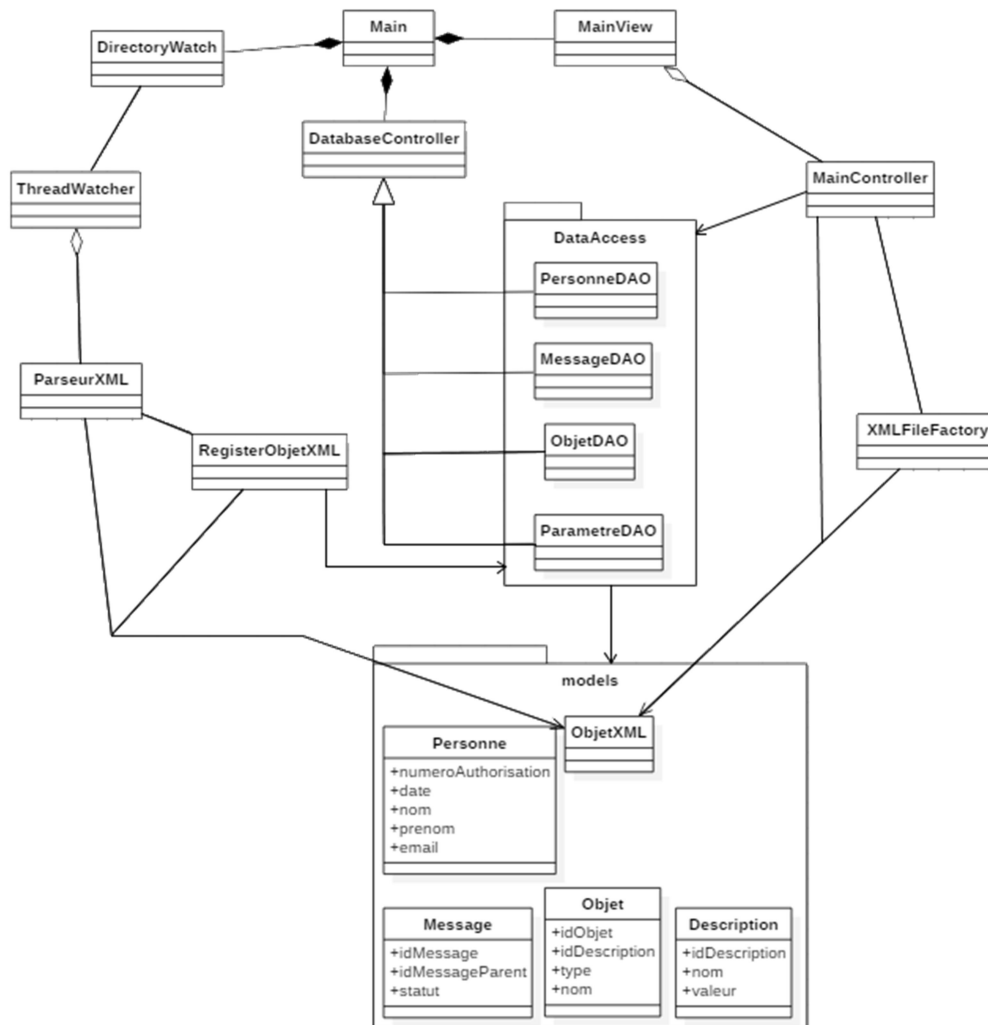


Diagramme de classes avec leur explication



Main	Point d'entrée du projet, lance le thread « DirectoryWatch » qui scrute un dossier pour voir si un nouveau fichier XML est arrivé, lance l'instanciation de la vue « MainView » et demande la connexion à une base de données au « DatabaseController »
DirectoryWatch + ThreadWatcher	Surveillent un dossier pour savoir si un nouveau fichier XML est arrivé et le transmet au « ParseurXML »
ParseurXML	Prend le fichier XML en paramètre, vérifie si le fichier est bien structuré et si oui construit un POJO « ObjetXML » et le donne au « RegisterObjetXML »
RegisterObjetXML	Prend l'« ObjetXML » qui est créé par le « ParseurXML » pour l'enregistrer en base de données via le package « dataAccess »
Package dataAccess et classes DAO	A accès à la base de donnée suivant ce que l'on cherche à traiter grâce à la connexion établie par la classe « DatabaseController »
DatabaseController	Crée la base de données si elle n'existe pas, la remplit avec des données si l'on le souhaite via une variable et s'y connecte
MainView	Fichier XML qui structure la vue du logiciel et définit son contrôleur comme « MainViewController »
MainViewController	Code java qui charge ou lit les champs de la vue. Cette classe appelle aussi le créateur de fichier XML « XMLFileFactory » en lui donnant un « ObjetXML » qui aura correctement instancié
XMLFileFactory	Prend un « ObjetXML » en paramètre et crée un fichier XML qui sera à envoyer à un contact
Package models	De simple POJO