

## Diagramme de classes

Notre application a été développée selon une architecture de type Modèle-Vue-Contrôleur,

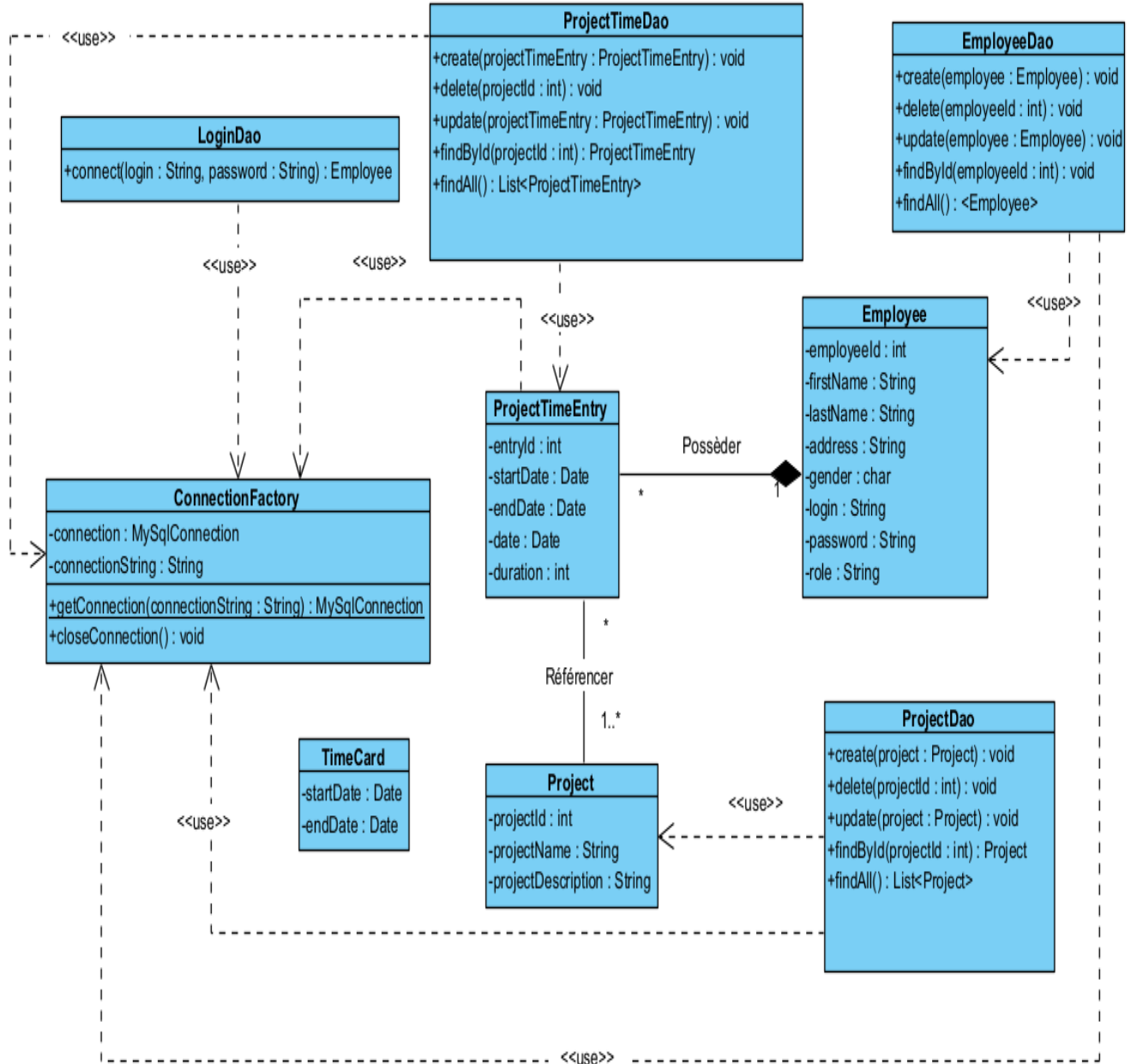
➔ La partie vue représente la partie du code responsable des interactions avec l'utilisateur (ici, c'est l'interface graphique)

➔ La partie modèle représente les classes métier (Employee, Project, ProjectTimeEntry et TimeCard) ainsi que les classes responsables de l'accès à la base de données (classes XXXDao).

➔ La partie contrôleur représente quant à elle, l'interface ou le pont entre la vue et le modèle. Elle traduit les interactions de l'utilisateur en appels aux méthodes des classes du modèle et renvoie les résultats de ces appels à la vue pour affichage.

Comme notre application contient beaucoup de classes, et il nous est impossible de toutes les schématiser. Nous avons opté pour une vue partielle du diagramme de classes.

Le diagramme suivant montre les relations entre les classes métier).

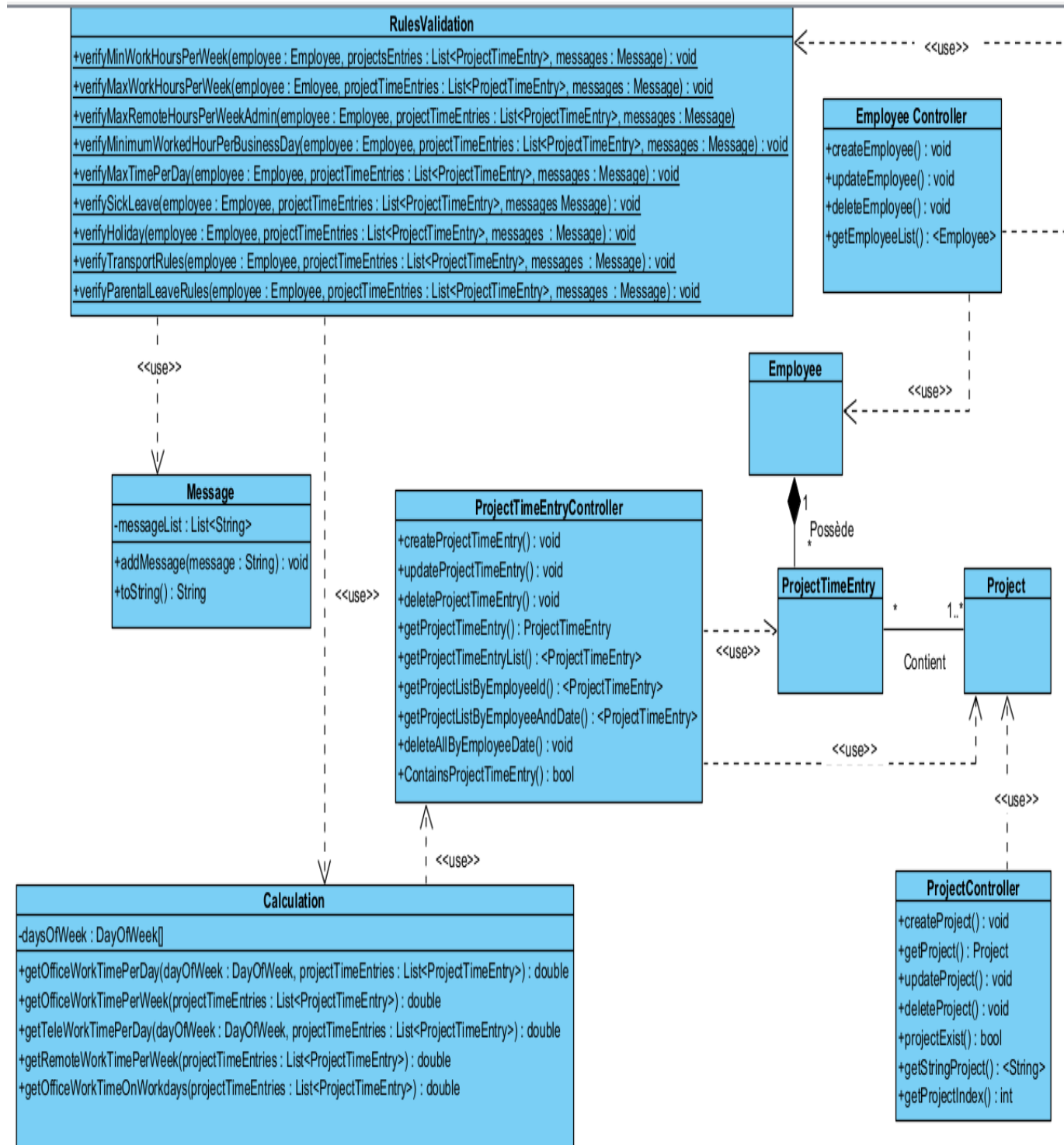


Les classes XXXDao représentent des classes de la couche Accès aux données. Elles implémentent les opérations CRUD sur les tables de la BD (Employee, Project et ProjectTimeEntry).

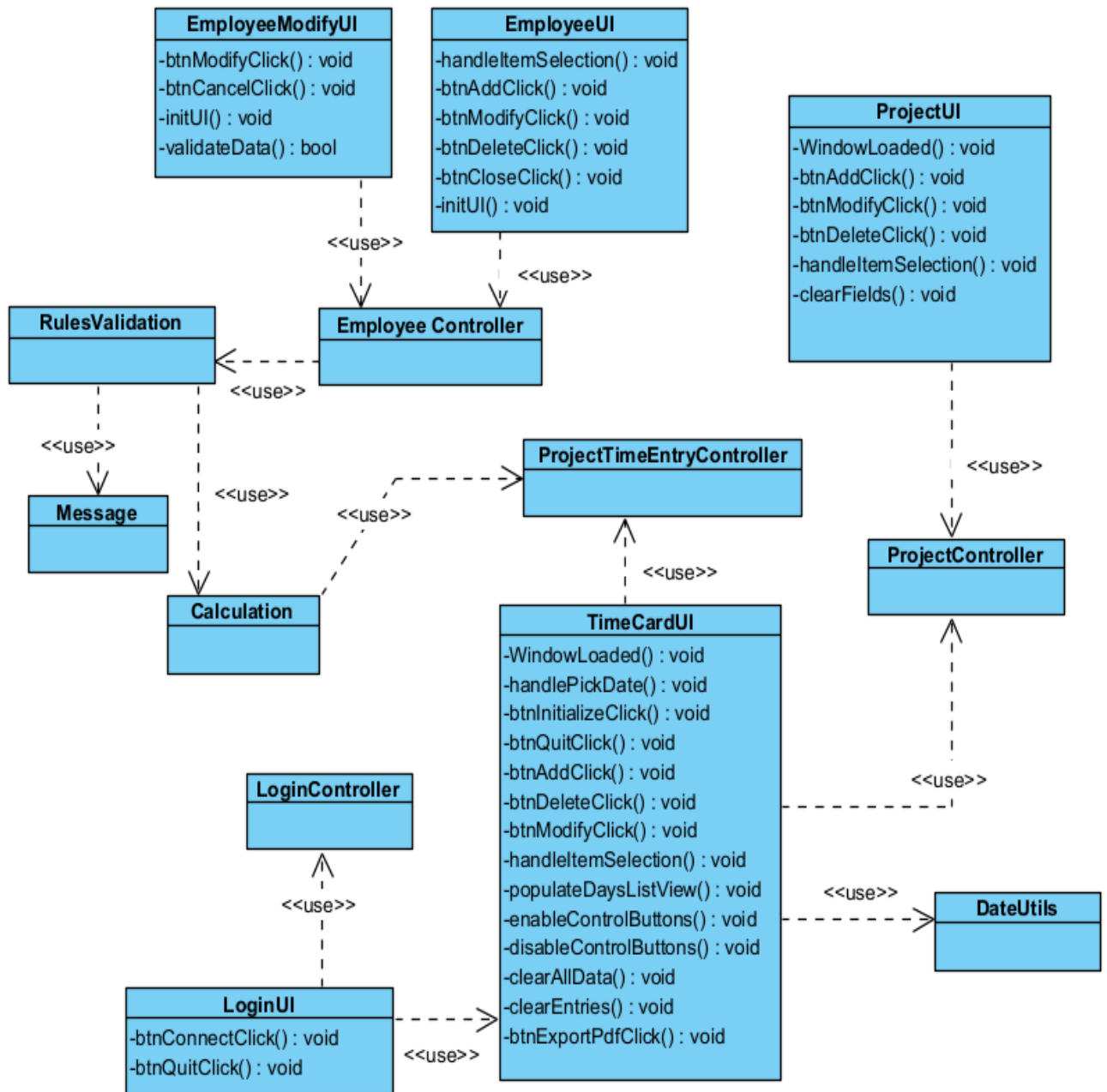
Contrairement aux autres classe XXXDao, la classe LoginDao implémente une seule méthode connect() qui permet d'authentifier un employé et lui permettre d'utiliser l'application. Elle accepte en paramètre un login et un mot de passe et renvoie une référence vers un objet Employee si la connexion réussie, ou null en cas d'échec.

La classe TimeCard comporte deux attributs de type Date : startDate qui indique le début de la semaine de la carte de temps et endDate qui indique la fin de la semaine de celle-ci. Elle n'est liée ici à aucune classe et sera utilisée par la classe ProjectTimeEntryService (non représentée ici) afin d'extraire les entrées de projets d'une certaine semaine de travail.

Le diagramme suivant montre les relations entre les classes métiers, les contrôleurs et les classes utiles (classes de calcul, de validation et utilitaire).



Le diagramme suivant montre les relations entre les classes de l'interface graphique, les classes contrôleur et les classes utiles



Les classe Calculation, RulesValidation et Message ont été déjà utilisées dans les sprints 1 et 2. Il s'agit de classes pour :

- ➔ Le calcul des différents totaux des heures de travail de la semaine/par jour.
- ➔ Les messages liés à la validation d'une carte de temps.
- ➔ La vérification des différentes règles de validation du système.

Les classes de l'interface graphiques sont

- LoginUI : écran de connexion d'un employé (fenêtre affichée lors du lancement de l'application).
- EmployeeUI : écran de gestion des employés (ajout, suppression, modification).
- EmployeeModifyUI : écran de modification de certaines informations d'un employé.
- ProjectUI : écran de gestion des projets (ajout, suppression, modification).
- TimeCardUI : écran de gestion des entrées de projets (ajout, suppression, modification).

Le rôle joué par l'employé qui se connecte à l'application détermine l'interface vers laquelle il sera redirigé. Nous avons mis en place deux rôles :

- Employee : rôle d'un employé ordinaire, qui consiste à saisir sa feuille de temps de l'enregistrer. Ce travail se fait dans l'interface TimeCardUI.
- Manager : rôle du responsable (unique), qui consiste à gérer les employés, les projets et à valider les cartes de temps. Ce rôle lui permet d'accéder aux interfaces d'administration du système.