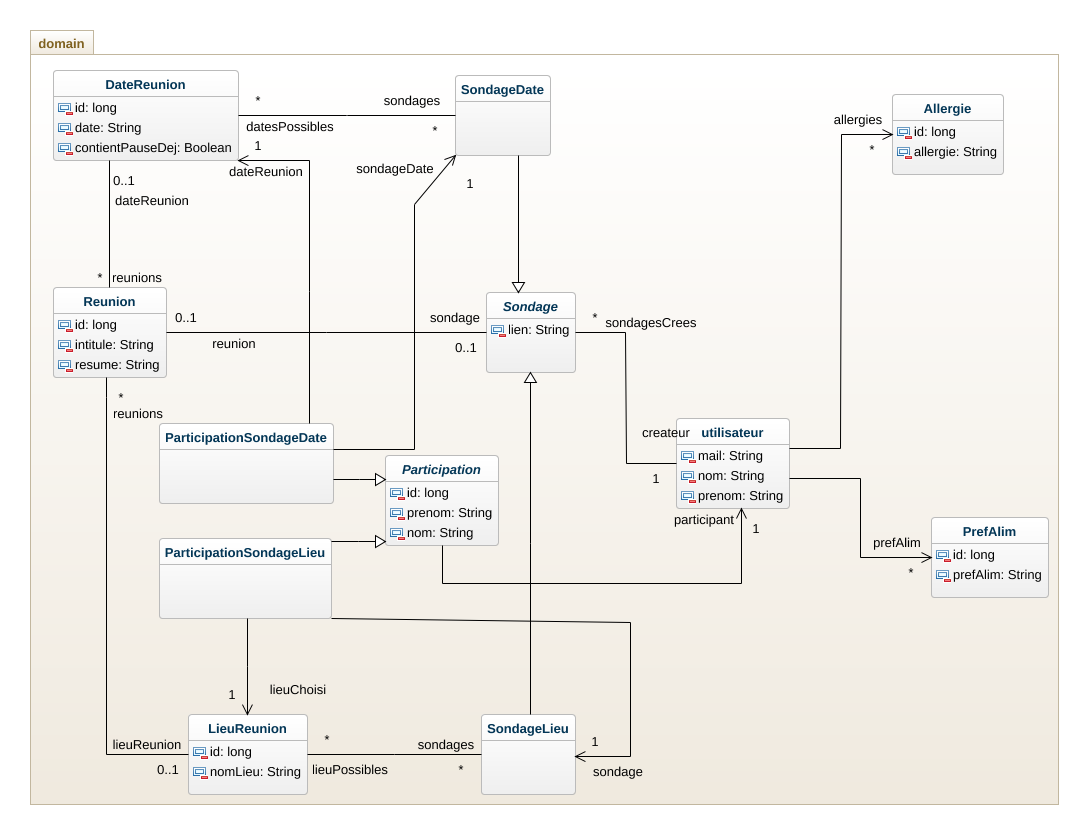
# Compte rendu back end projet

Pour mettre en place notre back end, nous avons utilisé Java et Jersey.

Pour commencer le back end, nous avons mis en place un diagramme de classe :

Puis nous avons créé une classe d’entité par classe présente sur ce diagramme afin de mettre en place notre base de données. Pour les sondages, nous avons distingué ceux concernant des lieux et ceux concernant des dates.

Après avoir créé ces entités nous avons organisé notre code comme suit :

* Un package «  domain » avec toutes les entités
* Un package « dao » regroupant toutes classes réalisant des opérations sur la base de données
* Un package « dto » avec toutes les classes permettant d’organiser les données d’un objet
* Un package « rest » constitué de tous nos classes de services
* Un package « servlet » où sont regroupés les servlets demandées pour le TP4
* Un package « jpa » constitué de la classe permettant de gérer l’EntityManager

## Le package « dao »

Après analyse du projet, nous avons défini trois classes de DAO :

* ParticipationDAO
* SondageDAO
* UtilisateurDAO

Ces classes ont été définies selon nos besoins. En effet, Participation, Sondage et Utilisateur sont les entités sur lesquelles nous avions besoin de faire des opérations de projection, de création et de modification.

## Le package « dto »

Dans notre diagramme classe nous avons plusieurs fois des associations ayant la multiplicité \* à chaque bout de l’association. Ceci a posé problème lorsque nous voulions récupérer des objets de type SondageDate, par exemple car le nombre d’élément contenus dans le JSON était infini. Pour résoudre ce problème nous avons mis en place des classe DTO.

Ces classes sont :

* Utilisateur DTO
* SondageDTO
* LieuReunionDTO
* DateReunionDTO

Pour résoudre le problème précédemment énoncé les listes d’objets contenues par les entités ont été remplacées par des listes de chaînes de caractère content la clé primaire de ces objets.

## Le package « rest »

Ce package regroupe trois services liés aux classes DAO exposées ci-dessus :

* ParticipationService
* SondageService
* UtilisateurService

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Type d’opération | Fonction | Objet retourné |
| /participation/sondageLieux | GET | Retourne toutes les participations à des sondages de type lieu | List<ParticipationSondageLieu > |
| /participation/sondageDates | GET | Retourne toutes les participations à des sondages de type date | List<ParticipationSondageDate> |
| /participation/sondageLieux | POST | Créer une participation pour un sondage de type lieu | Participation |
| /participation/sondageDates | POST | Créer une participation pour un sondage de type date | Participation |
| /participation/sondageLieux/count | GET | Retourne le nombre de participation pour chaque lieu et sondage de type lieu | List<ParticipationSondageLieu> |
| /participation/sondageDates/count | GET | Retourne le nombre de participation pour chaque date et sondage de type date | List<ParticipationSondageDate> |

Le service ParticipationService regroupe les urls ci-dessous :

Lorsqu’un utilisateur participe à un sondage s’il s’agit du créateur du sondage, son choix validera la date ou le lieu pour le sondage concerné. Un mail sera donc envoyé à tous les participants qui ont choisi le même lieu ou la même date que ce qui a été validé. De plus, un objet Reunion sera crée et insérer dans la base.

Le service SondageService regroupe les urls ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Type d’opération | Fonction | Objet retourné |
| /sondage | GET | Retourne tous les sondages | List<SondageDTO> |
| /sondage/sondagesLieux | GET | Retourne tous les sondages de type lieu | List<SondageDTO> |
| /sondage/sondagesLieux | POST | Créer un sondage de type lieu | Sondage |
| /sondage/sondagesDates | GET | Retourne tous les sondages de type date | List<SondageDTO> |
| /sondage/sondagesDates | POST | Créer un sondage de type date | Sondage |
| /sondage/sondagesDates/{lienSondage} | GET | Retourne le sondage de type date dont le lien est passé en paramètre de l’url | SondageDTO |
| /sondage/sondagesLieux/{lienSondage} | GET | Retourne le sondage de type lieu dont le lien est passé en paramètre de l’url | SondageDTO |
| /sondage/lieux | GET | Retourne tous les lieux concernés par des sondages | List<LieuReunionDTO> |
| /sondage/dates | GET | Retourne toutes les dates concernées par des sondages | List<DateReunionDTO> |
| /sondage/date/{lienSondage}/{date} | GET | Retourne l’objet DateReunion correspondant au lien du sondage et à la date passés en paramètre de l’URL | DateReunionDTO |
| /sondage/lieu/{lienSondage}/{lieu} | GET | Retourne l’objet LieuReunion correspondant au lien du sondage et au lieu passés en paramètre de l’URL | LieuReunionDTO |

Le service UtilisateurService regroupe les urls ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Type de requête | Fonction | Objet retourné |
| /utilisateur | GET | Retourne tous les utilisateurs | List<UtilisateurDTO> |
| /utilisateur | POST | Créer un utilisateur | Utilisateur |
| /utilisateur/{emailUtilisateur} | GET | Retourne l’utilisateur dont le mail est passé en paramètre de l’url | UtilisateurDTO |
| /utilisateur/allergies/{emailUtilisateur} | POST | Met à jour les allergies d’un utilisateur |  |
| /utilisateur/preferences/{emailUtilisateur} | POST | Met à jour les préférences alimentaires d’un utilisateur |  |