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](#_Toc66397204)

[INTRODUCTION 4](#_Toc66397205)

[ANALYSE ET CONCEPTION 6](#_Toc66397206)

[IMPLÉMENTATION 11](#_Toc66397207)

[RÉSULTAT 14](#_Toc66397208)

[PLAN 15](#_Toc66397209)

[RÉFÉRENCES 16](#_Toc66397210)

# **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.

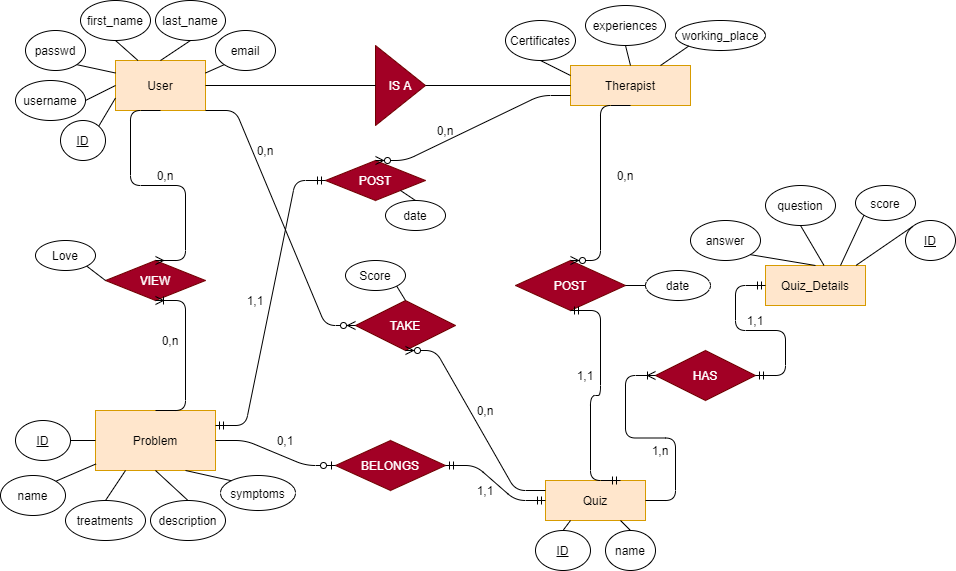
1. **Garantir la disponibilité et la récupérabilité des données**

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 de détails):*



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

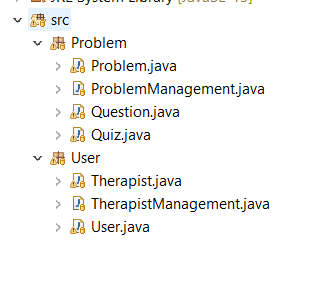
**- score:** score total du test.

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

1. **LE DIAGRAMME DE RÉLATION**

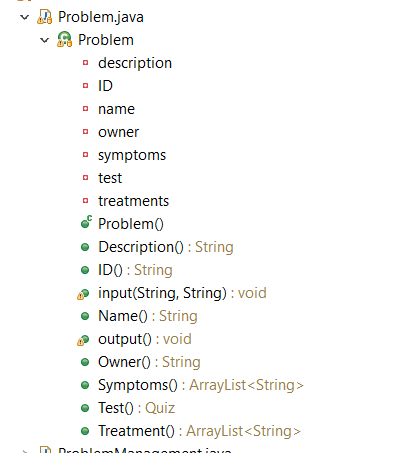
# **IMPLÉMENTATION**

La structure de l’application en JAVA:

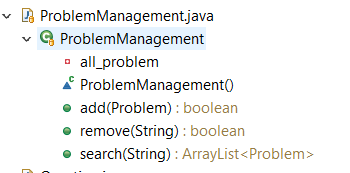


Pour chaque classe:

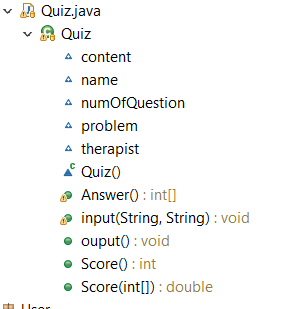
1. **PROBLEM:**



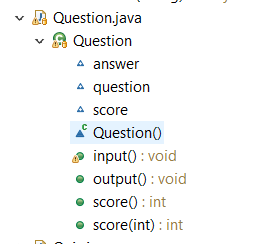
1. **ProblemManagement**



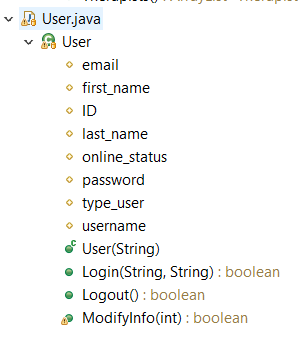
1. **Quiz**



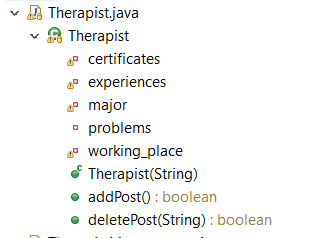
1. **Question**



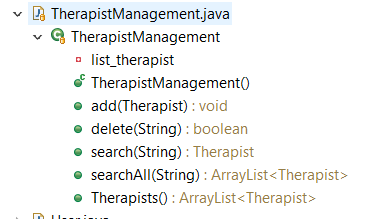
1. **User**



1. **Therapist**



1. **TherapistManagement**



# 

# **RÉSULTAT**

Jusqu'à présent, l'application a construit des classes relativement complètes avec les méthodes nécessaires. Cependant, le groupe a rencontré quelques difficultés:

- Aucun algorithme n'a été trouvé pour ajuster la chaîne de recherche en petits morceaux, servant la fonction "recherche de symptômes". L'équipe a l'intention de séparer les mots et d'utiliser des statistiques pour résoudre ce problème.

- L'ajout d'objets tels que les messages, les utilisateurs ... n'est pas enregistré après la fin du programme => besoin d'utiliser la base de données pour corriger.

# **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 | 21/3/2021 |
| 5 | Ajouter les données à la base de données et importer dans le code | Linh | 21/3/2021 |
| 6 | Affiner les algorithmes de recherche de classe. Démo de la recherche. | Linh | 27/3/2021 |
| 7 | Ajoutez les méthodes manquantes aux classes. | Vinh + Hà | 27/3/2021 |
| 8 | Construire des interfaces Swing pour les problèmes | Vinh | 03/04/2021 |
| 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 | 10/4/2021 |
| 10 | Tester la fonctionnalité de l'application et faire le rapport | Hà | 17/4/2021 |
| 11 | Modifiez les fonctions en fonction du rapport de test. | Linh + Vinh | 20/4/2021 |
| 12 | Compléter l'application et faire le rapport | Linh | 24/4/2021 |

# 

# **RÉFÉRENCES**

1. **tutorialspoint.com:** définit correctement les classes et les méthodes, en fonction du contexte.
2. **Autres algorithmes:** auto-développés par l'équipe.