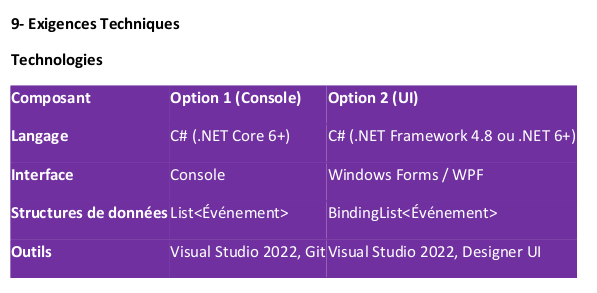
– Projet Final –

Agenda en C# avec WinForms

Shirley Allaire  
Session Hiver 2025  
Étudiante – Collège LaSalle  
Programmeur-analyste

# 1. Introduction

Bonjour, je me nomme Shirley — passionnée et énergique dans tout ce que j’entreprends.  
  
Pour commencer, je vais vous raconter toutes les étapes qui peuvent traverser l’esprit d’un programmeur… simplement pour monter un agenda comme projet scolaire.  
  
J’ai choisi ce thème parce qu’il m’a vraiment interpellée. Un agenda, c’est quelque chose qu’on utilise tous les jours pour organiser toutes les actions de nos journées. Quand j’ai reçu la documentation du projet à faire en C# le 25 février 2025, une indication à la toute fin m’a allumée —

Et là, boum, je pars en grand dans mon apprentissage de WinForms.  
Je suis allée voir mon historique YouTube… et sans rire, ça ferait peur à n’importe qui! J’ai 65 vidéos enregistrées et déjà regardées, parfois même plusieurs fois. J’ai commencé à monter des interfaces graphiques, à télécharger des NuGets — parce que je voulais faire ça en grand. J’étais en mode “découverte” : passer d’une ligne de code à créer une interface graphique… le rêve!  
  
Mais en parallèle, dans notre cours, on a commencé à apprendre les structures.  
  
Et là, vraiment : WOW.  
J’ai mis mon WinForm de côté parce que ce qu’on voyait en classe… c’était exactement ce qu’il me fallait pour développer mon agenda comme il faut. Parce qu’un agenda sans structure… ça ne fonctionne juste pas.  
  
Donc je repars dans une nouvelle direction : monter une bonne structure comme M. Mourad, notre professeur, nous montrait. Et j’ai vraiment trippé. D’un cours à l’autre, je construisais mon interface d’agenda, avec une grande vision, une grosse structure... peut-être trop grosse pour le temps qu’il me restait.  
  
Avec les autres projets finaux, les examens, la fatigue de la fin de session, l’écoeurantite de l’hiver… j’ai dû faire des choix.  
  
Quand mes derniers examens ont été remis, le dos allégé, je regarde ma structure pratiquement complétée. Je la fais fonctionner… et là, surprise : aucun rendez-vous n’apparaît. Ils n’étaient pas enregistrés nulle part.  
  
Alors je recommence à chercher : comment connecter Visual Studio 2022 à une base de données?  
C’est là que je tombe sur une vidéo de Tim Corey : https://www.youtube.com/watch?v=Et2khGnrIqc  
  
Et là, je me dis : ce gars-là, rien ne l’arrête. Il explique comment relier une base SQL à un petit projet WinForm de Visual Studio 2022. Ses explications sont claires, simples, efficaces.  
  
J’ai donc appliqué ses méthodes, en commençant par créer une classe Helper.cs qui affiche le nom de ma connectionString. Cette connectionString, je la crée dans le fichier App.config, qui est déjà présent dans un projet Console App standard. En quelques lignes de code, j’ai une méthode qui retourne ma connexion.  
  
Facile? Heuu… non   
Mon Visual Studio est installé sur une machine virtuelle (VM) parce que je suis sur Linux, et mon SQL Server tourne en local sur Windows. Impossible d’utiliser SSMS, donc j’utilise DBeaver.  
  
Premier défi : connecter Visual Studio (dans la VM) à ma base SQL sur mon PC hôte. Après plusieurs essais-erreurs et beaucoup de persévérance, j’ai enfin réussi à me connecter. Ma base de données AgendaDB est créée et ma connexion fonctionne!  
  
Trop heureuse, je reprends la vidéo de Tim Corey, et là je me dis :  
“C’est enfin facile! Les examens sont finis, j’ai 6 jours devant moi, ma structure est prête… go, je me lance dans WinForms!”  
  
Le reste de cette documentation va donc détailler toutes les étapes que j’ai suivies pour créer une interface graphique simple, un petit agenda qui — par manque de temps — permet :  
- d’ajouter de nouveaux contacts dans le carnet d’adresses  
- d’ajouter de nouveaux rendez-vous dans le carnet de RDV  
- de rechercher un contact (par prénom, nom de famille, etc.)  
- et d’afficher les rendez-vous à partir du calendrier intégré  
  
Je vous laisse maintenant découvrir ma documentation.

# 2. Structure du projet

Le développement de ce projet s’est fait de manière progressive et logique. Voici l’ordre dans lequel les composants ont été créés, avec une brève explication pour chaque élément technique :  
  
1. ---- App.config ----  
 Ce fichier contient la chaîne de connexion (`connectionString`) pour se connecter à la base de données SQL Server. Il est essentiel au bon fonctionnement de la couche de données.  
  
2. ---- Helper.cs ----  
 Une classe utilitaire qui permet de récupérer facilement la chaîne de connexion à la base de données depuis le fichier App.config.  
  
3. ---- DataAccess.cs ----  
 Cette classe contient toutes les méthodes d’accès aux données (insertion, recherche, etc.) et utilise Dapper pour exécuter les procédures stockées SQL.  
  
4. ---- StructAgenda.cs ----  
 Contient les structures de données utilisées pour représenter les entités principales de l’application : les contacts et les rendez-vous.  
  
5. ---- Form1.cs (Interface graphique) ----  
 C’est la forme principale du projet WinForms. Elle contient les composants visuels (textboxes, labels, boutons, calendrier, etc.) ainsi que les événements liés à l’interaction de l’utilisateur.  
  
6. ----Base de données SQL – AgendaDB----  
 Créée manuellement via DBeaver, elle contient les tables ‘Contact’, ‘RendezVous’, ainsi que plusieurs procédures stockées pour insérer et rechercher des données.

---- Diagramme de la base de données ----

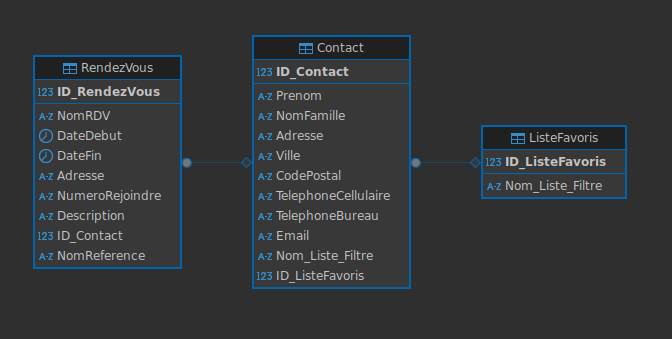
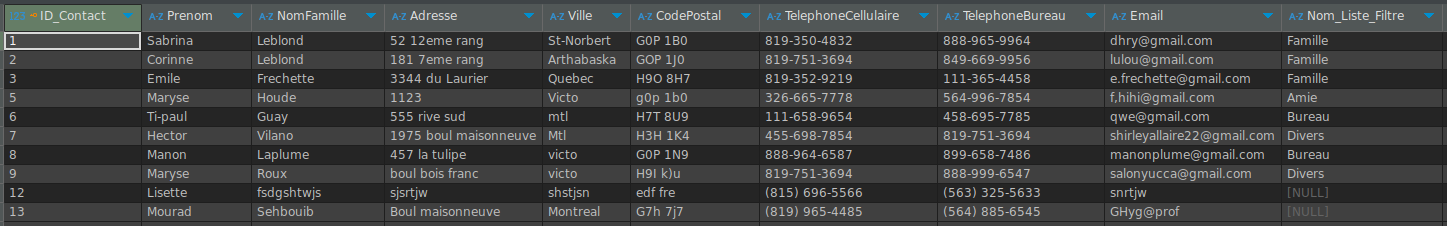


Diagramme relationnel représentant les entités et les liens entre ‘Contact’ et ‘RendezVous’.

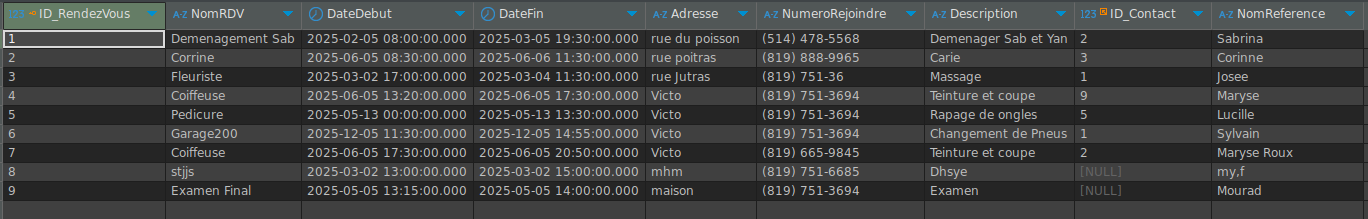
# 3. Fonctionnalités de l’application

L’application Agenda développée en C# avec WinForms intègre plusieurs fonctionnalités essentielles pour la gestion de contacts et de rendez-vous.  
  
---- Ajout de contacts ----  
L’utilisateur peut ajouter un nouveau contact avec les champs suivants : prénom, nom, ville, téléphone, courriel, etc. Les données sont insérées via une procédure stockée dans la base de données.



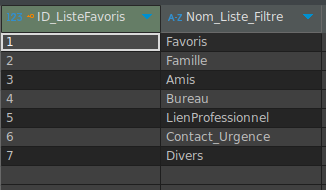
Interface pour l’ajout de contacts.

---- Ajout de rendez-vous----  
Une autre fonctionnalité permet d’ajouter des rendez-vous dans le carnet RDV, avec titre, description, heure, date et contact associé.



Formulaire d’ajout de rendez-vous.

---- Recherche de contact ----  
Il est possible de rechercher un contact selon différents critères (prénom, nom de famille, ville, etc.) et d’afficher les résultats dans une ListBox.



Exemple de recherche dans le carnet d’adresses.

### ---- Difficultés techniques rencontrées ----

Une des grandes difficultés du projet a été la connexion entre Visual Studio (installé sur une machine virtuelle) et la base de données SQL Server, qui tournait localement sur un environnement Windows.  
Le projet étant développé sous Linux, l’environnement ne permettait pas l’utilisation classique de SSMS (SQL Server Management Studio), d’où l’utilisation de DBeaver.  
  
La connexion nécessitait la bonne configuration des ports, du réseau et de la chaîne de connexion (`connectionString`) dans App.config. Cela a nécessité beaucoup de recherches, de vidéos tutoriels et d’essais-erreurs avant d’obtenir une connexion pleinement fonctionnelle.

# 4. Ressources utilisées

Tout au long du développement de ce projet, plusieurs ressources m’ont été utiles pour comprendre, apprendre et résoudre des problèmes techniques. Voici les principales :  
  
----- YouTube – IAmTimCorey -----

Vidéo extrêmement utile pour comprendre comment connecter une base de données SQL Server à un projet C# avec Dapper et WinForms. https://www.youtube.com/watch?v=Et2khGnrIqc  
  
----- DBeaver -----  
 Logiciel utilisé pour interagir avec la base de données SQL Server sous Linux, en remplacement de SSMS.  
  
----- NuGet Packages -----  
 - Dapper : pour la gestion des requêtes SQL via C#   
 - System.Data.SqlClient : pour l’interfaçage avec SQL Server  
  
----- Visual Studio 2022 (sous VM) -----  
 Environnement de développement principal.  
  
----- Sites web et forums -----  
 Stack Overflow, Microsoft Learn, blogs techniques, et documentations officielles ont été des ressources précieuses à plusieurs étapes du développement.

# 5. Conclusion

Ce projet d’agenda m’a permis de sortir de ma zone de confort, de découvrir de nouveaux outils et d’expérimenter concrètement les concepts appris en classe.  
  
J’ai compris à quel point la structure d’un programme est essentielle à sa stabilité et à sa clarté. Le développement en WinForms, que je pensais “facile”, s’est avéré plus complexe, mais aussi plus enrichissant que prévu.  
  
Malgré les obstacles, les erreurs et les moments de découragement, j’ai persévéré. J’ai appris à chercher, à comprendre, à adapter… et surtout à ne jamais abandonner.  
  
Même si toutes les fonctionnalités prévues n’ont pas été terminées, je suis fière du chemin parcouru et de tout ce que j’ai pu intégrer dans cette application. Ce projet représente non seulement un travail scolaire, mais aussi une \*\*expérience formatrice et personnelle\*\*.  
  
Je ressors de cette session plus outillée, plus confiante, et prête à relever les défis de la prochaine étape.  
  
Merci.

— Shirley Allaire