PROGRAMMATION EN JAVA

**PROJET – RAPPORT 2**

Nom Du Projet

**Bon-Esprit**

Nombre de la group: **03**

Membres:

1) DANG DIEM LINH 18126023

2) LUONG LONG HA 18126014

3) DUONG QUANG VINH 18126038

**Tableau de contenu**

[**L’HISTOIRE DE VERSIONS** 3](#_Toc68302579)

[**INTRODUCTION** 4](#_Toc68302580)

[**ANALYSE ET CONCEPTION** 6](#_Toc68302581)

[**IMPLÉMENTATION** 11](#_Toc68302582)

[**EXEMPLES DE DONNÉES** 12](#_Toc68302583)

[**RÉSULTAT** 13](#_Toc68302584)

[**PLAN** 14](#_Toc68302585)

[**RÉFÉRENCES** 15](#_Toc68302586)

# **L’HISTOIRE DE VERSIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| N.o | Version | La date de création | Description |
| 1 | **Bon-Esprit 01** | 10/3/2021 | Le code qui implémente les classes et les associations entre les classes dans l'application. Cette version est **une démo incomplète** du comportement des classes. |
| 2 | **Bon-Esprit 02** | 3/4/2021 | + **Base de données** supplémentaire pour l'application: les classes commencent à se lier avec la base de données.  + Ajuster et ajouter un certain nombre de méthodes des classes. |
| 3 | **Bon-Esprit 03** |  | Déployer du code sur **l'interface à l'aide de Swing.** Cette version se traduira par une application relativement complète qui peut être utilisée au niveau de l'interface utilisateur graphique. |

# **INTRODUCTION**

1. **LE BUT DU RAPPORT**

Tout site Web ou application nécessite une prise en charge de base de données. Ce rapport se concentrera sur la présentation de **la base de données** de l'application, y compris les objets suivants:

- **Articles** sur les problèmes de santé mentale.

- Le **quiz** pour tester la santé mentale de l'utilisateur.

- **Utilisateurs** généralistes et experts (thérapeutes),…

À la fois, notre rapport montre également la **connexion de la base de données** à l'application et ces données seront utilisées pour faire les **fonctions** de l'application telles que:

- **Chercher** de l'information.

- **Trier** / **Filtre** les informations.

- Quelques **fonctions statistiques** pour les utilisateurs experts liées aux vues, au nombre d'articles ...

- **Ajouter** - **modifier** - **supprimer** les articles. les utilisateurs,…

- Faire le **quiz** et répondre le score du test…

- Se **connecter** / Se **déconnecter**…

1. **LA MÉTHODE DE STOCKAGE DES DONNÉES DE BonEsprit**

Les données d'application sont stockées et organisées dans une base de données à l'aide du système d'administration **Microsoft SQL Server**. Les raisons pour choisir cette SGBD sont:

1. **L'installation est simplifiée:**

La complexité de l'installation du logiciel est considérablement minimisée grâce à l'installation automatique des mises à jour. Il peut être installé via un assistant de configuration et les mises à jour prérequises sont détectées et téléchargées automatiquement par l'installateur.

1. **Les fonctionnalités de sécurité sont meilleures**

Il utilise les stratégies de sécurité qui permet uniquement l'accès du personnel autorisé à la base de données. Les audits et événements de sécurité peuvent être écrits automatiquement dans les fichiers journaux *(log file)*.

1. **Performance améliorée**

Le serveur MS SQL dispose d'une fonction de compression de données transparente intégrée ainsi que d'un cryptage. Le serveur MS SQL dispose d'un contrôle d'accès couplé à des outils de gestion des autorisations efficaces. De plus, il offre des performances améliorées en matière de collecte de données.

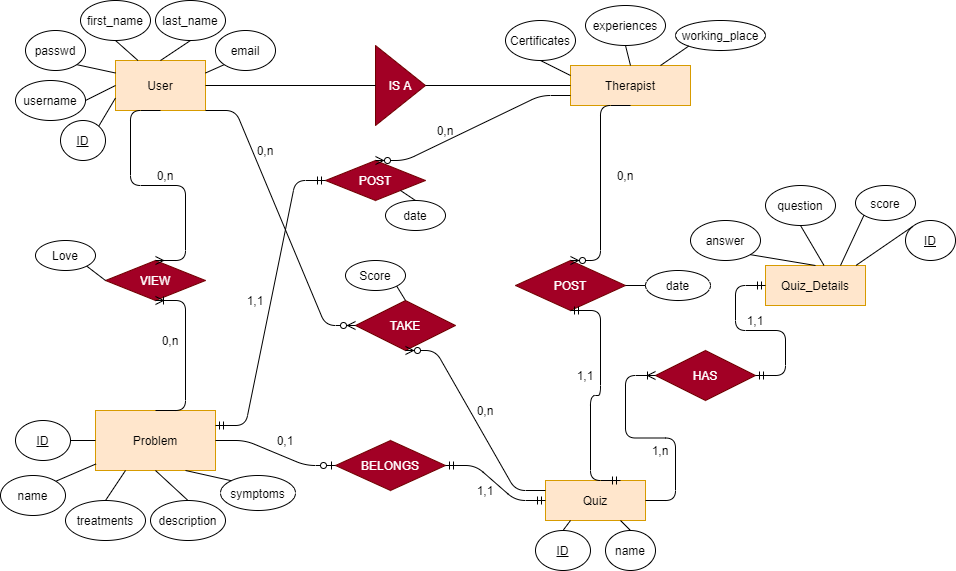
1. **Garantir la disponibilité et la récupérabilité des données**

Le serveur SQL comprend des outils efficaces de gestion des données et d'exploration de données, ainsi que le partitionnement de disque. La maintenance optimale de votre serveur peut être assurée en suivant des pratiques efficaces de gestion des données.

# **ANALYSE ET CONCEPTION**

1. **ER DIAGRAMME *(Entity relation)***

Voici le schéma ER de l'application BonEsprit *(voir le fichier* ***ER.png*** *pour plus claire):*



L'application a un total de 5 entités:

**1) USER:** l'entité représentant l'utilisateur de l'application, comprenant les informations suivantes:

- **ID**: unique et généré automatiquement lors de l'ajout d'un utilisateur.

- **Username**: nom d'utilisateur, unique.

- **Password**: le mot de passe du compte.

- **First\_name**: le nom de l'utilisateur

- **Last\_name**: le nom de famille de l'utilisateur

- **Email**: email de l'utilisateur, unique.

**2) THERAPIST:** experts dans le domaine de la psychologie, sont le type d'entités **héritées** de l'entité **USER**, avec les propriétés supplémentaires suivantes:

**- Certificates:** liste des certificats de thérapeute.

**- Experience:** années d'expérience.

**- Working\_place:** adresse du lieu de travail.

**3) PROBLEM**: l'entité qui représente des **articles** spécifiques pour les problèmes de santé psychologique et mentale, y compris les attributs suivants:

- **ID**: code de la poste, unique.

- **Name**: nom de l'article.

- **Description**: une brève description de l'article, moins de 500 caractères.

- **Symptoms**: symptômes du problème.

- **Treaments**: les traitements pour le problème.

**4) QUIZ:** l’entité qui représente le test pour vérifier certains symptômes indiquant la probabilité que l'utilisateur souffre de certaines maladies mentales. Cette entité comprend les propriétés suivantes:

- **ID**: code du quiz, unique.

- **Nom**: nom du quiz

**5) QUIZ\_DETAIL:** L'entité représente les questions de chaque quiz, y compris des attributs tels que:

**- ID:** code de la question, unique

**- question:** le contenu de la question

**- réponses:** liste de 4 réponses

**- score\_answers:** la liste comprend 4 points, correspondant à 4 réponses

**- score:** score total du test.

**- quiz:** le code du quiz auquel appartient la question.

1. **DÉTAILS DES RELATIONS DES ENTITÉS**

**PROBLEM ⬄ USER:**

* + Un article publié par un thérapeute. Un thérapeute peut publier plusieurs articles.
  + L'utilisateur peut afficher plusieurs articles. Un article peut être consulté par plusieurs utilisateurs.

**PROBLEM ⬄ QUIZ:**

* + Un article a 0 ou 1 quiz. Un quiz n'appartient qu'à un seul article.

**QUIZ ⬄ USER:**

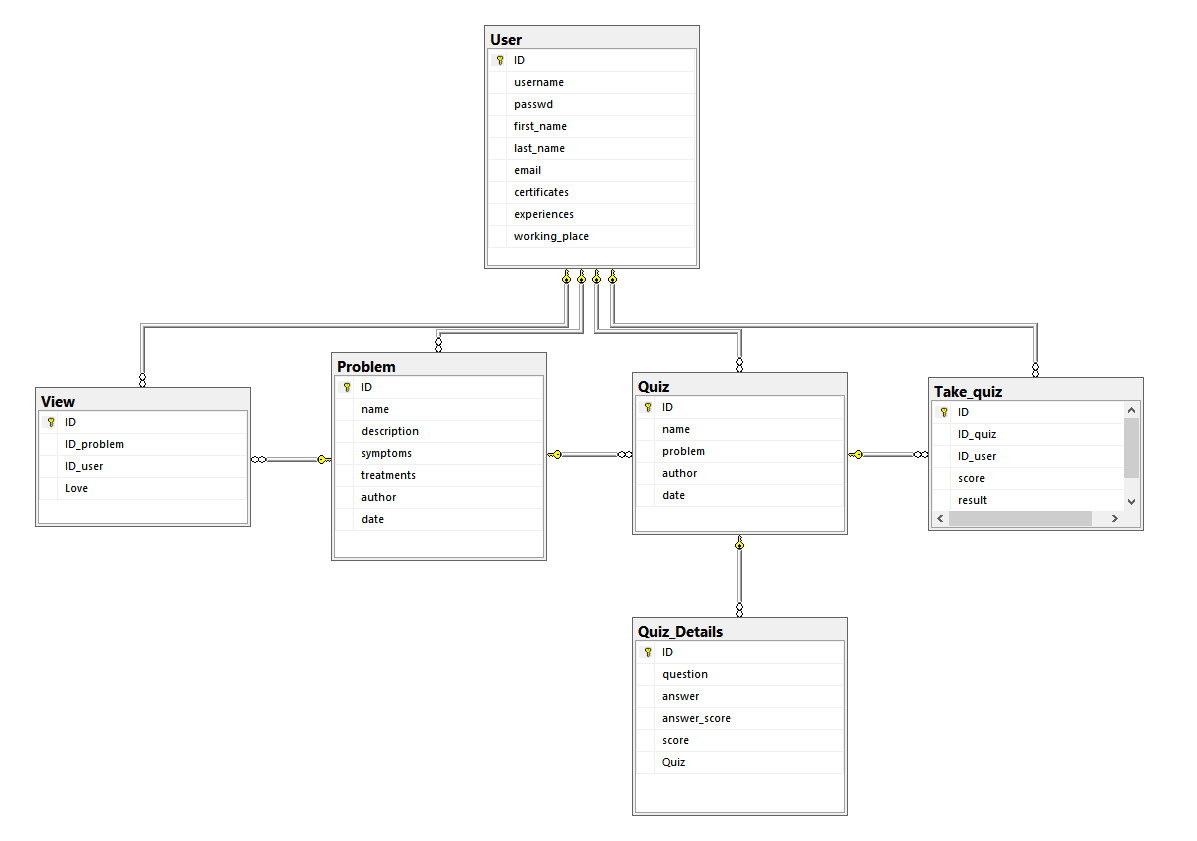
* + Un QUIZ a été publié par un seul thérapeute. Un thérapeute peut publier plusieurs QUIZ.
  + L'utilisateur peut tester de plusieurs quiz. Un quiz peut être réalisé par plusieurs utilisateurs.

**QUIZ ⬄ QUIZE\_DETAILS:**

* + Un QUIZ peut avoir plusieurs Quiz\_details. Chaque Quiz\_detail n'appartient qu'à un seul quiz.

1. **LE DIAGRAMME DE RÉLATION**

À partir du schéma ER, le schéma relationnel *(voir le fichier* ***DR.png*** *pour plus claire)* est reconstruit comme suit:



La plupart des entités sont reconstruites sous une forme tabulaire. Plusieurs relations plusieurs-à-plusieurs forment une nouvelle table de:

- Tableau "**VIEW**": montre les messages consultés par n'importe quel utilisateur, est le favori de l'utilisateur ou non.

- Tableau "**Take\_quiz**": indiquant les utilisateurs auxquels le quiz a été répondu, le score ainsi que les résultats du quiz pour l'utilisateur.

1. **CONTRAINTES D'INTÉGRITÉ**

**Tableau “USER”:**

- ID, le nom d'utilisateur doit être unique.

- Si c’est un utilisateur régulier, laissez la section du certificat, les années d'expérience et l'adresse de travail vides.

- Les années d'expérience doivent être un nombre entier positif.

**Tableau “PROBLEM”:**

- L'ID est unique.

- La description ne doit pas dépasser 500 caractères.

**Tableau QUIZ:**

- L'ID est unique.

- date: la date comptable par défaut est la date du jour.

**Tableau Quiz\_Details:**

- L'ID est unique

- le score est un entier positif.

**Tableau “VIEW”**

- L'ID est unique.

- “Love” a 2 valeurs: 1 - Amour, 0 - Pas d'amour.

**Tableau Take\_quiz:**

- L'ID est unique.

- Score: est un entier positif.

- Result: n'a que 3 valeurs: "DANGER", "MAYBE", "NORMAL"

*L'installation détaillée de la base de données se trouve dans le fichier* ***database\_script.sql*** *attaché.*

# **IMPLÉMENTATION**

1. **Connectez l'application à la base de données**
2. **Configurer la fonction de recherche**
3. **Configurer la fonction de tri**
4. **Configurer la fonction pour faire le quiz**
5. **Configurer la fonction voir l'article**

# **EXEMPLES DE DONNÉES**

1. **USER**
2. **PROBLEM**
3. **QUIZ**
4. **QUIZ\_DETAILS**
5. **VIEW**
6. **TAKE\_QUIZ**

# **RÉSULTAT**

|  |  |
| --- | --- |
| **AVANTAGES** | **DÉSAVANTAGES** |
| L'utilisation de systèmes de gestion de bases de données supplémentaires tels que MS SQL permet d'optimiser l'accès aux données et le contrôle de l'application facilement, intuitivement et rapidement. | Les performances sont encore assez limitées en n'utilisant que l'écran de la console pour tester le code => difficile à imaginer. Par conséquent, il est nécessaire d'utiliser une technologie liée à l'interface pour réaliser du code sur l'application. |
|  | Certaines fonctions n'ont pas d'algorithme à appliquer spécifiquement. |

# **PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| **N.o** | **Tâche** | **Qui fait?** | **Deadline** |
| 1 | Construire le package **PROBLEM** | Linh | 12/3/2021 |
| 2 | Construire le package **USERS** | Vinh | 12/3/2021 |
| 3 | Dessiner des diagrammes de classes et de packages | Hà | 7/3/2021 |
| 4 | Préparer les données brutes pour le problème | Vinh + Hà | 3/4/2021 |
| 5 | Ajouter les données à la base de données et importer dans le code | Linh | 3/4/2021 |
| 6 | Affiner les algorithmes de recherche de classe. Démo de la recherche. | Linh | 10/4/2021 |
| 7 | Ajoutez les méthodes manquantes aux classes. | Vinh + Hà | 10/4/2021 |
| 8 | Construire des interfaces Swing pour les problèmes | Vinh |  |
| 9 | Concrétiser les fonctions utilisateur telles que la connexion, la déconnexion, l'ajout, la suppression et l'édition de messages ... sur l'interface de l'application | Linh |  |
| 10 | Tester la fonctionnalité de l'application et faire le rapport | Hà |  |
| 11 | Modifiez les fonctions en fonction du rapport de test. | Linh + Vinh |  |
| 12 | Compléter l'application et faire le rapport | Linh |  |

# **RÉFÉRENCES**

1. **tutorialspoint.com:** définit correctement les classes et les méthodes, en fonction du contexte.
2. **Autres algorithmes:** auto-développés par l'équipe.
3. **tutorialspoint.com/sql/**: Appliquer la syntaxe de l'instruction pour installer toute la base de données de l'application.