

MCD MERISE

MasterAgence

**Membres**

***Fayssal EL ANSARI***

***Jonathan DI MARTINO***

15/06/2023

1. **Description du sujet**

Notre application de gestion d'agence immobilière vise à simplifier et optimiser les processus de vente, de location et de suivi des biens immobiliers. Elle offre une interface conviviale et facile à utiliser pour gérer l'ensemble de vos biens, clients, contrats, visites et transactions en un seul endroit. De plus, elle permet également un suivi efficace des employés et des performances de l'agence, vous permettant ainsi de prendre des décisions éclairées et d'améliorer continuellement votre activité.

Dans le cadre de notre projet, nous utilisons et testons deux moteurs de bases de données performants, MongoDB et Couchbase, séparément afin d'évaluer leurs performances et de déterminer lequel convient le mieux à notre application. Ces deux technologies offrent une excellente flexibilité, évolutivité et sécurité pour vos données, garantissant ainsi que l'application puisse répondre aux besoins croissants de votre agence immobilière.

En choisissant notre application, vous bénéficierez d'un outil puissant et polyvalent pour gérer votre agence immobilière de manière efficace. Grâce à des fonctionnalités avancées et des moteurs de base de données de pointe, vous pourrez vous concentrer sur l'essentiel : offrir un service de qualité à vos clients et développer votre entreprise. Nous nous engageons à trouver la meilleure solution en matière de technologie de base de données en testant et en comparant les performances de MongoDB et Couchbase, afin de vous fournir une expérience utilisateur optimale.

***Les structures de l'application sont représentées par les entités suivantes :***

1. **Appartement** : Cette entité représente un bien immobilier disponible pour la réservation. Il contient des informations telles que le prix par défaut, l'emplacement, le nombre de chambres, de salles de bains, etc.
2. **Clients** : Cette entité représente un utilisateur client de la plateforme qui peut effectuer des réservations. Il contient des informations comme l'email, le mot de passe, le numéro de téléphone, le pays d'origine, et l'état de vérification du compte.
3. **Réservation** : Cette entité représente une réservation effectuée par un client pour un appartement à une date spécifique. Elle contient des informations comme l'ID de l'appartement, l'ID de l'utilisateur, la date de début et de fin de la réservation, et le prix de la réservation.
4. **Transaction** : Cette entité représente une transaction financière associée à une réservation. Elle contient des informations telles que l'ID du client, l'ID de la réservation, le montant, le type de transaction (débit ou crédit), le statut de la transaction, etc.
5. **Utilisateur** : Cette entité représente un utilisateur de la plateforme (qui peut être un administrateur, un gestionnaire immobilier, etc.) Elle contient des informations comme le nom, l'email et le mot de passe.

***Les services proposés de l’application***

Les services utilisant la requête N1QL sera développer dans la partie Couchbase sinon les autres services seront développés dans les deux Moteurs MongoDB et Couchbase.

* **Recherche d'appartements par utilisateur** : Ce service pourrait prendre un ID utilisateur comme paramètre, puis utiliser le service Apartments pour rechercher tous les appartements qui appartiennent à cet utilisateur.
* **Recherche de transactions par client** : Ce service pourrait prendre un ID client comme paramètre, puis utiliser le service Transactions pour rechercher toutes les transactions qui ont été effectuées par ce client.
* **Recherche de réservations par appartement** : Ce service pourrait prendre un ID d'appartement comme paramètre, puis utiliser le service Reservations pour trouver toutes les réservations qui ont été faites pour cet appartement.
* **Recherche d'appartements par nombre de chambres** : Ce service pourrait prendre un nombre de chambres comme paramètre, puis utiliser le service Apartments pour trouver tous les appartements qui ont ce nombre de chambres.
* **Recherche de clients par pays d'origine** : Ce service pourrait prendre un pays comme paramètre, puis utiliser le service Clients pour trouver tous les clients qui viennent de ce pays.
* **Trouver tous les appartements dans une certaine gamme de prix** : Ce service peut utiliser une requête N1QL pour récupérer tous les appartements dont le prix par défaut se situe dans une certaine plage de prix.
* **Trouver toutes les transactions d'un client spécifique dans une certaine plage de dates** : Ce service peut utiliser une requête N1QL pour récupérer toutes les transactions d'un client spécifique qui ont eu lieu entre deux dates spécifiques.
* **Trouver tous les utilisateurs qui n'ont pas encore vérifié leur compte** : Ce service peut utiliser une requête N1QL pour récupérer tous les utilisateurs qui n'ont pas encore vérifié leur compte.
* **Trouver toutes les réservations pour un appartement spécifique** : Ce service peut utiliser une requête N1QL pour récupérer toutes les réservations pour un appartement spécifique.
* **Trouver tous les clients d'un pays d'origine spécifique** : Ce service peut utiliser une requête N1QL pour récupérer tous les clients d'un pays d'origine spécifique.
* **Trouver tous les appartements ayant un certain nombre de salles de bains** : Ce service peut utiliser une requête N1QL pour récupérer tous les appartements ayant un certain nombre de salles de bains.

1. **Dictionnaire de données MERISE**

***Entité Apartment***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Propriété** | **Description** | **Format des données** | **Type** | **Identifiant** | **Contraintes** |
| **\_id** | **Identifiant unique de l'appartement** | **String (24 caractères hexadécimaux)** | **ObjectId** | **Oui** | **Doit être unique** |
| **title** | **Titre de l'appartement** | **String** | **String** | **Non** | **Aucune spécifique mentionnée** |
| **description** | **Description détaillée de l'appartement** | **String** | **String** | **Non** | **Aucune spécifique mentionnée** |
| **defaultPrice** | **Prix par défaut de l'appartement** | **Float** | **Float** | **Non** | **Doit être un nombre positif** |
| **numberOfRooms** | **Nombre de pièces dans l'appartement** | **Integer** | **Integer** | **Non** | **Doit être un nombre positif** |
| **numberOfBathrooms** | **Nombre de salles de bains dans l'appartement** | **Integer** | **Integer** | **Non** | **Doit être un nombre positif** |
| **area** | **Surface de l'appartement en mètres carrés** | **Float** | **Float** | **Non** | **Doit être un nombre positif** |
| **address** | **Adresse de l'appartement** | **String** | **String** | **Non** | **Aucune spécifique mentionnée** |
| **location** | **Coordonnées géographiques de l'appartement** | **Latitude/Longitude** | **Float** | **Non** | **Doit être des valeurs valides de latitude/longitude** |
| **ownerId** | **Identifiant du propriétaire de l'appartement** | **String** | **String** | **Non** | **Aucune spécifique mentionnée** |
| **images** | **Liste des liens vers les images de l'appartement** | **Array of Strings** | **Array** | **Non** | **Aucune spécifique mentionnée** |
| **rates** | **Liste des tarifs de l'appartement** | **Array of Objects** | **Array** | **Non** | **Aucune spécifique mentionnée** |
| **createdAt** | **Date et heure de création de l'entrée de l'appartement** | **Date** | **Date** | **Non** | **Doit être une date valide** |
| **updatedAt** | **Date et heure de la dernière mise à jour de l'entrée de l'appartement** | **Date** | **Date** | **Non** | **Doit être une date valide** |
| **\_class** | **Nom de la classe de l'entité** | **String** | **String** | **Non** | **Aucune spécifique mentionnée** |

***Entité Reservation***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Propriété | Description | Format des données | Type | Identifiant | Contraintes |
| \_id | Identifiant unique de la réservation | String (24 caractères hexadécimaux) | ObjectId | Oui | Doit être unique |
| apartmentId | Identifiant de l'appartement réservé | String | String | Non | Doit correspondre à un appartement existant |
| userId | Identifiant de l'utilisateur qui a fait la réservation | String | String | Non | Doit correspondre à un utilisateur existant |
| startDate | Date de début de la réservation | Date | Date | Non | Doit être une date valide |
| endDate | Date de fin de la réservation | Date | Date | Non | Doit être une date valide |
| price | Prix de la réservation | Float | Float | Non | Doit être un nombre positif |
| \_class | Nom de la classe de l'entité | String | String | Non | Aucune spécifique mentionnée |

***Entité Client***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Propriété | Description | Format des données | Type | Identifiant | Contraintes |
| \_id | Identifiant unique du client | String (24 caractères hexadécimaux) | ObjectId | Oui | Doit être unique |
| email | Email du client | String | String | Non | Doit être une adresse email valide |
| password | Mot de passe du client | String | String | Non | Aucune spécifique mentionnée |
| phoneNumber | Numéro de téléphone du client | String | String | Non | Aucune spécifique mentionnée |
| countryOfOrigin | Pays d'origine du client | String | String | Non | Aucune spécifique mentionnée |
| isVerified | Indicateur si le client est vérifié ou non | Boolean | Boolean | Non | Aucune spécifique mentionnée |
| profilePicture | URL de la photo de profil du client | String | String | Non | Doit être une URL valide |
| reservations | Liste des réservations faites par le client | Array of Strings | Array | Non | Aucune spécifique mentionnée |
| transactions | Liste des transactions faites par le client | Array of Strings | Array | Non | Aucune spécifique mentionnée |
| \_class | Nom de la classe de l'entité | String | String | Non | Aucune spécifique mentionnée |

***Entité Transaction***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Propriété | Description | Format des données | Type | Identifiant | Contraintes |
| \_id | Identifiant unique de la transaction | String (24 caractères hexadécimaux) | ObjectId | Oui | Doit être unique |
| clientId | Identifiant du client qui a fait la transaction | String | String | Non | Doit correspondre à un client existant |
| reservationId | Identifiant de la réservation liée à la transaction | String | String | Non | Doit correspondre à une réservation existante |
| amount | Montant de la transaction | Integer | Integer | Non | Doit être un nombre positif |
| currency | Devise de la transaction | String | String | Non | Aucune spécifique mentionnée |
| bankName | Nom de la banque effectuant la transaction | String | String | Non | Aucune spécifique mentionnée |
| accountNumber | Numéro de compte bancaire du client | String | String | Non | Aucune spécifique mentionnée |
| transactionType | Type de transaction (par exemple, CREDIT) | String | String | Non | Aucune spécifique mentionnée |
| transactionMode | Mode de transaction (par exemple, NET\_BANKING) | String | String | Non | Aucune spécifique mentionnée |
| transactionStatus | Statut de la transaction (par exemple, PENDING) | String | String | Non | Aucune spécifique mentionnée |
| transactionDateTime | Date et heure de la transaction | Date | Date | Non | Doit être une date valide |
| \_class | Nom de la classe de l'entité | String | String | Non | Aucune spécifique mentionnée |

***Entité Users***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Propriété | Description | Format des données | Type | Identifiant | Contraintes |
| \_id | Identifiant unique de l'utilisateur | String (24 caractères hexadécimaux) | ObjectId | Oui | Doit être unique |
| name | Nom de l'utilisateur | String | String | Non | Aucune spécifique mentionnée |
| email | Email de l'utilisateur | String | String | Non | Doit être une adresse email valide |
| password | Mot de passe de l'utilisateur | String | String | Non | Aucune spécifique mentionnée |
| \_class | Nom de la classe de l'entité | String | String | Non | Aucune spécifique mentionnée |

1. **Description textuelle des associations**

* Un Client peut avoir plusieurs Reservation, chaque Reservation appartient à un Client spécifique.
* Un Client peut avoir plusieurs Transaction, chaque Transaction appartient à un Client spécifique.
* Une Reservation est associée à une Transaction spécifique et inversement.
* Un User peut-être associé à plusieurs Reservation (supposant que User et Client sont interchangeables ici), chaque Reservation est faite par un User spécifique.

1. **Définition du modèle Entité-Association MERISE**

