

Projet de Base de données

Refuge & Réservations

Benomar Sara

Lmimouni Ayman

Aroussi Meryem

Elaasri Youssef

Dehbi Yakoub

ENSIMAG

École nationale supérieure d'informatique et de mathématiques
appliquées

681, rue de la passerelle - Domaine universitaire - BP 72

38402 SAINT MARTIN D'HERES

November 29, 2023

Contents

	Page
1 Documentation du projet	2
1.1 Analyse du Problème	2
1.2 Passage au relationnel	3
2 Bilan du projet	5
2.1 Organisation	5
2.2 Gestion des Points Difficiles	6

1 Documentation du projet

1.1 Analyse du Problème

Le projet débute par l'analyse du texte expliquant les besoins du client. Notre stratégie consistait à identifier les objets nécessaires pour construire les tables de notre future base de données, en incluant les dépendances fonctionnelles entre eux ainsi que les contraintes de valeurs et de multiplicité.

L'email du refuge a été utilisé pour déterminer tous les attributs de la table "refuge". De plus, en connaissant l'objet "Repas", nous avons pu établir une relation permettant de déduire le prix associé à ce repas. L'idée d'avoir une propriété propre a commencé à prendre forme.

Il a été décidé qu'un refuge peut disposer d'au plus un numéro. En ce qui concerne les "Formations", nous avons observé qu'une formation peut proposer au moins une activité, introduisant ainsi une nouvelle contrainte de multiplicité.

Pour assurer la conformité de notre base de données au Règlement Général sur la Protection des Données (RGPD), nous avons créé un identifiant utilisateur ("IdUsr") pour préserver les traces de réservations sans avoir besoin de stocker les données clients directement. Ainsi, les données clients sont mieux protégées et peuvent être supprimées si le client en fait la demande.

Afin de bien identifier les réservations pour les refuges et les formations, des identifiants ont été introduits pour jouer le rôle de clés primaires.

Conception du Diagramme Entités/Associations (UML)

La deuxième étape a été la conception du diagramme Entités/Associations (UML). La table "Refuge" regroupe toutes les informations relatives au refuge. Afin de respecter la contrainte de multiplicité stipulant qu'un refuge peut avoir au plus un numéro, nous avons établi une association sémantique entre la table "Refuge" et la table "Ref_NumTel". Une propriété propre relie également "Refuge" à "Repas", permettant d'indiquer le prix d'un repas dans un refuge spécifique.

La table "ReservationRefuge" est naturellement liée à la table "Refuge" pour indiquer à quel refuge la réservation correspond. Elle est également liée à la table "CompteUtilisateur", permettant d'identifier à qui appartient cette réservation.

Les données utilisateur sont stockées dans une table distincte du même nom, associée à la table "CompteUtilisateur", qui ne contient que l'identifiant de l'utilisateur. Cela garantit la conformité au RGPD. De plus, étant donné qu'un utilisateur peut être adhérent ou non, la table "Adherents" est

une sous-entité faible de la table "Utilisateur".

Cette dernière est associée aux tables "ReservationFormation" et "LocationMateriel". Ces deux tables sont respectivement liées à la table "Formations" par une association sémantique, et à la table "LotMateriel" par une propriété propre déterminant le nombre de pièces réservées et le nombre cassées/perdus.

La table "LotMateriel" est associée à une table texte pour indiquer si un lot peut ou non avoir une description. Elle est également associée à une seule et unique catégorie dans la table "Categorie". Cette dernière possède une association réflexive permettant de représenter le fait qu'une catégorie mère peut avoir plusieurs sous-catégories ou aucune.

La table "Activité" est associée à "Formation" pour indiquer de quelle formation il s'agit. Elle est également associée à la table "LotMateriel", indiquant le matériel utilisé pour une certaine activité.

1.2 Passage au relationnel

Le modèle relationnel de la base de données reflète la structure définie dans le diagramme Entités/Associations (UML) et se traduit en un ensemble de tables interconnectées. Chaque table représente une entité distincte avec ses attributs, les clés primaires et les clés étrangères nécessaires pour maintenir l'intégrité des données.

1. **CompteUtilisateur** : La table **CompteUtilisateur** contient les informations relatives aux utilisateurs, avec un identifiant unique (idUsr). L'entité **Utilisateur** est associée à cette table, utilisant l'identifiant idUsr comme clé étrangère.
2. **Adherent** : L'entité **Adherent** est représentée par une table du même nom, reliant les utilisateurs adhérents via l'identifiant idUsr, qui agit comme clé primaire et clé étrangère vers **CompteUtilisateur**.
3. **Refuge** : Les informations sur les refuges sont stockées dans la table **Refuge**, avec l'email du refuge comme clé primaire. La table **Propose** associe les repas proposés par chaque refuge avec leurs prix spécifiques.
4. **ReservationRefuge** : La réservation d'un refuge par un utilisateur est enregistrée dans la table **ReservationRefuge**, avec un identifiant unique (idResRefuge). Les liens vers l'utilisateur adhérent et le refuge sont établis via les clés étrangères.
5. **Formation** : Les détails des formations sont capturés dans la table **Formation**, avec l'année et le rang comme clés primaires. La table **ReservationFormation** enregistre les réservations des formations par les adhérents.

6. Utilise : L'utilisation du matériel par une activité est modélisée par la table **Utilise**, reliant les activités aux lots de matériel spécifiques.
7. LotMateriel : Les caractéristiques des lots de matériel sont enregistrées dans la table **LotMateriel**, avec une clé primaire composée de la marque, du modèle, et de l'année. La présence d'une date de péremption est facultative.
8. LocationMateriel : Les locations de matériel effectuées par les adhérents sont enregistrées dans la table **LocationMateriel**, avec un identifiant unique (idLocationMateriel) et des références aux adhérents.
9. ReservationPieces : Les réservations spécifiques de pièces de matériel sont gérées par la table **ReservationPieces**, reliant les réservations aux lots de matériel et aux locations.
10. DatePeremption : La table **DatePeremption** stocke les dates de péremption associées aux lots de matériel, liées par la table **A_pour_datePeremption**.

Chaque table est conçue pour refléter les relations et les contraintes établies lors de la phase de conception, garantissant ainsi une représentation cohérente des données du projet.

2 Bilan du projet

2.1 Organisation

2.1.1 Analyse Approfondie et Collaboration Client

Notre démarche a débuté par une lecture attentive et approfondie du texte de description du projet, paragraphe par paragraphe. L'objectif premier était de bien appréhender les besoins du client. Pour ce faire, des échanges fréquents avec le client ont été instaurés afin de garantir que les étapes suivantes du projet répondraient de manière adéquate à ses besoins. En parallèle de ces échanges, une analyse approfondie du problème a été entreprise pour extraire les dépendances fonctionnelles et les contraintes. Il était également crucial de anticiper les étapes futures du projet pour assurer une cohérence globale. Cette phase a permis à chaque membre de l'équipe de se projeter dans le déroulement du projet.

2.1.2 Conception du Diagramme Entités/Associations (UML)

Après la conclusion du premier sprint d'analyse, nous avons entamé la conception du Diagramme Entités/Associations (UML). Cette étape a marqué une division des tâches au sein de l'équipe, avec une répartition des différentes tables à construire. Étant donné que ces tables étaient interconnectées, le travail s'effectuait simultanément, nécessitant une communication constante entre les membres pour garantir la cohérence globale du projet. Des appels réguliers à un expert en bases de données ont été réalisés pour valider la pertinence de nos choix et prendre des décisions majeures.

2.1.3 Implémentation et Développement

Le dernier sprint du projet a impliqué la construction du script SQL pour générer les tables et les remplir, ainsi que la création des fonctions JAVA pour interagir avec la base de données. Grâce à la vision commune partagée par tous les membres de l'équipe, nous avons pu diviser ces phases de développement de manière efficace. Cependant, des échanges réguliers ont été maintenus pour s'assurer de la cohérence du travail accompli. Une décision stratégique a été prise pour centraliser l'ensemble du projet sur un dépôt Git, permettant un accès fluide au travail des autres membres à tout moment. Cette approche collaborative a renforcé la coordination au sein de l'équipe et a favorisé une progression harmonieuse du projet.

2.2 Gestion des Points Difficiles

La gestion des dépendances fonctionnelles complexes entre les entités du projet s'est avérée être un défi majeur. Pour surmonter cela, nous avons opté pour une approche collaborative renforcée par l'utilisation de Git en tant que plateforme centrale. Cela a permis à chaque membre de l'équipe d'accéder en temps réel aux contributions des autres, facilitant ainsi la synchronisation des changements et assurant une vue d'ensemble cohérente du projet.

Lors de la conception du Diagramme Entités/Associations (UML), la complexité des discussions autour des choix de modélisation a été atténuée en utilisant draw.io, une plateforme collaborative de création de diagrammes. Cela a permis à chaque membre de contribuer simultanément à la conception, de partager des idées visuelles, et d'ajuster le diagramme en temps réel. Cette approche a considérablement amélioré la compréhension collective et la clarté des décisions prises.

La communication avec le client s'est parfois révélée être un défi, mais nous avons adapté notre approche en formulant des questions courtes et précises. Cela a permis d'obtenir des réponses rapides et spécifiques, facilitant ainsi la prise de décision et l'avancement du projet malgré les contraintes de communication.

En résumé, la centralisation des travaux sur Git, l'utilisation d'outils collaboratifs tels que draw.io, des questions précises lors des échanges avec le client, et une collaboration étroite pour résoudre les problèmes techniques ont été les clés du succès pour surmonter les points difficiles rencontrés durant le projet.