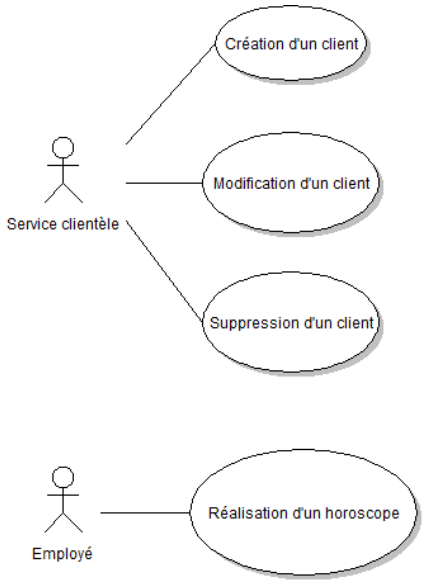
B3326 – FOUCHET Pierre – DRIGON Romaric

3IF – DASI – Dossier d'analyse

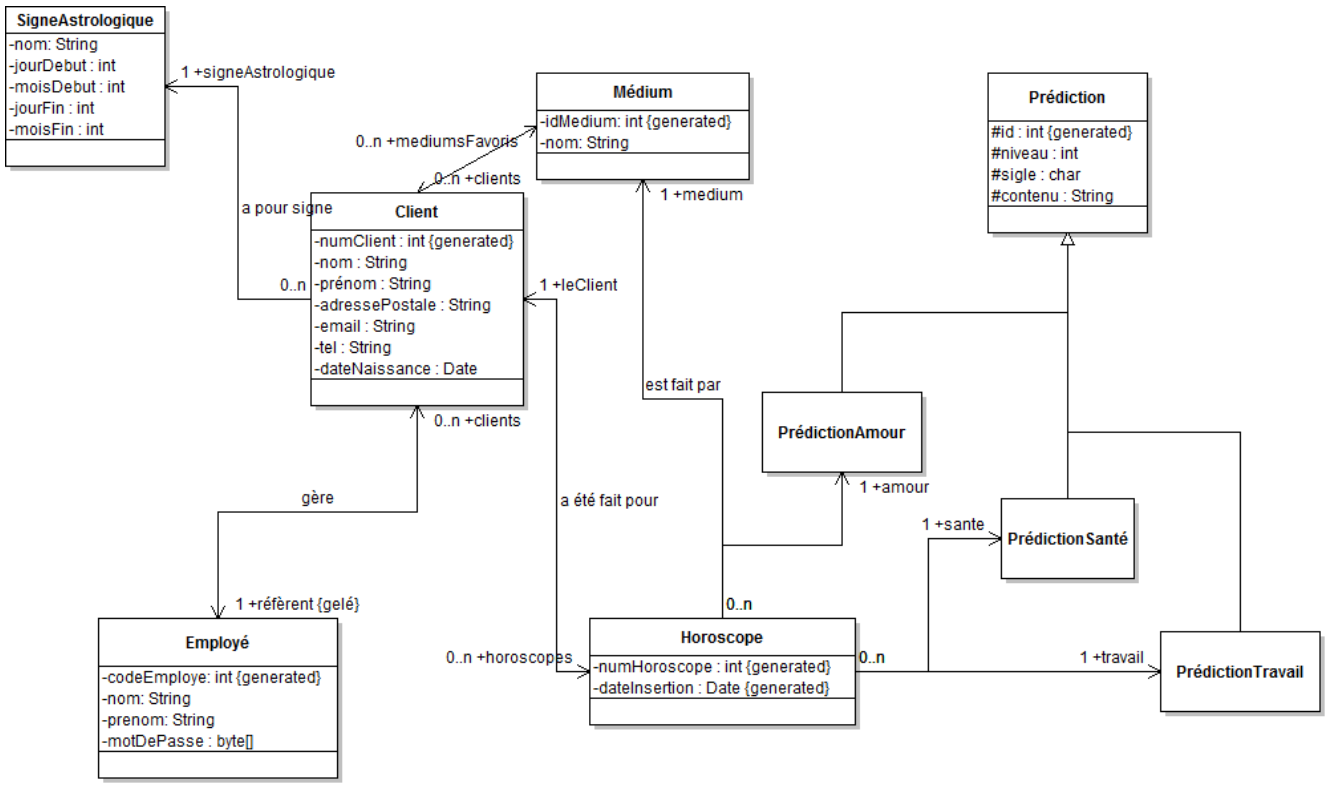
Cas d'utilisation

L'application doit permettre de réaliser les fonctionnalités présentées par le diagramme ci-contre :

- client fenêtré, utilisé par les employés du service clientèle :
  - accès à la liste des clients, leurs détails
  - modification/suppression d'un client existant
  - ajout d'un nouveau client
- client Web, pour les employés (« médiums ») :
  - réalisation de l'horoscope pour un client



Modèle du domaine



Note sur le schéma : bien que les traits sont confondus par l'outil de schéma, il n'y pas de lien direct entre Médium et PrédictionAmour

Nous retrouvons ainsi les principaux objets métier :

- les clients, de la société, s'inscrivent en donnant un certain nombre de détail, et on calcule leur signe astrologique (stockés dans une table séparée, avec notamment les jours de début et de fin des périodes)
- les employés de la société se voient affecter des clients à gérer
- le client choisir un ou plusieurs médium(s) favori(s)
- un de ces médiums réalise, pour un client, un horoscope
- un horoscope est composé de prédictions, à savoir une prédiction amour, une santé et une travail

Dans notre modélisation, nous avons fait plusieurs choix :

- utiliser une classe Prédiction, dont héritent trois fils, permet une meilleure réutilisation du code
- stocker l'association horoscope <> client, navigable dans les 2 sens : bien que le besoin n'ai pas été exprimé dans le cahier des charges, il peut être utile de lister tous les horoscopes réalisés pour un client
- stocker la date de réalisation d'un horoscope, qui pourrait être utile pour les impressions par exemple
- un employé, pour se connecter, en plus de son numéro, fournit un mot de passe. Il est utile pour sécuriser l'application Web, ouverte à tous dans le cas d'un extranet, et être certain de l'identité de la personne connectée sur l'application client fenêtré

Diagrammes de classe complets en annexe

Contraintes et règles de gestion métier

Les contrôles de surface concernant la validité des données entrées (formation correcte des adresses emails par exemple) doivent être pris en charge par l'IHM. La date de naissance n'est pas contrôlée, elle peut être supérieur à la date actuelle, néanmoins elle ne posera pas de problème pour le calcul du signe astrologique associé.

Une règle de gestion concernant l'assignement de l'employé au client lors de sa création a été définie. En effet, l'employé possédant le moins de clients sera associé lors de l'enregistrement par le service client d'un nouveau client. Cette assignation se fait de manière automatique.

Description des services

La couche service permet d'effectuer un certain nombre de traitements sur le modèle :

- Création d'un client : créer un nouveau client en utilisant les paramètres présent dans la classe Client du modèle du domaine et le persiste. Fonction associée : *createClient(…)*
- Mise à jour d'un client : modifie l'état interne du Client. Toutes les valeurs des attributs sont modifiables néanmoins le signe astrologique est systématiquement calculé par rapport à la date de naissance. De plus, le réfèrent ne devrait jamais être modifié. Fonction associée : *updateClient(Client client)*
- Suppression d'un client : supprime un client ainsi que ses horoscopes de la base de donnée (cascade). Fonction associée : *deleteClient(Client leClient)*
- Création d'un horoscope pour un client. Ce service ne vérifie pas si l'employé est connecté. Il permet de choisir le triplet de prédictions Amour-Santé-Travail, le Medium signant et le client puis persiste l'entité. Fonction associée : *createHoroscope(…)*
- Des méthodes de récupération d'entités (*getAllClients(), getAllMediums(), getPrediction(), retrieveClient()*).
- Une méthode permettant de récupérer les horoscopes selon l'identifiant ou la date d'insertion. Cette méthode permettrait de préparer les différents horoscopes à envoyer au client (cette partie-là n'étant que peu expliquée dans le cahier des charges, la plus grande latitude a été recherchée). Fonction associée : *getDetailsHoroscope(…)*
- Des méthodes privées présentées dans la javadoc et utilisées par les services ci-dessus.

De plus, un certain nombre de services existent afin de remplir la base de données initialement. Ces services, présent dans la classe *PreparerBD* n'ont pas pour but d'être réutilisés dans l'application.

B3326 – FOUCHET Pierre – DRIGON Romaric

Diagramme de séquences

Voici le diagramme de séquence concernant la fonctionnalité de réalisation d'un horoscope. La granularité concernant les notes est très fine, ainsi la plupart des méthodes sont codées en une seule.

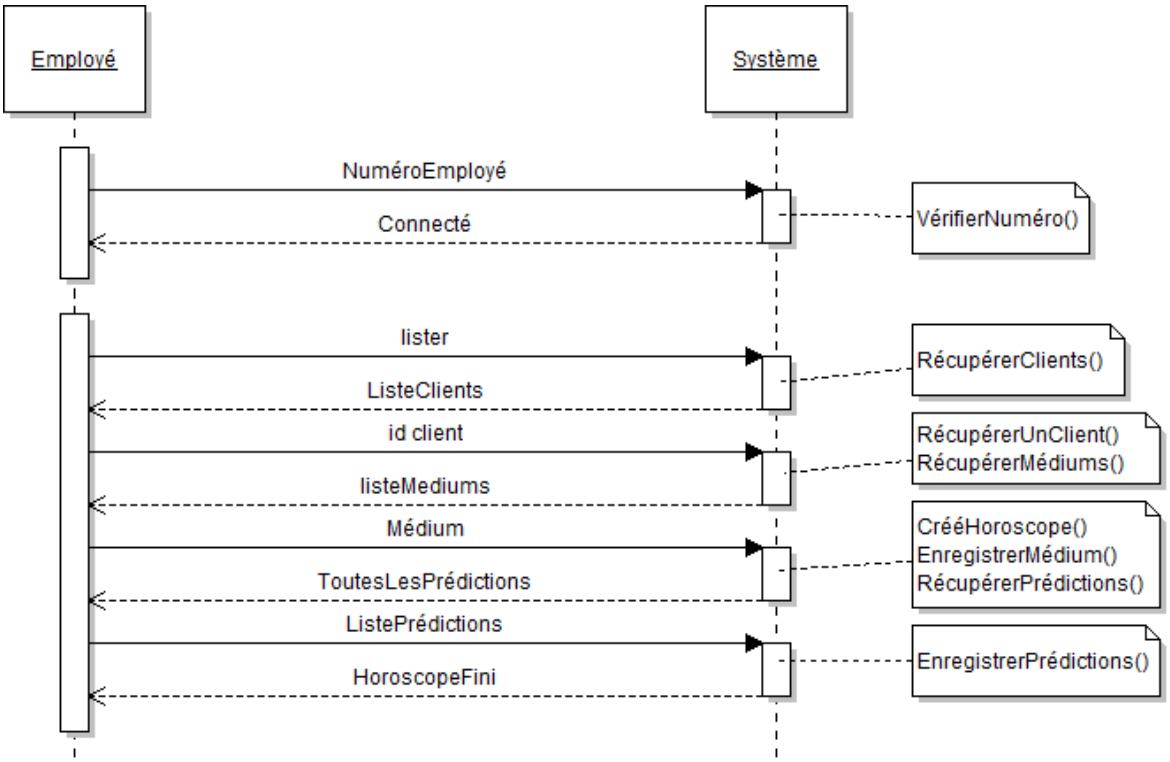


Diagramme de séquence - Création horoscope

Le second diagramme ci-dessous se veut représentatif de la logique adoptée pour créer, modifier et supprimer un client en utilisant la persistance.

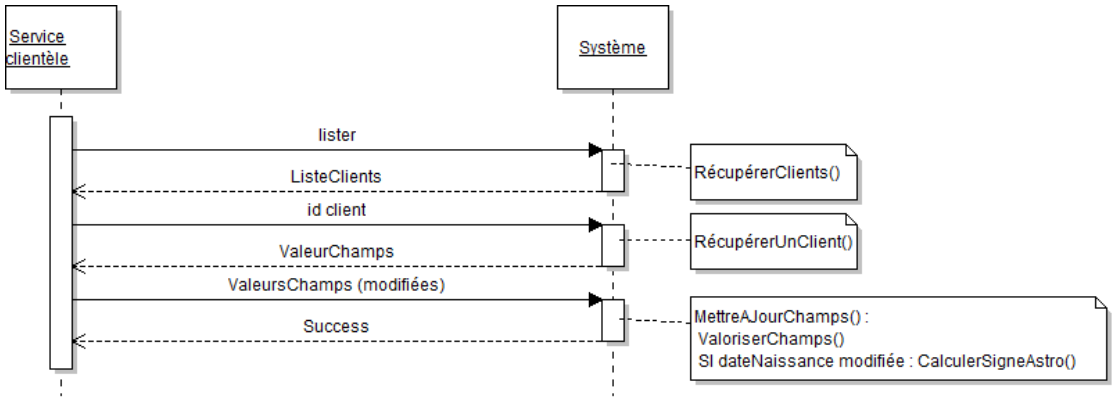


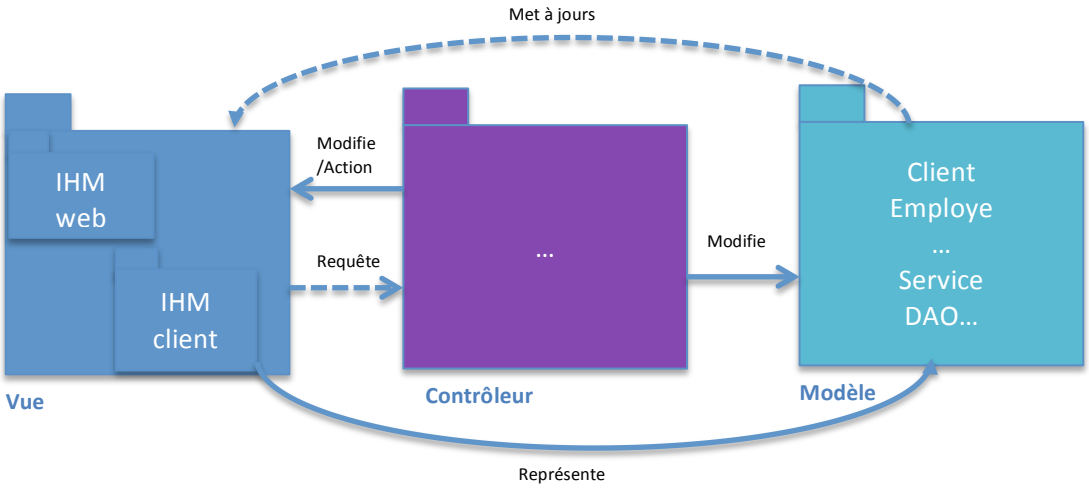
Diagramme de séquence - Modification client

On peut considérer que les cas d'utilisations créer, supprimer et modifier un client partagent la même séquence initiale, se composant d'une connexion avec le numéro d'employé et listant les différents clients.

B3326 – FOUCHET Pierre – DRIGON Romaric

Architecture de l'application

L'application est organisée selon une architecture 3 couches ou Modèle-Vue-Contrôleur. Ici seule la partie Modèle est implémentée.

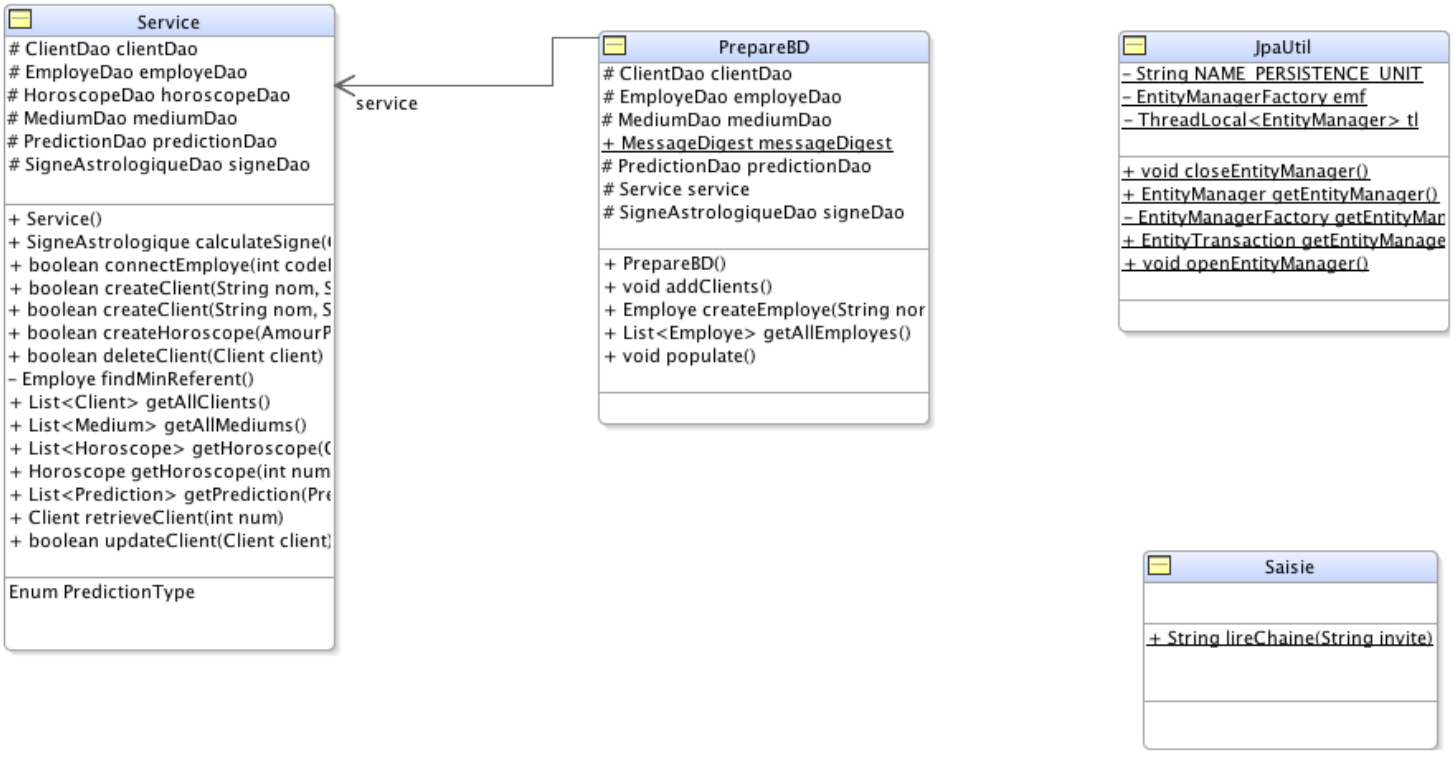


Le package Modèle est composé de différentes couches : objets métiers, Métier (ici Service) et DAO. Nous ne rentrerons pas dans les détails dans ce dossier car il s'agit du modèle de développement vu en cours, néanmoins des explications concernant le nommage peuvent être utile :

- La couche service est représentée par les classes Service et PreparerBD.
- La couche Objets métiers correspond aux noms des entités présentes dans le modèle du domaine
- La couche DAO est composée des objets métiers suffixés par –Dao.

Annexes

Annexe 1- Diagramme de classe des classes « utilitaires »



Annexe 2 – Diagramme des classes DAO

Annexe 3 – Diagrammes des classes métier