

SEVRAIN

Hugo

SIO SLAM 2^{ème} année

Note de synthèse

03/01/2022 – 04/02/2022

Lycée Paul Claudel Laon



Mycélium, Laon



Sommaire

1. Présentation de l'organisation.....	3
2. L'informatique dans l'organisation.....	4
3. Les deux dernières semaines.....	5
3. a) Contexte.....	5
3. b) Financier :.....	6
3. c) Préparateur :.....	6
4. Conclusion.....	7
5. Annexes.....	8
Annexe n°1 : Schéma de la base de données.....	8
Annexe n°2 : Connexion de l'employé.....	9
Annexe n°3 : Partie « Financier ».....	10
Annexe n°4 : Partie « Préparateur ».....	11
Annexe n°5 : Génération du PDF.....	12

1. Présentation de l'organisation

[Mycélium](#) est une SARL unipersonnelle créée le 7 mars 2013 dont le siège social se situe au 60 Rue de Manoise, 02000 Laon. Elle est spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques. Son effectif est compris entre 1 et 2 salariés. Sur l'année 2013 elle réalise un chiffre d'affaires de 198 600 €.

2. L'informatique dans l'organisation

La gestion (des stocks, des interventions pour les clients...) est gérée à l'aide de l'intranet de Mycélium.

Pour les tests de performances, un benchmarking est fait à l'aide du logiciel 3DMark par FutureMark.

3. Les deux dernières semaines

3. a) Contexte

Après deux semaines d'observation chez Mycélium, je n'ai pas poursuivi le stage.

Pendant les deux dernières semaines, j'ai réalisé, lors d'un TP, un programme de gestion des stocks en C# et plus précisément à l'aide de WinForms.

Après avoir découvert le contexte de ce TP, j'ai créé la base de données ainsi que les classes des entités et managers (voir annexes) en plus d'avoir géré la connexion à la base de données depuis l'application à l'aide du package NuGet `MySql.Data`

En plus d'avoir appris de nouvelles choses que je vais détailler par la suite, je me suis familiarisé davantage avec le développement en C# ainsi que l'utilisation de WinForms et les possibilités que ce dernier offre.

La première fenêtre à développer fut la fenêtre de connexion. J'ai donc rajouté à la base de données (voir annexe n°1) une table `employe` contenant le nom, prénom, login (unique), mot de passe (haché) et type de l'employé. Le type est un entier (INT) correspondant soit à un financier soit à un préparateur. Lorsque l'utilisateur tente de se connecter à l'aide de son login et mot de passe, un test est fait pour savoir s'ils correspondent à ceux d'un employé présent dans la base de données, un message d'erreur apparaît dans le cas échéant (voir annexe n°2).

Le mot de passe est haché en SHA256 pour ne pas être stocké en clair dans la base de données.

SHA-256 signifie Secure Hash Algorithm 256-bits et est utilisé pour le hachage qui est irréversible.

Par exemple, le mot "toto" haché devient `31f7a65e315586ac198bd798b6629ce4903d0899476d5741a9f32e2e521b6a66`.

Lorsque l'utilisateur tente de se connecter, un premier test est fait pour savoir si l'employé existe (`isEmployeExisting`). Si la combinaison du login et du mot de passe n'existe pas dans la base de données ou si la connexion a réussi mais le type de l'employé est différent de 1 ou 2, un message d'erreur apparaît (voir annexe n°2).

Si la connexion réussit et que le type est égal à 0 ou 1, un message apparaît (voir annexe n°2, un message similaire apparaît pour le financier) et la fenêtre s'ouvre :

3. b) Financier :

Pour le financier, il doit d'abord afficher les commandes, ce que j'ai fait avec un `DataGridView` de WinForms dont les données proviennent d'une commande SQL `SELECT` qui affiche les commandes en attente de paiement ainsi que celles étant validées mais pas encore en cours de préparation ni expédiées. J'ai également ajouté une `checkbox` qui permet d'afficher ou non les commandes validées.

Le financier peut également valider les commandes en attente de paiement (et également remettre en attente de paiement en cas d'erreur), ce que j'ai géré avec un bouton dont le code qu'il effectue dépend de l'état de la commande sélectionnée à l'aide d'un « switch case » (voir annexe n°3).

3. c) Préparateur :

Pour le préparateur, il voit d'abord les commandes (état en attente de paiement exclu) triées par état puis par date de commande.

Le préparateur a accès à un bouton lui permettant de changer l'état d'une commande, mais contrairement au financier, cette action est irréversible. La commande a une date d'expédition mise à jour lorsque l'état passe de 3 à 4.

Le bouton devient masqué et le label devient affiché si l'état de la commande sélectionnée est égal à 4 (voir annexe n°4).

Enfin, le préparateur peut générer le PDF d'une fiche de commande (voir annexe n°5) grâce à un bouton n'étant accessible que pour les commandes expédiées.

J'utilise le package NuGet `Aspose.Pdf` pour gérer les fichiers PDF (voir annexe n°5).

Je commence par créer le document et j'y ajoute une page dans laquelle j'insère des fragments (`TextFragment`).

Les premiers fragments sont les informations de l'entreprise ainsi que celles de la commande. Je récupère les informations à insérer de la base de données afin de les insérer dans un tableau : la première et dernière lignes sont insérées directement, mais celles contenant les informations sur les articles de la commande sont insérées à l'aide de la structure itérative `for` (voir annexe n°5).

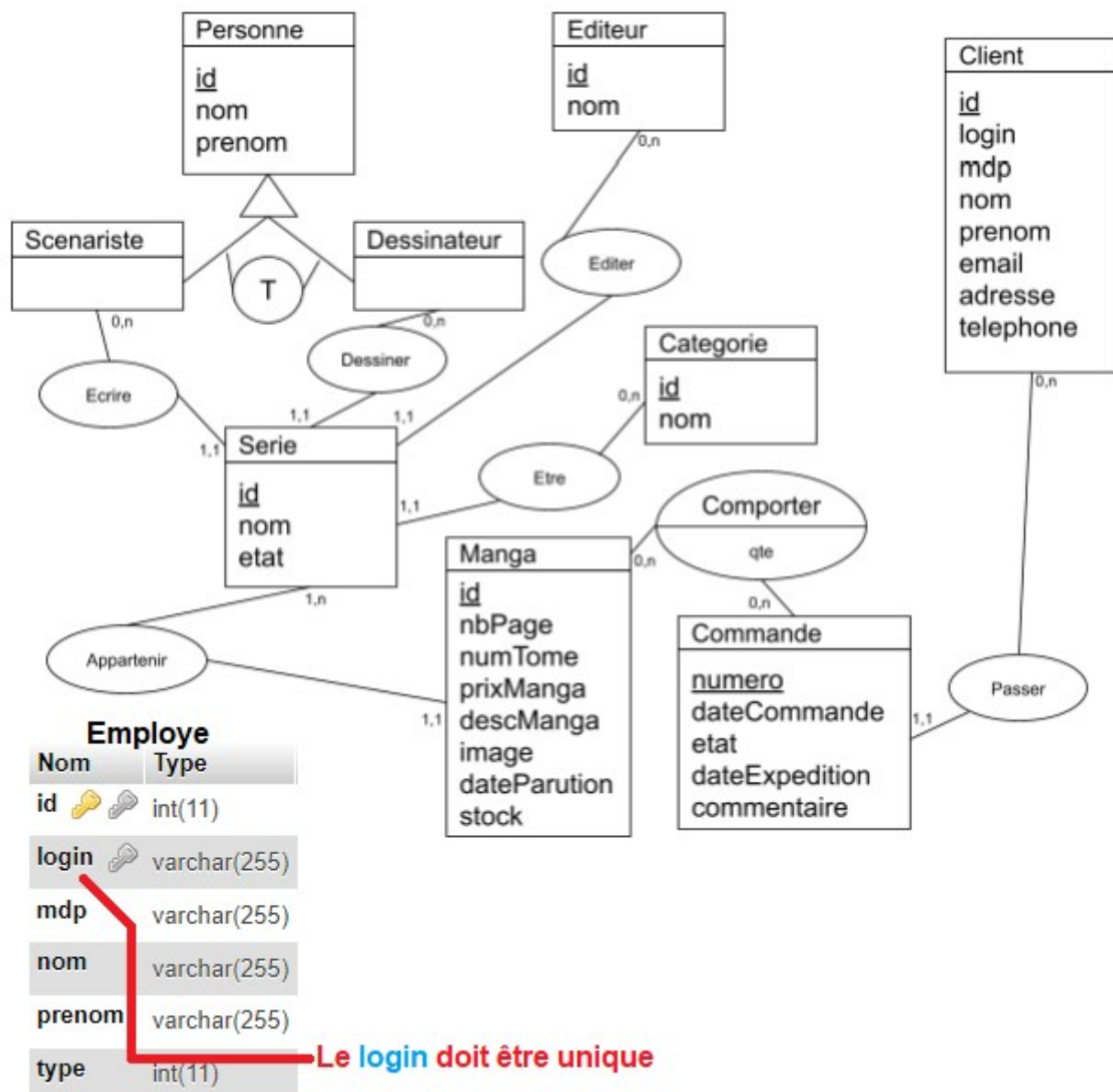
4. Conclusion

Pour conclure, malgré quelques difficultés et une rupture du stage, cette expérience m'a été enrichissante. J'ai une nouvelle fois observé comment fonctionne le monde professionnel, et j'ai également découvert le fonctionnement et l'utilité de l'Intranet d'une organisation. De plus, j'ai appris ce qu'est un test de performances et le logiciel 3DMark.

Durant les deux dernières semaines, j'ai eu l'occasion d'expérimenter avec le langage C# davantage, ce qui m'a permis de revoir les notions que je ne maîtrisais pas entièrement en plus de renforcer les compétences que j'avais déjà acquises. J'ai également utilisé WinForms et j'ai appris à faire de nouvelles choses avec les widgets de ce dernier. J'ai découvert le package NuGet `Aspose.PDF` qui m'a aidé à générer un fichier au format PDF dans lequel figure le contenu d'une base de données à l'aide d'un code C# déclenché par un bouton WinForms.

Ces semaines m'ont conforté dans ma décision de poursuivre une carrière professionnelle dans l'informatique et plus précisément le développement.

5. Annexes

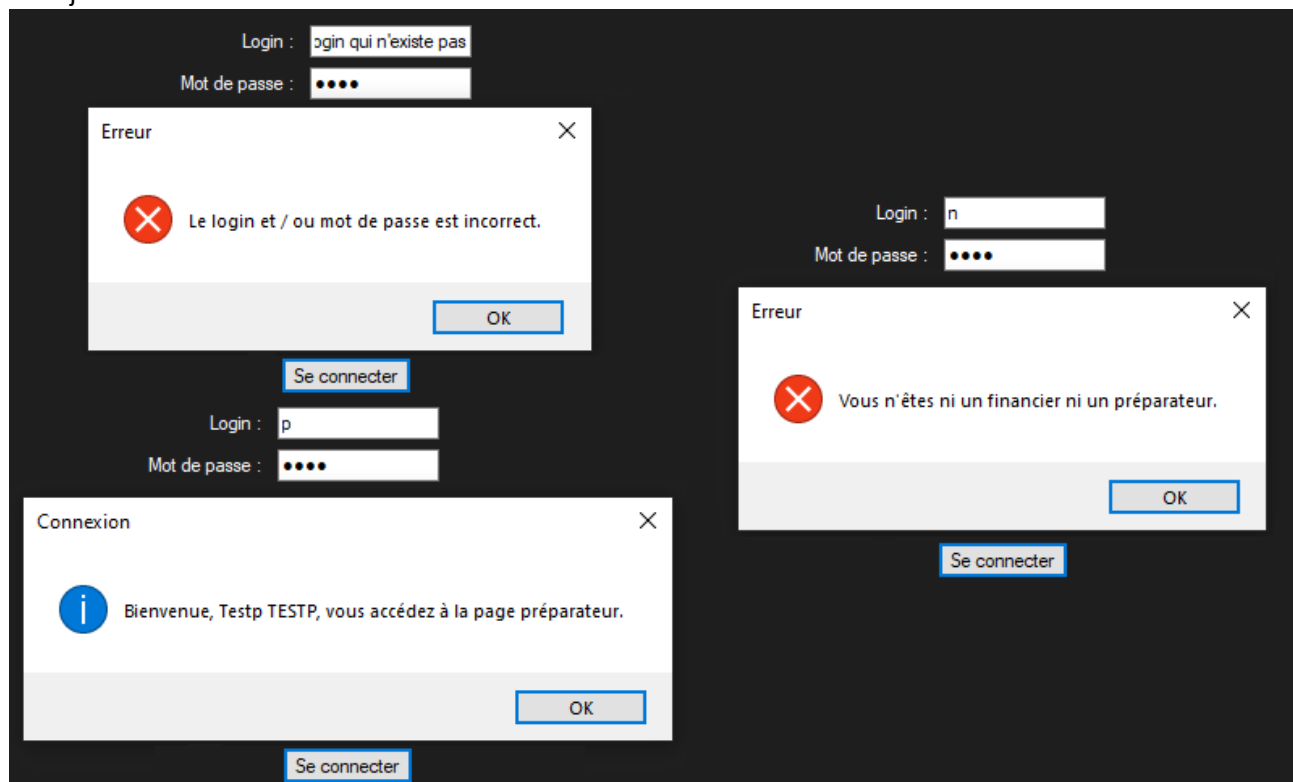


Annexe n°1 : Schéma de la base de données


```
public bool IsEmployeeExisting(string login, string mdp)
{
    MySqlConnection maConnexion = BDDUtilities.OuvrirConnexion();
    MySqlCommand getEmployeeCommand = maConnexion.CreateCommand();
    getEmployeeCommand.CommandText = "SELECT COUNT(*) FROM employe WHERE
login=@login AND mdp=@mdp";
    getEmployeeCommand.Parameters.AddWithValue("@login", login);
    getEmployeeCommand.Parameters.AddWithValue("@mdp", mdp);
    MySqlDataReader reader = getEmployeeCommand.ExecuteReader();

    reader.Read();
    bool resultat = reader.GetInt32(0) == 1;

    BDDUtilities.FermerConnexion(maConnexion);
    return resultat;
}
```



Annexe n°2 : Connexion de l'employé

```
switch (commandState)
{
    case 1:
        commandeManager.ValidateCommand(numeroCommande);
        break;
    case 2:
        commandeManager.RemoveCommandValidation(numeroCommande);
        break;
    default:
        break;
}
```

Commandes :

	Numero	DateCommande	Etat
▶	2	24/01/2022	1

Recherche par numéro de commande :

Les commandes dont etat = 2 sont masquées

☐ Également afficher les commandes validées

Afficher les commandes

Commandes :

	Numero	DateCommande	Etat
▶	2	24/01/2022	1
	3	24/01/2022	2

Recherche par numéro de commande :

Les commandes dont etat est strictement supérieur à deux sont toujours masquées pour le financier

☒ Également afficher les commandes validées

Afficher les commandes

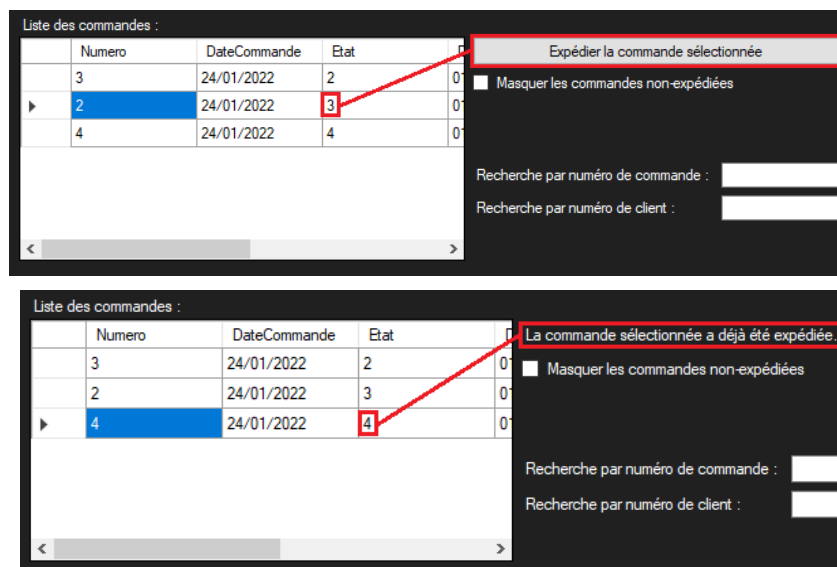
Annexe n°3 : Partie « Financier »

```

switch (commandState)
{
    case 2:
        changeCommandStateButton.Text = "Passer la commande sélectionnée en cours de
préparation";
        changeCommandStateButton.Visible = true;
        expediatedCommandLabel.Visible = false;
        generatePDFButton.Visible = false;
        break;
    case 3:
        changeCommandStateButton.Text = "Expédier la commande sélectionnée";
        changeCommandStateButton.Visible = true;
        expediatedCommandLabel.Visible = false;
        generatePDFButton.Visible = false;
        break;
    case 4:

        changeCommandStateButton.Visible = false;
        expediatedCommandLabel.Visible = true;
        generatePDFButton.Visible = true;
        break;
    default:
        changeCommandStateButton.Visible = false;
        expediatedCommandLabel.Visible = false;
        generatePDFButton.Visible = false;
        break;
}

```



Annexe n°4 : Partie « Préparateur »

// Initialisation de l'objet Document

```
Aspose.Pdf.Document document = new Aspose.Pdf.Document();
```

// Ajout d'une page

```
Page page1 = document.Pages.Add();
```

// Ajout d'un titre à la page

```
TeXFragment title = new TeXFragment("KuroNeko");
```

```
page1.Paragraphs.Add(title);
```

```
for (int i = 0; i < articleAmount; i++)
```

```
{
```

```
    if (numeroCommande == comporter[i].NumeroCommande)
```

```
    {
```

```
        int idManga = comporter[i].IdManga - 1;
```

```
        string imageFilename = manga[idManga].Image;
```

```
        string descManga = manga[idManga].DescManga;
```

```
        float prixManga = manga[idManga].PrixManga;
```

```
        int qte = comporter[i].Qte;
```

```
        float sousTotal = prixManga * qte;
```

```
        quantiteTotal += qte;
```

```
        prixTotal += sousTotal;
```

```
        Aspose.Pdf.Row mangaInfoRow = table.Rows.Add();
```

```
        mangaInfoRow.Cells.Add(descManga);
```

```
        mangaInfoRow.Cells.Add(prixManga.ToString() + "€");
```

```
        mangaInfoRow.Cells.Add(qte.ToString());
```

```
        mangaInfoRow.Cells.Add(sousTotal.ToString() + "€");
```

```
    }
```

```
}
```

KuroNeko
1 Place Foch, 02000 LAON
Tél: 03.00.01.02.03
Email: contact@kuroneko.com

Commande #1
SEVRAIN Hugo
adresse test
Commentaire : Commande 1
Date de commande : 24/01/2022
Date d'expédition : 04/02/2022

Article	Prix unitaire	Quantité	Sous-total
manga1	69.99€	2	139.98€
manga2	5€	1	5€
Total		3	144.98€

Annexe n°5 : Génération du PDF