

Documentation

I. Modélisation du problème:

- **Analyse du problème:**

[illegible]

id_membre → adr_email, id_membre → mot_de_passe id_membre → nom id_membre → prenom id_membre → adr_postale id_membre → somme_due id_membre → somme_due id_adherent → id_membre id_res_ref → date_res, id_res_ref → heure_res id_res_ref → nb_nuits id_res_ref → nb_nuits id_res_ref → prix_res id_res_ref → emailref id_res_ref → id_membre id_res_form → id_form id_res_form → idadh id_res_form → anneeform id_res_form_att → rangattente id_loc_mat → idadherent id_loc_mat → nbpieces id_loc_mat → daterecup id_loc_mat → dateretour id_loc_mat → nbrpiecesperdu es	Ext(id_reserv_form_attente) \subseteq Ext(id_reserv_form) dateretour - daterecup \leq 2 semaines nbpiecereserve(location) \leq nbpiece(materiel)Ext(id_reserv_form)		Conserver les actions effectuées par un membre (formations/réservations/locations) après la suppression des données personnelles.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	------------------------------------------------------------------------------------------------------------------------------------------

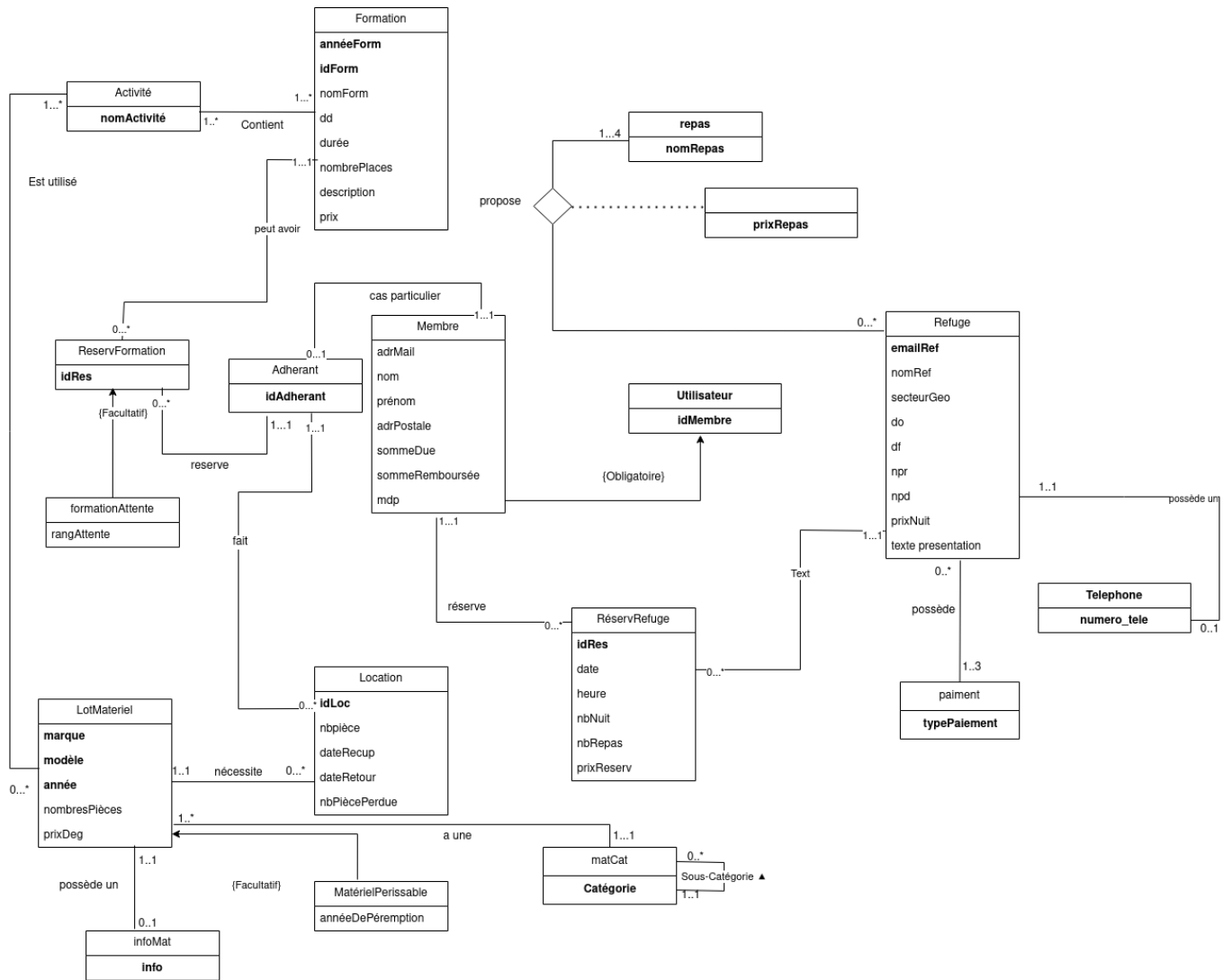
● **Explication des choix sur les points difficiles :**

Dans l'ensemble des entités, la construction et attributs a été faite comme mentionnée dans le sujet. On cite quelques points difficiles:

→ **Intégration du numéro de téléphone des refuges**: Les refuges peuvent avoir un numero de telephone comme ils peuvent ne pas l'avoir ce qui nous interdit de mettre l'attribut "num_tel" dans l'entité refuge ,en effet notre principe dès le début du projet était d'essayer de conserver les formes normales en maximum ainsi qu'éviter les redondances, et donc le fait d'avoir un attribut qui peut être nulle dans l'entité refuge était un

“non”(1FN) pour nous, ce qui nous a amené à mettre cette condition dans les contraintes de multiplicité.

- **Identification des formations:** Dans le sujet ,il était indiqué que les formations doivent être identifiées par l’année durant laquelle elle est dispensée ainsi que par un numéro unique correspondant à son rang dans les présentations de l’année .Ainsi pour la même raison de conservation de forme normales comme précédemment , on a décidé de diviser la clef en deux éléments séparés , ce qui a donné la clef suivante: (**anneeform , id_form**) pour avoir un attribut de type atomique.
 - **Choix de gestion des repas proposés:** Un seul refuge peut proposer plusieurs repas et tous les refuges ne proposent pas les même repas, ce qui nous a amené à construire une nouvelle entité de repas qui contient le nom du repas et un attribut d’association entre cette dernière et la table refuge qui est le prix du repas.
 - **Gestion de l’entité du LotMateriel:** Vu que le matériel est organisé sous forme de lots, on a pensé à créer une entité “LotMateriel” au lieu de stocker chaque matériel pour éviter d’éventuelles redondances.
 - **Les réservation de formation en attente:**Pour gérer ce problème, on a décidé de considérer les réservations de formation en attente comme sous-type des réservations de formation, ainsi cette dernière entité contiendra comme attributs : id_adh et rangattente.
 - **Le droit à l’oubli (RGPD):** Pour l’établir , on a fait le choix de créer un entité utilisateur qui a comme clé id_membre , qui a comme sous type Membre et contient tous les données personnels de l’utilisateur qui peuvent être supprimés aisement si jamais ce dernier décide d’utiliser son droit à l’oubli
- **Modèle entité association (notation UML) :**
On a traduit le tableau des dépendances fonctionnelles sous forme de diagramme UML, en tenant compte des contraintes de multiplicités.



II. Implantation de la base de données:

- **Traduction du schémas entité association en schéma relationnel:**

Dans cette étape, on a commencé par traduire les entités simples, puis les entités faibles, les associations binaires (1..1), celles de cardinalités (0..1), celles de cardinalités (x..*), et finalement les sous entités ou on a décidé d'adopter une méthode très simple qui se résume en ce qui suit "A et la sous-entité B, créer une relation A avec les propriétés de l'entité A dont la clef sera composée des attributs clés de l'entité A. Créer une relation B avec les attributs de l'entité B et la clef de la relation A (clef étrangère). La clef de la relation B est composée des attributs clés de la relation A", ce qui donne les entités suivantes:

Refuge{emailref, nomRef, SecteurGeo, date_ouverture, date_fermeture, nb_places_repas, nb_places_dormir, prix_nuit, texte_presentation};

Telephone{num_tel, emailref}

Refuge_Paiement{type_paiement, emailref}

ProposeRepas{emailref, nom_repas, prix_repas}

ReserveRefuge{idMembre, idRes, date_res, heure_res, nb_nuits, nb_repas, prix_res, emailref}

Repas{nom_repas}

Utilisateur{idMembre}

Membre{idMembre, adr_mail, nom, prenom, adr_postale, somme_due, somme_reglee, mot_de_passe}

Adherent{idAdherent, idMembre}

Locations{idloc, nbPieces, dateRecup, dateRetour, nbPiecesPerues, marque, modele, annee, idAdherent}

ReservFormation{idRes, idAdherent, idForm, anneeForm}

FormationAttente{idRes, rangAttente}

Formation{idForm, anneeForm, nomForm, date_debut, duree, nbPlaces, uneDescription, prix}

FormationActivite{nomAct, anneeForm, idForm}

Activitee{nomActivite}

LotMateriel{marque, modele, annee, nbPieces, prixDeg, categorie}

MatUtilActiv{nomAct, marque, modele, annee}

infoMateriel{info, marque, modele, annee}

matCat{categorie}

materielPerissable{marque, modele, annee, datePeremption}

categorieMere{sousCategorie, categorieMere}

Paiement{typePaiement}

Formes Normales :

Le schéma relationnel satisfait les 3 formes normales ainsi que la forme normale de Boyce-Codd-Kent.

1ère forme normale : tous les attributs sont atomiques et ne peuvent pas être NULL, exemple : le numéro de téléphone est optionnel pour les refuges , au lieu de l'insérer comme attribut , une table Téléphone a été créée avec une cardinalité 0..1 du côté du téléphone et 1..1 du côté du refuge.

2ème forme normale : Pour chaque table , Tous les attributs non clés sont pleinement dépendants de chacune des clés (les dépendances fonctionnelles sont toutes élémentaires)

exemple : la table **Locations**

on a bien les dépendances fonctionnelles suivantes :

id_loc_mat → idadherent

id_loc_mat → nbpieces

id_loc_mat → daterecup

id_loc_mat → dateretour

id_loc_mat → nbrpiecesperdu

3ème forme normale : Tout attribut non clé ne dépend pas d'un ensemble d'attributs qui n'est pas une clé.

Forme normale de Boyce-Codd-Kent : Pour toute dépendance fonctionnelle ($X \rightarrow Y$) non triviale, X contient une clé de R

Cela est vérifiée par le fait que toutes les dépendances fonctionnelles possèdent un côté droit ne contenant que les clés.

Cela est le résultat d'une bonne analyse et d'une traduction prudente au relationnel (On a évité toute redondance lors de la création de la création du schéma entité association)

III. Analyse des fonctionnalités

1. Parcours et Réservation des refuges:

Le parcours des refuges s'effectue en fonction de leur disponibilité. En effet, nous calculons le nombre de places disponibles dans chaque refuge pendant la date du jour, ceci inclut le nombre de places pour dormir et le nombre de places pour les repas.

Les refuges sont, par la suite, affichés de manière ordonnée, avec un classement ascendant basé sur le nombre de places libres, créant ainsi une visualisation organisée qui guide les utilisateurs vers les options les plus propices à leur demande.

En matière de réservations, la disponibilité est étroitement liée au nombre de places disponibles en fonction de la date de réservation et du nombre de nuitées (le cas échéant), et/ou le nombre de repas. En d'autres termes, elle est déterminée par la capacité d'accueil durant la période souhaitée par le client, prenant en considération les réservations déjà confirmées. Cette évaluation intègre attentivement la disponibilité tant pour l'hébergement

que pour les repas. Dans l'éventualité où le refuge spécifié ne dispose pas d'un nombre suffisant de places, la demande de réservation est déclinée. Cette approche assure une gestion efficace des réservations.

Dans le cas où la réservation est bien prise en compte, nous calculons le prix du séjour et nous l'ajoutons à la somme due par le client dans la table MEMBRE.

2. Parcours et Réservation des formations:

Les formations sont affichées par ordre croissant de leur date démarrage et par l'ordre alphabétique.

Concernant la réservation d'une formation, on vérifie en premier temps que l'utilisateur est un adhérent, ensuite on passe à la vérification de la disponibilité des places de la formation choisie. Dans le cas où des places sont libres, la réservation est effectuée et le paiement est pris en compte. Dans le cas contraire, la réservation est considérée et l'utilisateur est mis en liste d'attente sans effectuer le paiement.

Pour le calcul des places disponibles dans une formation, on ne prend pas en compte les réservations dont le client est en liste d'attente.

3. Parcours et Réservation des matériels:

Le parcours des matériels s'effectue en fonction de leur disponibilité. En effet, nous calculons le nombre de pièces disponibles dans chaque lot dans le jour. L'affichage est par activité.

En matière de location du matériel, la disponibilité est liée au nombre de pièces disponibles en fonction de la date de récupération et en prenant en considération les locations déjà confirmées ainsi que les matériels périmés. Dans l'éventualité où le lot spécifié ne dispose pas d'un nombre suffisant de pièces, la demande de location est refusée. Cette approche assure une gestion efficace des locations.

La fonctionnalité du retour du matériel est possible. On demande ainsi la location faite et le nombre de pièces perdues/abîmées et on met à jour la somme due au client.

Le nombre de pièces perdues/abîmées est toujours pris en compte lors du calcul du nombre de pièces disponibles pour l'affichage ainsi que pour la location.

4. Droit à l'oubli:

On procède par vérifier en premier temps si la somme due est réglée ensuite, on crée un utilisateur fictif à qui on affecte les réservations et locations fait par le client (en remplaçant l'id du membre par celui de l'utilisateur fictif dans la table de réservation de refuges et dans la table adhérent s'il l'est) puis on supprime le membre de la table MEMBRE.

IV. Implantation des fonctionnalités (Interface):

Les fonctionnalités peuvent être visualisées dans cette interface simple et basique. On demande en premier temps au client s'il veut se connecter ou supprimer son compte en exerçant son droit à l'oubli. En cas de connexion, on affiche tous les services proposés tels que l'affichage, la réservation/location ou le rendu du matériel. Le choix est simplement fait par la saisie d'un numéro ou lettre qui convient (Une entrée de lettre en majuscule peut engendrer un problème si les lettres affichées pour le choix n'y sont pas).

Un guide n'est pas forcément nécessaire vue la simplicité de l'interface. Il faut juste respecter le format des variables demandés (pour la date dd-mm-yyyy et pour l'heure hh:mm:ss).

Le fichier à exécuter est TestClientApp.java pour lancer l'interface.

V. Bilan

Le groupe chargé de l'implémentation en Java s'est retrouvé dans une situation délicate, confronté au fait que le sous-groupe responsable de l'analyse et du schéma relationnel n'avait pas encore finalisé son travail. Cette dépendance a entraîné un blocage pour le groupe d'implémentation, car la réalisation du projet exigeait une analyse complète en amont. Face à cette contrainte, le groupe d'implémentation a pris des mesures préliminaires, comme la préparation de l'environnement de développement et la définition des interfaces, dans l'attente de l'achèvement de l'analyse. Cependant, en raison du manque de temps, le groupe d'implémentation n'a pas pu intégrer toutes les fonctionnalités optionnelles initialement envisagées. Malgré cette contrainte temporelle, une communication étroite a été maintenue entre les équipes pour résoudre rapidement les problèmes émergents. Ainsi, bien que certaines fonctionnalités optionnelles n'aient pas pu être implémentées, le groupe a réussi à mener à bien le projet principal en faisant preuve de flexibilité et en concentrant ses efforts sur les aspects prioritaires dans les délais impartis.