

CAHIER DES CHARGES

Application C# en localhost – Gestion d'une auto-école

Niveau : Bachelor

1. Introduction

Dans le cadre de la formation Bachelor , ce projet a pour objectif la conception et le développement d'une application informatique en **C#**, fonctionnant en **localhost**, destinée à la gestion complète d'une auto-école.

Actuellement, la gestion des élèves, des moniteurs, des véhicules et des leçons est réalisée de manière manuelle. Cette situation entraîne des difficultés d'organisation, des risques d'erreurs et une perte de temps.

L'application proposée vise à **automatiser ces processus**, à améliorer la fiabilité des données et à fournir des outils de suivi et de statistiques.

2. Objectifs du projet

Les objectifs principaux du projet sont les suivants :

- Informatiser la gestion administrative de l'auto-école
- Gérer les clients (élèves), les moniteurs, les véhicules et les modèles de véhicules
- Planifier et suivre les leçons de conduite
- Appliquer les principes de la **Programmation Orientée Objet (POO)** en C#
- Mettre en place des **interfaces métier**
- Produire des **statistiques de gestion** (planning, kilométrage, chiffre d'affaires)

3. Architecture technique

L'application sera développée selon une architecture moderne et structurée :

- **Type d'application** : Application C# en localhost
- **Framework** : ASP.NET Core MVC
- **Architecture logicielle** : Architecture en couches
 - Couche Présentation
 - Couche Métier

- Couche Accès aux Données
- **ORM** : Entity Framework Core
- **Base de données** : MySQL ou PostgreSQL en local

Cette architecture permet une bonne séparation des responsabilités, une maintenance facilitée et une évolution du projet.

4. Modèle Entités–Associations

4.1 Entités obligatoires

Le modèle conceptuel de données repose sur les entités suivantes :

- Client
- Etablissement
- Moniteur
- ModeleVoiture
- Voiture
- Lecon
- Facturation
- Mois (entité technique pour les statistiques)

4.2 Associations principales

- Un **client** peut suivre plusieurs **leçons**
- Un **moniteur** encadre plusieurs **leçons**
- Une **voiture** peut être utilisée pour plusieurs **leçons**
- Un **modèle de voiture** peut correspondre à plusieurs **voitures**
- Un **client** peut avoir plusieurs **facturations**

5. Modèle POO – Classes C#

Chaque entité du modèle Entités–Associations est représentée par une **classe C#**.

Les relations entre les entités sont modélisées à l'aide :

- de clés étrangères
- de propriétés de navigation

Les règles de gestion et les traitements ne sont pas implémentés directement dans les classes, mais délégués à des **services métier**, conformément aux bonnes pratiques de la POO.

6. Interfaces métier

Les principales interfaces métier à implémenter sont :

- **ILeconService**
 - Ajout et gestion des leçons
 - Vérification des contraintes de planning
- **IFacturationService**
 - Calcul des montants
 - Gestion des différents modes de facturation
- **IStatistiqueService**
 - Production des statistiques
 - Génération des indicateurs de gestion

Ces interfaces garantissent la modularité et la testabilité de l'application.

7. Règles de gestion

Les règles de gestion suivantes doivent être respectées :

- Il est interdit de planifier deux leçons qui se chevauchent pour :
 - un même client
 - un même moniteur
 - un même véhicule
- Un moniteur et un véhicule doivent être disponibles au moment de la leçon
- La cohérence entre le modèle de voiture et le véhicule utilisé doit être respectée
- Les données doivent respecter l'intégrité référentielle

8. Requêtes de statistiques

L'application doit permettre de produire les statistiques suivantes :

- Planning hebdomadaire ou journalier par moniteur
- Kilométrage mensuel par véhicule
- Nombre de leçons par client ou par moniteur
- Chiffre d'affaires mensuel de l'auto-école

Ces statistiques sont réalisées à l'aide de requêtes **LINQ** ou **SQL**.

9. Livrables attendus

Les livrables attendus pour ce projet sont :

- Le projet Visual Studio (.NET)
- La base de données locale
- Les scripts SQL de création et d'initialisation
- Les diagrammes UML / MCD
- Un jeu de données de test
- La documentation technique et utilisateur

10. Conclusion

Ce projet permet de mettre en œuvre les compétences acquises en Bachelor , notamment en **C#, Programmation Orientée Objet, bases de données et architecture logicielle**.

Il constitue une application réaliste répondant à un besoin concret de gestion d'une auto-école.

NB : Rendu du projet en groupe sur l'espace teams le 15 Mars 2026