Documentation technique module 3

**24.06.2025 | EST Porrentruy**

M1 Communication

M3 Informatique industrielle

Candidats : B. Hofer & L.Domon

Responsable de filière : M. C.Cailler

Responsable du module 1 : M. W.Grüter

Responsable du module 3 : M.D. Montavon

Tables des matières

[1 Introduction 3](#_Toc199856568)

[1.1 Fonctionnalités principales 3](#_Toc199856569)

[1.2 Technologies utilisées 3](#_Toc199856570)

[1.3 Structure du projet : 4](#_Toc199856571)

[2 Base de données 5](#_Toc199856572)

[2.1 Schéma de la base de données 5](#_Toc199856573)

[2.1.1 MCD 5](#_Toc199856574)

[2.1.2 MLD 5](#_Toc199856575)

[2.2 Configuration de la connexion 6](#_Toc199856576)

[3 Interface utilisateur 7](#_Toc199856577)

[3.1 Manager (principal) 7](#_Toc199856578)

[3.2 RecetteCreation 8](#_Toc199856579)

[3.3 RecetteInformation 9](#_Toc199856580)

[3.4 Composant réutilisable 9](#_Toc199856581)

[4 Accès aux données 10](#_Toc199856582)

[4.1 DatabaseManager 10](#_Toc199856583)

[4.2 databaseService 10](#_Toc199856584)

[4.2.1 Méthode de récupération 10](#_Toc199856585)

[4.2.2 Méthodes d’insertion 10](#_Toc199856586)

[4.2.3 Méthodes de mise à jour 11](#_Toc199856587)

[4.2.4 Méthode de suppression 11](#_Toc199856588)

[5 Gestion des formulaires 12](#_Toc199856589)

[5.1 FormManager 12](#_Toc199856590)

[5.2 Global 12](#_Toc199856591)

[6 Dépendances 13](#_Toc199856592)

[6.1 Packages NuGet 13](#_Toc199856593)

[7 Annexe 13](#_Toc199856594)

[7.1 Github 13](#_Toc199856595)

[7.2 Documentation généré 13](#_Toc199856596)

# Introduction

Ce document contient la documentation technique du projet réaliser dans le cadre du module 3. Cette application permet la gestion de recette et de leurs opérations et de la gestion de lots avec traçabilité de ceux-ci. L’application communique avec une base de données, ensuite l’automate utilise cette base de données pour effectuer ces lots.

## Fonctionnalités principales

* Gestion des recettes
* Gestion des lots
* Gestion des opérations
* Historique des événements
* Interface de recherche

## Technologies utilisées

* Langage : C#
* Framework : .NET Frameword 4.7.2
* Interface : Windows Forms
* Base de données : MySQL

## Structure du projet :

**M3 Pratique**/

├── **Config**/

│ ├── App.config # Configuration de l'application

│ └── packages.config # Packages NuGet

├── **Data**/

│ ├── DatabaseManager.cs # Gestion des connexions

│ └── DatabaseService.cs # Services de données

├── **Models**/

│ ├── Etat.cs # Modèle État

│ ├── Evenement.cs # Modèle Événement

│ ├── Lot.cs # Modèle Lot

│ ├── Operation.cs # Modèle Opération

│ └── Recette.cs # Modèle Recette

├── **UI**/

│ ├── **Components**/ # Composants réutilisables

│ │ ├── EvenementCarte.cs

│ │ ├── LotCarte.cs

│ │ ├── OperationCarteInformations.cs

│ │ ├── OperationCarteCreation.cs

│ │ └── RecetteCarte.cs

│ ├── Manager.cs # Formulaire principal

│ ├── RecetteCreation.cs # Création/édition recette

│ └── RecetteInformation.cs # Affichage recette

├── **Resources**/ # Ressources (images)

├── FormManager.cs # Gestionnaire de formulaires

├── Global.cs # État global de l'application

└── Program.cs # Point d'entrée

# Base de données

## Schéma de la base de données

### MCD

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

### MLD

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

## Configuration de la connexion

Les chaînes de connexion sont définies dans `App.config` :

<connectionStrings>

  <add name="ServerMySQLLocal"

      connectionString="Server=localhost;Database=m3\_pratique;Uid=root;

\_ Pwd;" />

  <add name="ServerMySQL"

      connectionString="Server=172.16.151.202;Database=gr01\_m3- \_ pratique;Uid=GR01;Pwd=gr01;" />

</connectionStrings>

# Interface utilisateur

## Manager (principal)

Une image contenant texte, capture d’écran, logiciel, nombre

Le contenu généré par l’IA peut être incorrect.Le formulaire principal est divisé en 4 setions :

1. Liste des lots (gauche)

* Recherche par nom
* Filtrage par état
* Affichage des cartes de lots

1. Détails de lot sélectionné (centre-haut)

* Informations du lot
* Historique des événements

1. Création de lot (centre bas)

* Quantité de pièces
* Sélection de la recette

1. Liste de recettes (droite)

* Recherche pas nom
* Bouton pour créer une nouvelle recette
* Affichage des cartes de recettes

## RecetteCreation

Une image contenant texte, capture d’écran, diagramme

Le contenu généré par l’IA peut être incorrect.

Permet de créer ou de modifier une recette :

* Champ de nom de recette
* Liste des opérations modifiable
* Ajout/suppression d’opération
* Réorganisation des opérations (monter/descendre)

## RecetteInformation

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Affiche les détails d’une recette :

* Nom et date de création
* Liste des opérations
* Boutons modifier/supprimer (si non utilisée)

## Composant réutilisable

* EvenementCarte  : affichage d’un événement
* LotCarte  : affichage d’un lot
* OpérationCarteCreation  : affichage d’une opération en edit
* OperationCarteInformation : affichage d’une opération
* RecetteCarte  : affichage d’une recette

# Accès aux données

Au lancement de l’applications tous les modèle sont charger depuis la base de données, il est possible de rafraichir les données depuis le bouton « rafraichir ».

## DatabaseManager

Gère la connexion à la base de données MySQL :

ConnectDB()

Établit la connexion

GetConnexion()

Retourne la connexion active

CloseConnexion()

Ferme la connexion

## databaseService

Fournis les méthodes CRUD pour toutes les entités :

### Méthode de récupération

GetRecettes()

Récupère toutes les recettes.

GetOperation()

Récupère toutes les opérations.

GetLots(List<Recette> recette)

*recette*  : lie la recette correspondante au lot.

Récupère tous les lots.

### Méthodes d’insertion

AjouterRecette(string nom, List<Opération operations)

*nom*  : nom de la recette.

*operations*  : liste des opérations lié à la recette.

Ajoute une recette puis ajoute les opérations dans la base de données.

AjouterOperation(Operation operation, long idRecette)

operation  : opération à ajouter.

idRecette  : id de la recette liée à cette opération.

Ajoute une opération dans la base de données.

AjouterLot(int quantite, lonf idEtat, long idRecette, string recetteNom)

*quantite*  : quantité de pièce.

*idEtat*  : id de l’état du lot.

*idRecette*  : recette utilisée pour le lot.

*recetteNom*  : nom de la recette.

Ajoute un lot dans la base de données.

### Méthodes de mise à jour

ModifierRecette(long idRecette, string nom)

*idRecette*  : id de la recette à modifier.

*nom*  : nouveau nom de la recette.

Modifie le nom d’une recette.

### Méthode de suppression

SupprimerRecette(long idRecette)

idRecette  : id de la recette à supprimer.

Supprime une recette.

SupprimerOperationByRecette(Long idRecette)

idRecette  : id de la recette contenant les opérations à supprimer.

Supprime toutes les opérations liées à une recette.

# Gestion des formulaires

## FormManager

Classe statique gérant l’ouverture et la réutilisation des formulaires :

* Pattern Singleton pour éviter les doublons.
* Méthodes pour ouvrir chaque type de formulaire.
* Gestion de l’état des fenêtres (minimisées, maximisées, normal).
* Passage de données entre formulaires.

## Global

Classe statique servant de cache local et de façace pour DatabaseService :

* Listes statiques pour chaque entité.
* Méthodes de récupération avec tri/filtrage.
* Délégation vers DatabaseService pour les opérations CRUD

# Dépendances

## Packages NuGet

**MySql.Data (9.2.0)** : Connecteur MySQL pour .NET

# Annexe

## Github

[**https://github.com/BilelHofer/M3-pratique**](https://github.com/BilelHofer/M3-pratique)/

├── **Document**/ # Document en lien avec le projet M3

│ ├── **Base de données**/ # MLD/MCD et dump de la db

│ ├── **Consignes**/ # consignes reçu

│ ├── **Dossier**/ # Dossier technique et powerpoint

│ ├── **Maquette**/ # maquettage initial

│ ├── Planification.gan # planification prévisionnel

│ ├── Planification.pdf # pdf # planification prévisionnel en pdf

│ ├── Suivie planification.pdf # suivi de la planification

├── **src**/ # code source

## Documentation généré

TODO mettre la doc générer au format .html ig