|  |
| --- |
| **BTS Services informatiques aux organisations SESSION 2024**  **ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)**  **Épreuve E5 - Conception et développement d’applications (option SLAM)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPTION D’UNE RÉALISATION PROFESSIONNELLE** | | | **N° réalisation :** |
| **Nom, prénom : Vadenne Johan** | | **N° candidat :** | |
| **Épreuve ponctuelle** | **Contrôle en cours de formation** | **Date :** ...... / ...... /............ | |
| **Organisation support de la réalisation professionnelle**  Entreprise VAD (Visual Application Designer) | | | |
| **Intitulé de la réalisation professionnelle**  Nom du logiciel : Gestion Message  Type : Lourde  Environnement : Windows | | | |
| **Période de réalisation :** 04/03/2024 – 23/03/2024 **Lieu :** La Rochelle  **Modalité :  Seul(e)  En équipe** | | | |
| **Compétences travaillées**  Concevoir et développer une solution applicative  Assurer la maintenance corrective ou évolutive d’une solution applicative  Gérer les données | | | |
| **Conditions de réalisation[[1]](#footnote-1) (ressources fournies, résultats attendus)**  Description : Gestion Message est une application conçue pour éditer des messages destinés à être utilisés dans d'autres applications. Chaque message est composé d'un code à 4 caractères (par exemple : 0245, 5210, 0010), d'un contenu (par exemple : "Connexion interrompue", "Authentification échouée"), d'un groupe (par exemple : base de données, API) et d'un type (par exemple : ALERTE, QUESTION, ERREUR).  Objectif :  L'objectif de Gestion Message est de simplifier la gestion des messages en évitant la réécriture répétitive de ceux-ci, ce qui peut entraîner une surcharge de code. Cette tâche incombe souvent uniquement au développeur et peut nécessiter la recompilation de l'application ou sa mise en ligne pour corriger un simple message. Gestion Message offre une solution en fournissant une interface utilisateur graphique (GUI) permettant à l'utilisateur ainsi qu'au client de collaborer efficacement pour éditer et personnaliser les messages. Une fois les messages édités, il suffit de les intégrer dans le code et de les appeler simplement en utilisant leur code correspondant. | | | |
| **Description des ressources documentaires, matérielles et logicielles utilisées[[2]](#footnote-2)**  Ordinateur :  PC fixe Fénelon Notre Dame  PC portable Asus Personnelle  Logiciel :  Visual Studio  Looping  draw io  DB Browser (SQLite)  Documentation :  https://visualstudio.microsoft.com/fr/ https://chat.openai.com/ | | | |
| **Modalités d’accès aux productions[[3]](#footnote-3) et à leur documentation[[4]](#footnote-4)** UML BDD Une image contenant texte, capture d’écran, ligne, diagramme UML Classe Une image contenant texte, capture d’écran, diagramme, Police  Description générée automatiquement Schéma environnement informatique Une image contenant texte, capture d’écran, diagramme, conception  Description générée automatiquement  Fenêtre  Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement | | | |

|  |
| --- |
| **BTS Services informatiques aux organisations SESSION 2024**  **ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle**  **(verso, éventuellement pages suivantes)**  **Épreuve E5 - Conception et développement d’applications (option SLAM)** |

|  |
| --- |
| **Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs** Authentification Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement   Code authentification dans Cl\_Utilisateur public bool utilisateurConnexion()  {  try  {  if (!ValeurCorrecte()) { return false; }  // Création de la requette SQL  string RequeteSQL = """  SELECT  IdUtilisateur  FROM  T\_Utilisateur  WHERE  NomUtilisateur = @NomUtilisateur  AND MotDePasse = @MotDePasse;  """;  SQLiteCommand CommandSQLite = new SQLiteCommand(RequeteSQL, this.MaConnexion); // création de la commande SQLite  // Ajout des paramètres a la requête préparer  CommandSQLite.Parameters.AddWithValue("@NomUtilisateur", HachSHA256(\_NomUtilisateur));  CommandSQLite.Parameters.AddWithValue("@MotDePasse", HachSHA256(\_MotDePasse));  this.MaConnexion.Open(); // ouvre la connexion à la base de données  SQLiteDataReader LectureRequete = CommandSQLite.ExecuteReader(); // Exécute la commande en mode lecture  int IdUtilisateurLu = 0;  // récupère l'IdUtilisateur  while (LectureRequete.Read())  {  IdUtilisateurLu = LectureRequete.GetInt32(0);  }  LectureRequete.Close(); // ferme la lecture de la requête  this.MaConnexion.Close(); // ferme la connexion à la base de données  // si l'utilisateur existe on retourn vrai, sinon faux  if (IdUtilisateurLu > 0)  {  return true;  }  else  {  return false;  }  }  catch  {  Cl\_AfficheMessageBox.MessageAlerte("Une erreur s'est produite. Veuillez contacter les développeurs.\nCode erreur 002");  return false;  }  } Sécurité Hachage sha256  Requête préparer  Vérification des valeurs  Exception gérée Insert, Update, Delete Type message     //  // override insert  //  public override void Insert()  {  try  {  if (ValeurCorrecte()) // Vérifie si toutes les données sont bien normées  {  // création de la requête INSERT  string RequeteSQL = """  INSERT INTO T\_TypeMessage  (LabelTypeMessage) VALUES(@LabelTypeMessage);  """;  SQLiteCommand CommandSQLite = new SQLiteCommand(RequeteSQL, this.MaConnexion); // création de la commande SQLite  // Ajout des paramètres a la requête préparer  CommandSQLite.Parameters.AddWithValue("@LabelTypeMessage", LabelTypeMessage);  this.MaConnexion.Open(); // ouvre la connexion à la base de données  CommandSQLite.ExecuteNonQuery(); // Exécute la commande INSERT  this.MaConnexion.Close(); // ferme la connexion à la base de données  }  }  catch  {  Cl\_AfficheMessageBox.MessageAlerte("Une erreur s'est produite. Veuillez contacter les développeurs.\nCode erreur 006");  }  }  //  // override update  //  public override void Update()  {  try  {  if (ValeurCorrecte()) // Vérifie si toutes les données sont bien normées  {  // création de la requête UPDATE  string RequeteSQL = """  UPDATE T\_TypeMessage  SET LabelTypeMessage = @LabelTypeMessage  WHERE IdTypeMessage = @IdTypeMessage;  """;  SQLiteCommand CommandSQLite = new SQLiteCommand(RequeteSQL, this.MaConnexion); // création de la commande SQLite  // Ajout des paramètres a la requête préparer  CommandSQLite.Parameters.AddWithValue("@LabelTypeMessage", LabelTypeMessage);  CommandSQLite.Parameters.AddWithValue("@IdTypeMessage", IdTypeMessage);  this.MaConnexion.Open(); // ouvre la connexion à la base de données  CommandSQLite.ExecuteNonQuery(); // Exécute la commande UPDATE  this.MaConnexion.Close(); // ferme la connexion à la base de données  }  }  catch  {  Cl\_AfficheMessageBox.MessageAlerte("Une erreur s'est produite. Veuillez contacter les développeurs.\nCode erreur 007");  }  }  //  // override delete  //  public override void Delete()  {  try  {  if (IdTypeMessage <= 0)  {  Cl\_AfficheMessageBox.MessageAlerte("Il n'y a aucun type message sélectionné!");  return;  }  // création de la requête DELETE  string RequeteSQL = """  DELETE FROM T\_TypeMessage  WHERE IdTypeMessage = @IdTypeMessage;  """;  SQLiteCommand CommandSQLite = new SQLiteCommand(RequeteSQL, this.MaConnexion); // création de la commande SQLite  // Ajout des paramètres a la requête préparer  CommandSQLite.Parameters.AddWithValue("@IdTypeMessage", IdTypeMessage);  this.MaConnexion.Open(); // ouvre la connexion à la base de données  CommandSQLite.ExecuteNonQuery(); // Exécute la commande DELETE  this.MaConnexion.Close(); // ferme la connexion à la base de données  }  catch  {  Cl\_AfficheMessageBox.MessageAlerte("Une erreur s'est produite. Veuillez contacter les développeurs.\nCode erreur 008");  }  } Sécurité Requête préparer  Vérification des valeurs  Exception gérée |

1. En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO. [↑](#footnote-ref-1)
2. Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l’annexe II.E du référentiel du BTS SIO. [↑](#footnote-ref-2)
3. Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l’épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d’organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l’épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d’un espace de stockage et de la présentation de l’organisation du stockage. [↑](#footnote-ref-3)
4. Lien vers la documentation complète, précisant et décrivant, si cela n’a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données. [↑](#footnote-ref-4)