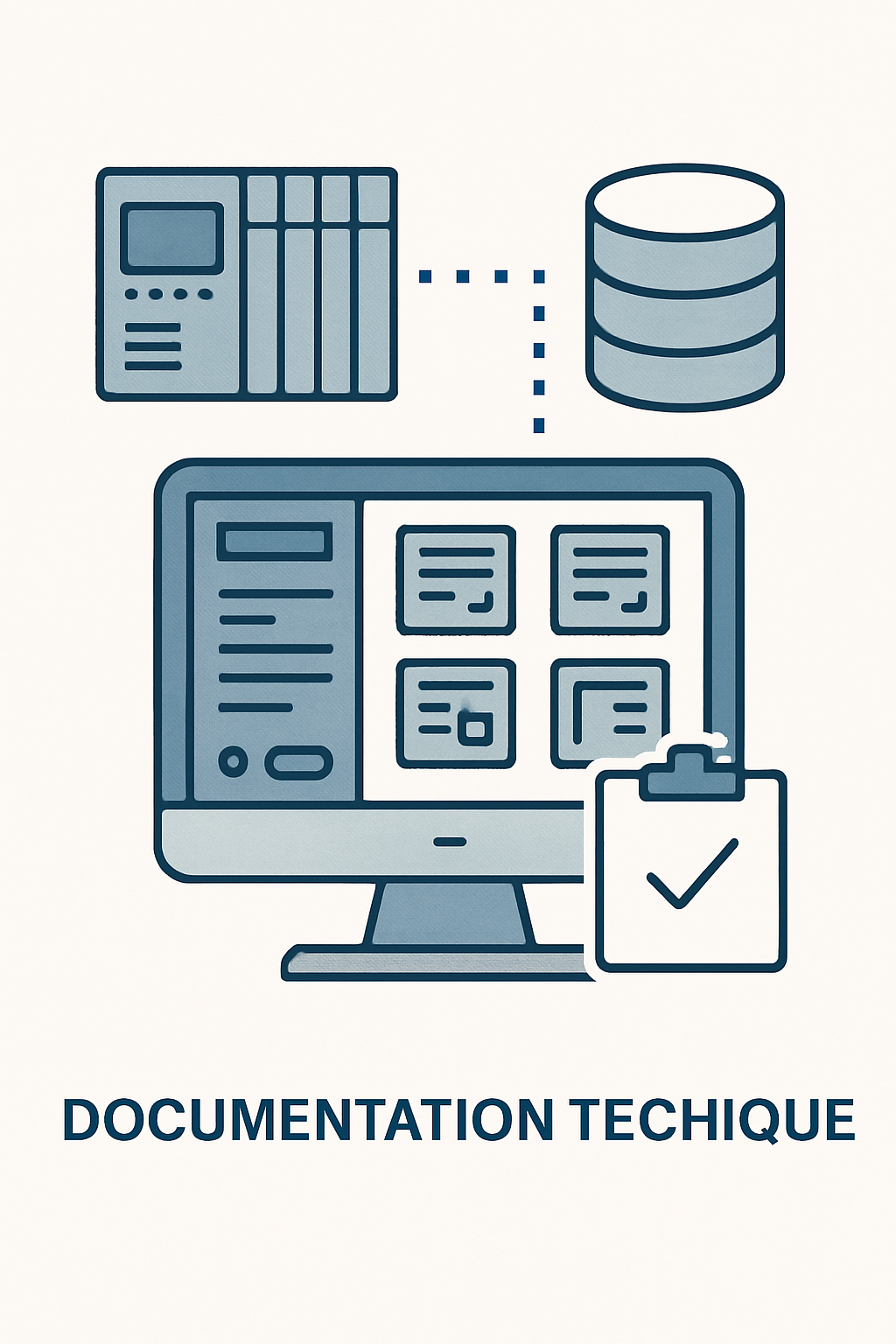
Documentation technique module 3

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

Table des matières

[Introduction 3](#_Toc201669505)

[Fonctionnalités principales 3](#_Toc201669506)

[Technologies utilisées 3](#_Toc201669507)

[Dépendances 3](#_Toc201669508)

[Structure du projet 4](#_Toc201669509)

[1. Base de données 5](#_Toc201669510)

[1.1 Schéma de la base de données 5](#_Toc201669511)

[1.1.1 MCD 5](#_Toc201669512)

[1.1.2 MLD 5](#_Toc201669513)

[1.2 Configuration de la connexion 6](#_Toc201669514)

[2. Interface utilisateur 7](#_Toc201669515)

[2.1 Manager (principal) 7](#_Toc201669516)

[2.2 RecetteCreation 8](#_Toc201669517)

[2.3 RecetteInformation 9](#_Toc201669518)

[2.4 Composant réutilisable 9](#_Toc201669519)

[3. Accès aux données 10](#_Toc201669520)

[3.1 Composants principaux de l'accès aux données 10](#_Toc201669521)

[4. Gestion des formulaires 11](#_Toc201669522)

[4.1 FormManager 11](#_Toc201669523)

[4.2 Global 11](#_Toc201669524)

[Annexe 12](#_Toc201669525)

[Github 12](#_Toc201669526)

[Documentation détaillée générée 12](#_Toc201669527)

# Introduction

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

En annexe, vous retrouverez une documentation détaillée des fonctions.

## Fonctionnalités principales

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

## Technologies utilisées

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

## Dépendances

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

## 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

Pour se connecter à la base de données, notre application utilise les chaînes de connexion qui 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

## Une image contenant texte, capture d’écran, logiciel, nombre Le contenu généré par l’IA peut être incorrect.Manager (principal)

Le formulaire principal est divisé en 4 sections :

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 modifiables
* 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 démarrage de l’application, tous les modèles sont chargés depuis la base de données. Il est également possible de rafraîchir les données à tout moment en utilisant le bouton « Rafraîchir ».

## Composants principaux de l'accès aux données

DatabaseService

Cette classe est responsable de la communication avec la base de données. Elle fournit des méthodes pour interagir avec les données de la base de données.

DatabaseManager

Cette classe gère la connexion à la base de données MySQL.

# 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, normales).
* Passage de données entre formulaires.

## Global

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

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

# 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 reçues

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

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

│ ├── Planification.gan # planification prévisionnelle

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

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

├── **src**/ # code source

## Documentation détaillée générée

<https://bilelhofer.github.io/>