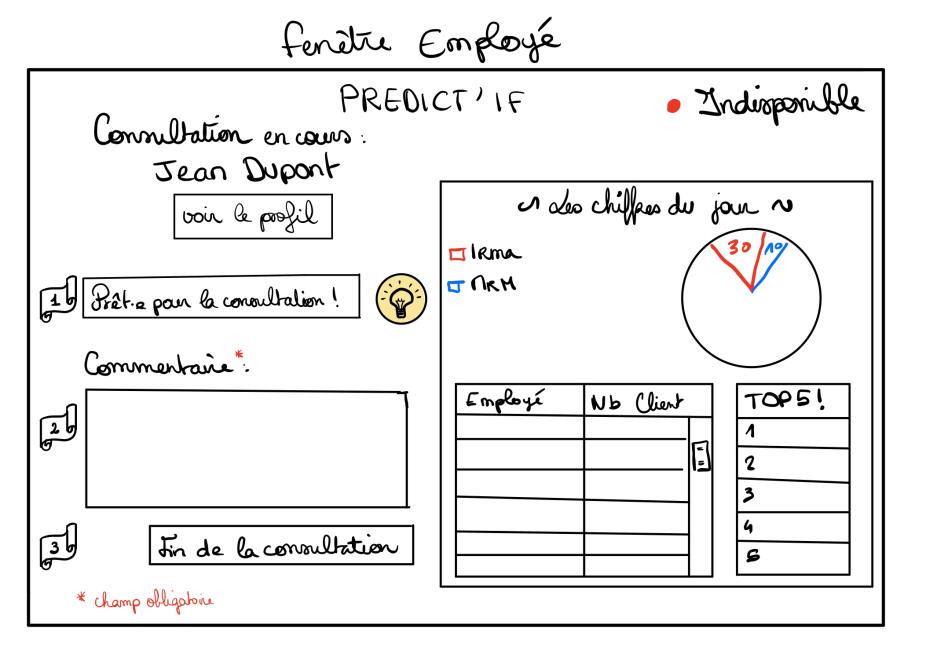
Liste des services :

Employee AuthenticateEmployee(mail, password)

Prend le mail et le mot de passe entrés par l’utilisateur en paramètre.

S’ils correspondent à un tuple mail associé à son mot de passe dans la BD SQL, alors le service renvoie une instance Employee, sinon NULL.

#### Fenêtre Employé



|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  | Service GetStatistics() |
| Si mode = disponible :  Débuter la consultation | Bouton « Prêt.e pour la consultation ! » | Clic | ServiceBeginConsultation() |
| Si mode = indisponible :  Voir le profil client | Bouton « Voir le profil » | Clic | Redirection vers la fenêtre profil +historique |
| Si mode = indisponibles :  Astuces consultation | Bouton Astuce (logo ampoule) | Clic | Redirection vers la pop-up aide consultation |
| Si mode = indisponibles et champs Commentaire is not NULL:  Finir la consultation | Bouton «Fin de la consultation » | Clic | Service EndConsultation( ) |

|  |  |  |
| --- | --- | --- |
| Modes | | |
| **Mode disponible**  - Indication « Disponible » affichée  - Bouton « Voir le profil » grisé  - Bouton « Prêt.e pour la consultation » grisé  - Bouton Astuce (logo ampoule) grisé  - Bouton «Fin de la consultation » grisé | **Mode indisponible et consultation non débutée**  - Bouton « Prêt.e pour la consultation » disponible  - Bouton Astuce (logo ampoule) grisé  - Bouton «Fin de la consultation » grisé | **Mode indisponible et consultation en cours**  - Bouton « Prêt.e pour la consultation » grisé  - Bouton Astuce (logo ampoule) disponible  - Bouton «Fin de la consultation » disponible |
| - Indication « Indisponible » affichée  - Bouton « Voir le profil » disponible  - Nom du client en cours affiché | |

Liste des services :

List <statistics>??? GetStatistics ()

Renvoie une liste de statistiques.

Void BeginConsultation(Employee)

Prend une instance Employee en paramètre.

Met à jour l’instance Consultation pour l’attribut « Date Heure Début Consultation» associé.

Void EndConsultation(Employee)

Pas de paramètres.

Met à jour l’instance Consultation pour l’attribut « Date Heure Fin Consultation» associé.

#### Fenêtre aide consultation

|  |  |  |  |
| --- | --- | --- | --- |
| Intention | Contrôle | Action/Evt | Réponse |
| INITIALISATION |  |  |  |
| Générer des prédictions | Champs remplis +  Bouton « C’est parti ! » | Numéros + Clic | GenererPredictions(love, health, work) :  - Si ok : affichage de prédictions  - Sinon : affichage erreur |
| Quitter la pop-up | Bouton « X » | Clic | Redirection vers la fenêtre Employé |

* 1. Liste des services :
  2. List <String> GeneratePredictions(Int love, Int health, Int work,PARAMETRES IFASTRONET?)

Prend love, health et work en paramètre.

Fait appel au service web IfAstroNet et renvoie les prédictions dans une liste de String.