Pour le cours d’analyse (plus exemple ci-après)

L’axe fonctionnel a permis de définir les cas d’utilisation ou les fonctionnalités importantes pour l’utilisateur.

L’axe statique va permettre de définir les diagrammes de classes correspondant aux cas d’utilisation :

La démarche est la suivante :

* Diagramme des cas d’utilisation : définir les itérations (regroupement des cas d’utilisation qui ont des points communs)
* Pour chaque itération, on peut définir un diagramme de classes partiel ; la marche à suivre est
  + Rechercher pour chaque cas d’utilisation de l’itération, les classes concernées
  + Définir pour chaque classe les attributs nécessaires (comme je l’ai montré en classe une bonne façon de faire est de s’appuyer sur les écrans correspondant au cas d’utilisation choisi)

Pour le travail d’analyse :

Je vous propose de m’envoyer par groupe le travail suivant :

* L’énoncé (sujet choisi…)
* Le diagramme des cas d’utilisation principal
* Le diagramme des cas d’utilisation montrant les itérations
* La ou les itérations par lesquelles vous allez commencer votre projet
* Le diagramme de classe partiel correspondant à une itération (attributs)
* Pour le cours de C#, chaque étudiant devra programmer une itération

Attention : bien respecter les concepts OBJETS

**Exemple : la demande de formation ( commencé au cours d’analyse)**

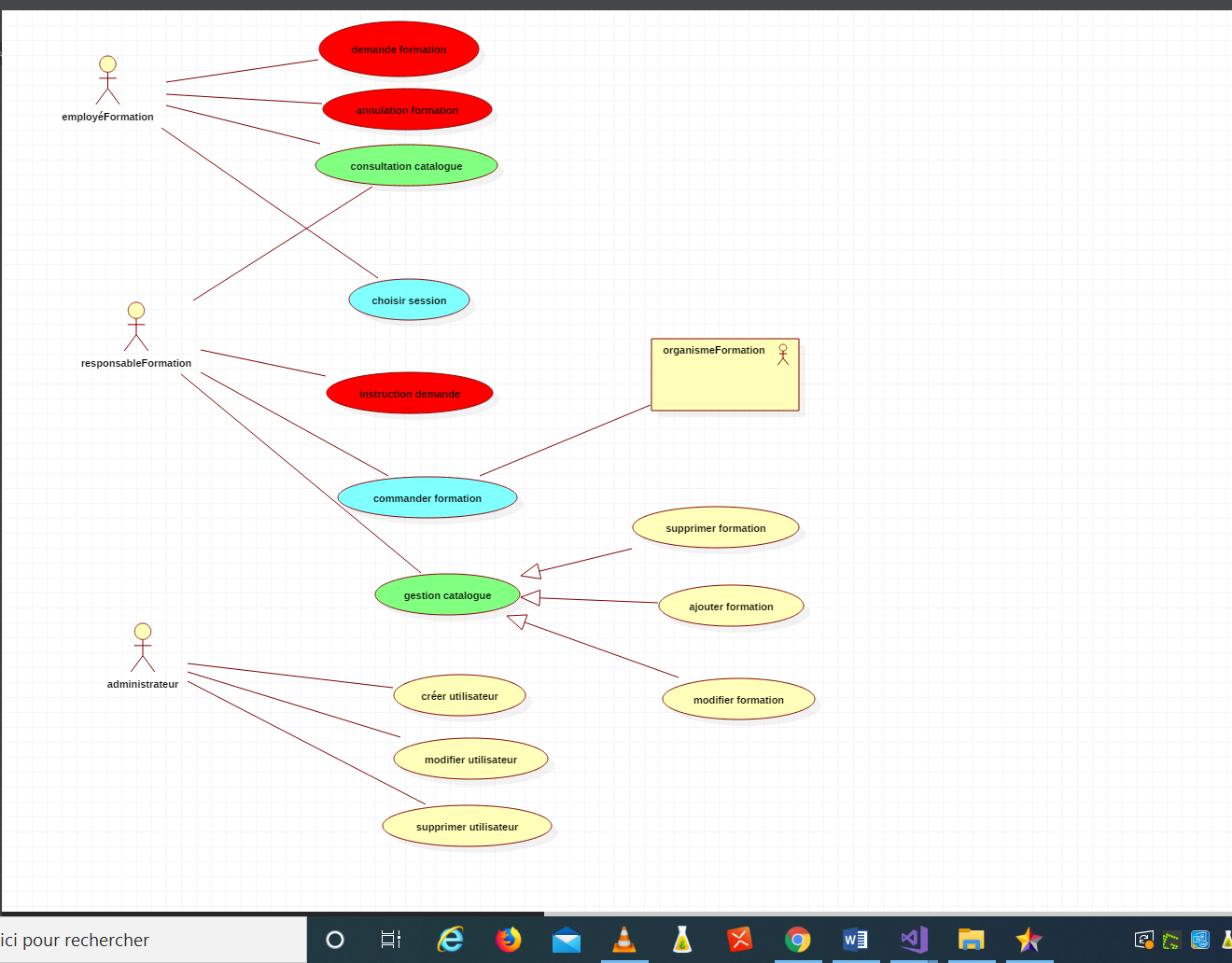


Diagramme des cas d’utilisation « Demande de formations » montrant les différentes itérations

1. En rouge « itération Demande de formations »
2. En vert « gestion du catalogue »
3. En bleu « inscription à une session de formation »
4. En jaune « gestion des utilisateurs ».

***Itération 1 : « traitement de la demande »***

Elle concerne 3 cas d’utilisation :

1. La demande de formation
2. Instruction de la demande
3. Annulation de la demande

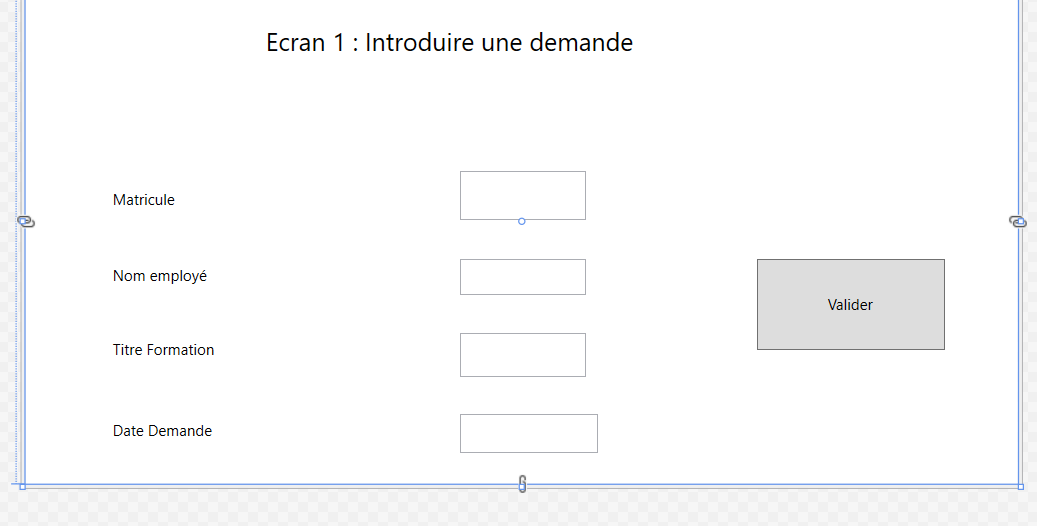
**Le cas « La demande de formation » :**

Quelles sont les classes nécessaires pour répondre à cette demande ?

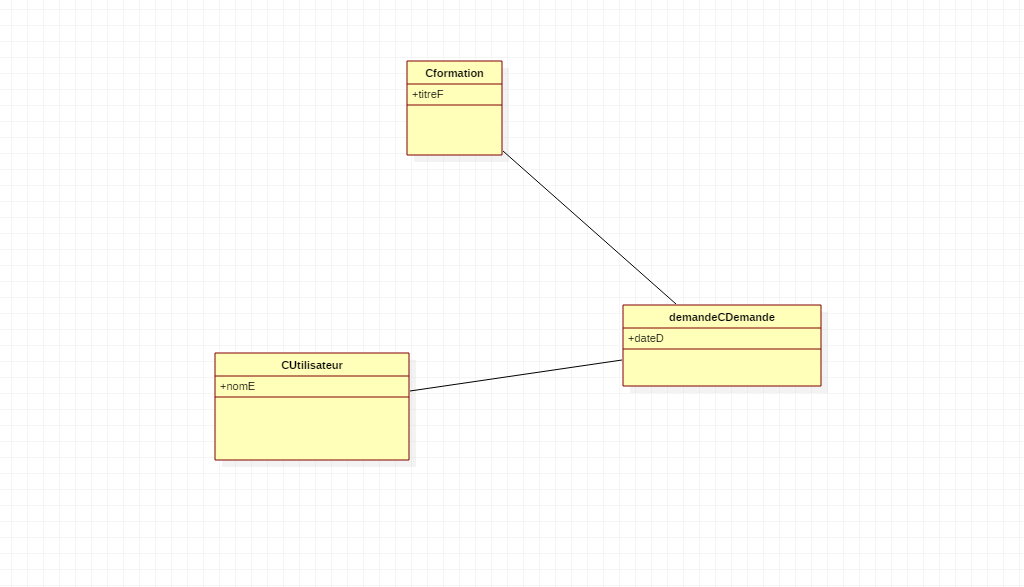
Les classes CUtilisateur, CFormation et Cdemande

La démarche proposée est d’imaginer les écrans correspondant à chaque attribut et de définir ainsi les attributs indispensables pour chaque classe

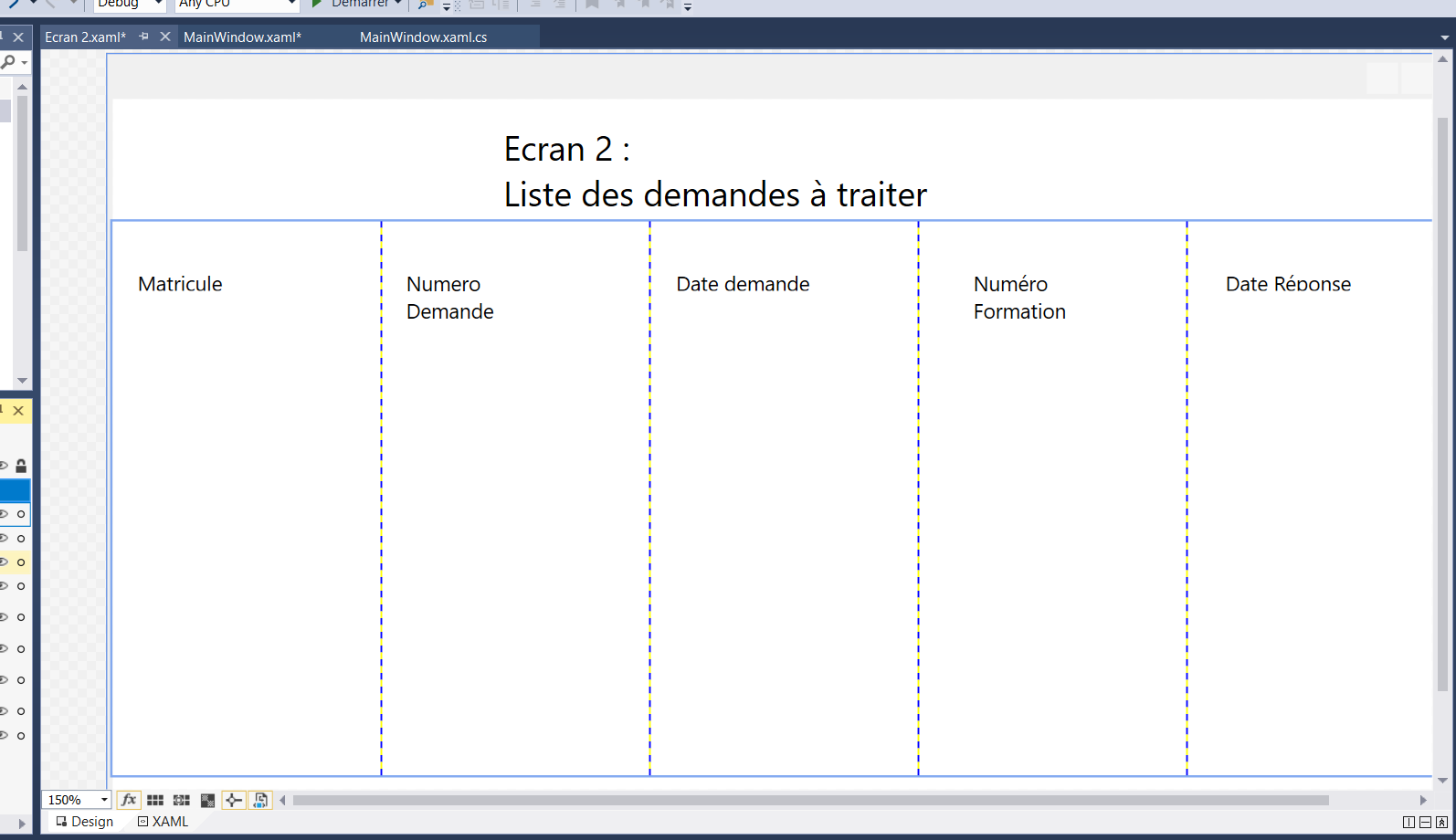
Ecran 1 : L’employé demande une formation



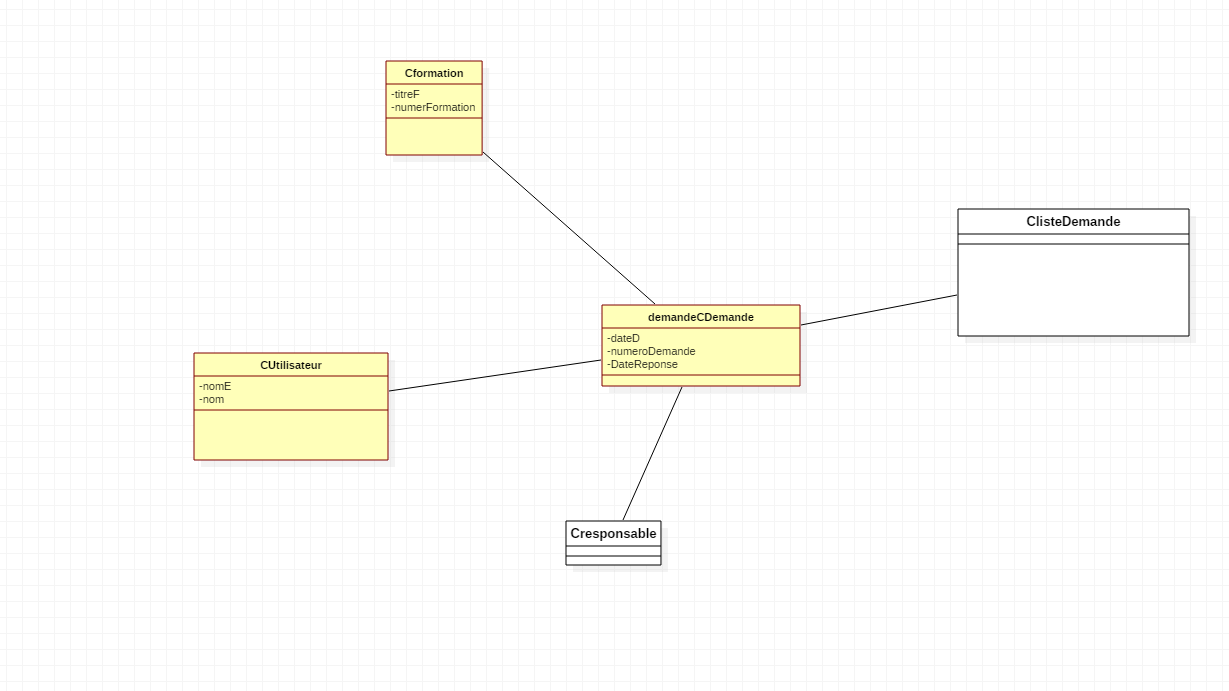
On initie le diagramme de classes (et leurs attributs)



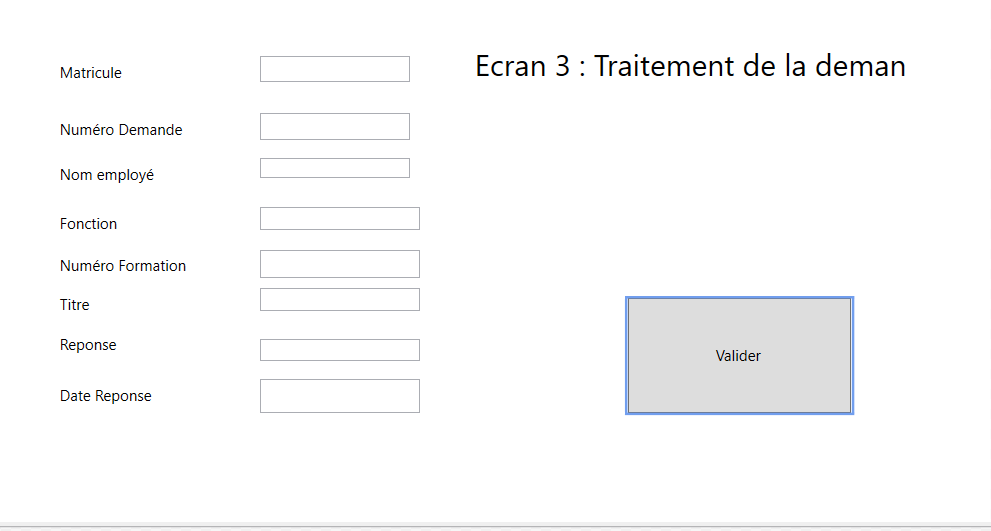
Ecran2 : le responsable choisit une demande dans la liste proposée

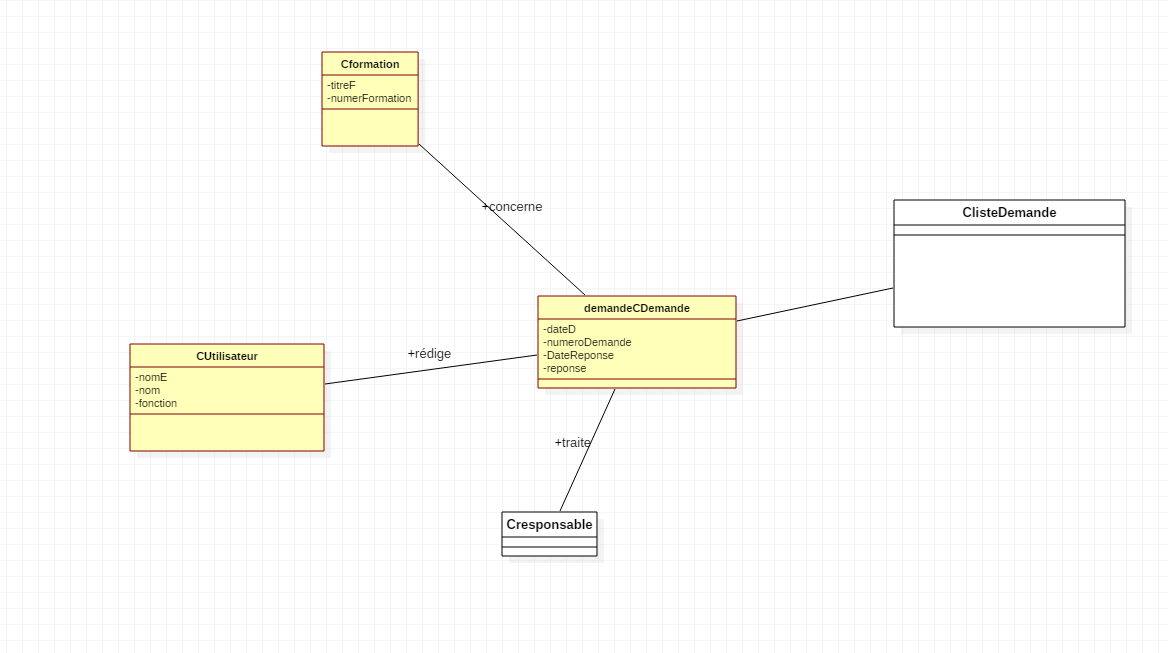


On complète le diagramme de classes partiel



Ecran 3 le responsable traite une demande





Le diagramme de classes partiel correspond à l’itération « Traiter la demande » (l’annulation n’est pas traitée ici car elle n’ajoute pas d’éléments supplémentaires)

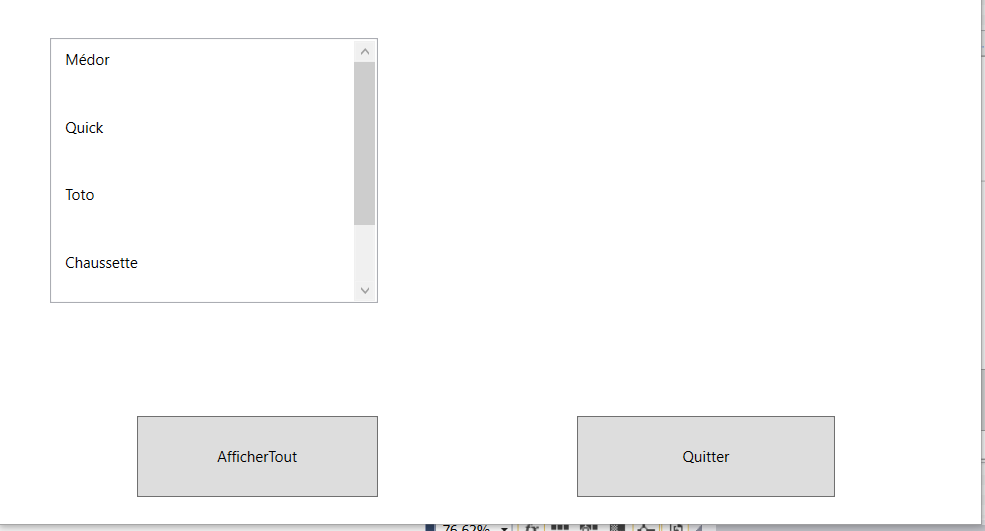
Pour le cours de SGBD :

Terminer les exercices sur les requêtes imbriquées (Bases de données Facsys et base de données Vins). J’enverrai une correction à la fin du mois de Mars.

Pour le cours d’applications informatiques :

Ahmet a posté sur votre groupe l’application « Interface3 » développée au cours d’analyse en mode console.

L’objectif est de reprendre les classes de cette application et de l’intégrer dans une application WPF.



Marche à suivre :

* L’utilisateur choisit un animal dans la liste ; un deuxième formulaire affiche le nom, la race et l’âge de l’animal. Le dompteur envoie un message à l’animal pour lui dire de courir ou de faire du bruit
* Afin de respecter l’architecture en couches d’une application (Interface graphique, classes métiers, source de données), les classes ne doivent pas (ou très peu être modifiées).
* A notre retour à l’école, nous intégrerons une base de données dans cette application WPF.  
  Afin d’avancer plus rapidement, je vous demande de réaliser le schéma de la base de données correspondant à cette application  et de l’implémenter avec SQL Server (créer les tables, les clés primaires et les clés étrangères)  
  Les questions à se poser sont les suivantes :
  1. Quelles sont les différences principales entre un diagramme de classe et une base de données ?
  2. Quelles sont les données à conserver pour que cette application puisse fonctionner ?
  3. SQl Server permet de travailler avec un assistant ou vous pouvez également écrire vos propres scripts de création de tables (cfr cours de SGBD avec Oracle). Personnellement, la dernière façon me semble plus fiable.

Bon travail et bon courage