

ROCHETTE DIMITRI

SCHNEIDER ELECTRIC

AMTFAQ

Schneider
Electric

DU 23/05/2024 AU 28/06/2024
BTS SIO 1ÈRE ANNÉE

SOMMAIRE

1. Mise en contexte	3
1.1 Présentation de l'entreprise	3
1.1.1 Activité à Mâcon	3
1.1.2 Produits fabriqués à Mâcon	3
1.1.3 Nombre de salariés à Mâcon	3
1.2 Présentation du projet	3
1.2.1 Objectif	3
1.2.2 Consignes et contraintes	4
1.3 Étude du projet existant	4
1.3.1 Classe principale	4
1.3.2 Sous-classes	5
1.3.3 Sous-sous-classes	5
2. Rédaction d'un cahier des charges	6
2.1 Fonctionnement	6
2.2 Interface	6
2.3 Ecrans	6
2.4 Logs	7
2.5 Étapes	7
2.6 Analyse des données	8
2.6.1 Base de données	8
2.6.2 Classes	9
3. Développement	10
3.1 Création de l'interface	10
3.1.1 Bouton d'accueil	10
3.1.2 Flèches de navigation	10
3.1.3 Bouton d'ajout d'une fiche	10

3.1.4 Bouton pour rafraîchir la page	10
3.1.5 Identifiant de l'utilisateur	10
3.1.6 Entrée textuelle pour la recherche	10
3.2 Prise en main des éléments déjà créés	11
3.3 Création des fenêtres	11
3.3.1 Méthodes	11
3.3.2 Création de compte	12
3.3.3 Accueil, Catégorie, Type, Sous type	13
3.3.4 Consultation de fiche	14
3.3.5 Création et édition de fiches	15
3.3.6 Recherche	16
3.4 Système de logs	17
3.5 À faire / finir	18
4. Bilans	19
4.1 Semaine 1	19
4.2 Semaine 2	19
4.3 Semaine 3	19
4.4 Semaine 4	19
4.5 SEMAINE 5	19

1. MISE EN CONTEXTE

1.1 PRÉSENTATION DE L'ENTREPRISE

Dès mon arrivée à Schneider Electric à Mâcon, j'ai été briefé sur l'entreprise, ses objectifs, son domaine d'activité, et les risques présents dans l'usine. J'ai ensuite pu intégrer le service SI (Système d'Information), divisé en plusieurs parties: cybersécurité, gestion SAP/ERP et développement de logiciels internes.

1.1.1 Activité à Mâcon

Schneider Electric est un spécialiste mondial en gestion de l'énergie et en automatisation. L'entreprise propose des solutions énergétiques intégrées pour de nombreux secteurs d'activités.

1.1.2 Produits fabriqués à Mâcon

Le site de production et de logistique de Schneider Electric à Mâcon est spécialisé dans la production d'équipements de distribution électrique. Ils fabriquent des produits tels que des disjoncteurs, des interrupteurs, des tableaux électriques, etc. Le site s'étend sur plus de 30 000 m².

1.1.3 Nombre de salariés à Mâcon

Actuellement, le site emploie 520 salariés à Mâcon. Environ 80 % d'entre eux vivent dans un rayon de trente kilomètres autour de Mâcon.

Clientèle : Schneider Electric fournit des solutions énergétiques à divers secteurs, notamment l'industrie, les data centers et les infrastructures électriques. Ils ont récemment décroché une certification pour le nucléaire et fournissent des data centers dans le monde entier.

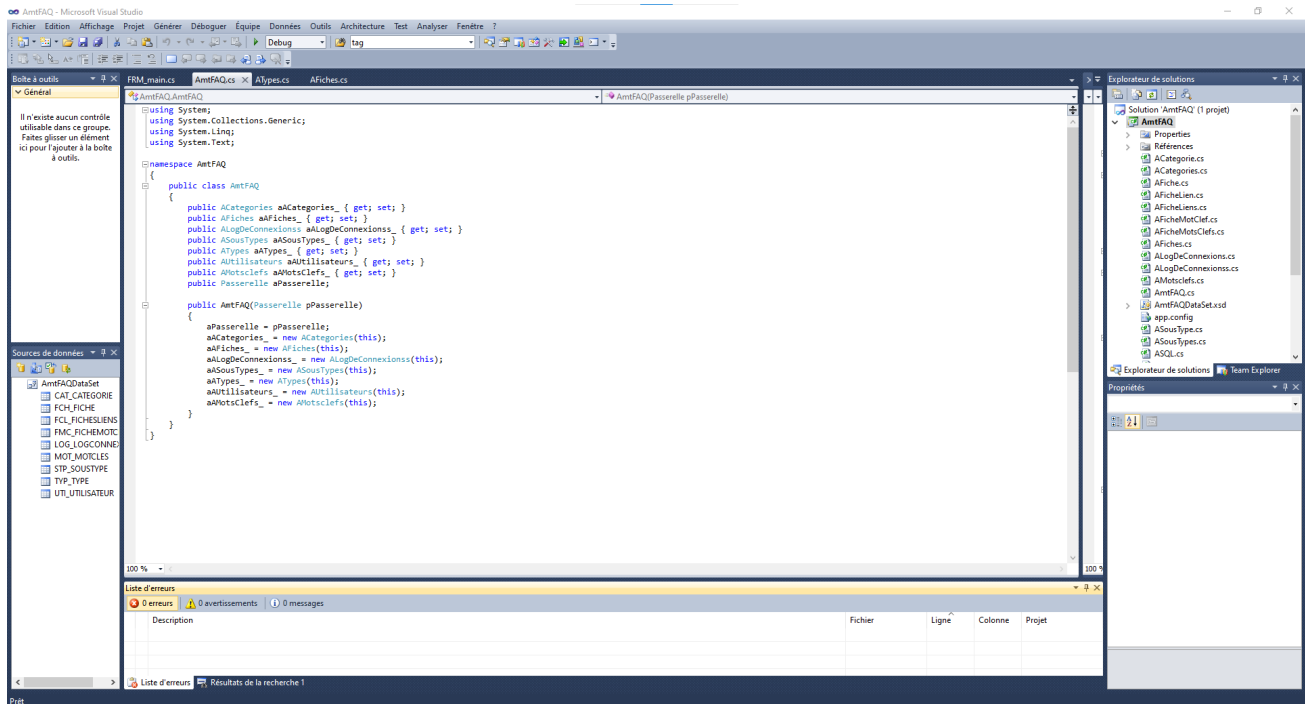
1.2 PRÉSENTATION DU PROJET

1.2.1 Objectif

J'ai pour mission de développer un logiciel qui sera utilisé dans l'entreprise pour répondre aux questions connues des employés afin qu'ils n'aient pas à faire appel à d'autres personnes, ce qui les rend donc plus autonomes. Ce logiciel devra être déployé sur tous les postes de l'entreprise car il couvrira les question des personnes dans les bureaux comme celles des personnes dans l'usine.

1.2.2 Consignes et contraintes

Ce logiciel est à développer en C#, et comprend de la programmation orientée objet, il utilise également du SQL, pour son développement, j'ai à ma disposition les logiciels Microsoft Visual Studio 2010 ainsi que Microsoft SQL Serveur Management Studio.



Fenêtre de Visual Studio 2010

1.3 ÉTUDE DU PROJET EXISTANT

1.3.1 Classe principale

On utilise la classe parent AmtFAQ pour mettre toutes les autres classes dedans, ce qui permet de retrouver plus facilement les classes et d'utiliser l'auto-complétion de Visual Studio.

```
namespace AmtFAQ
{
    20 références
    public class AmtFAQ
    {
        7 références
        public ACategories aACategories_ { get; set; }
        9 références
        public AFiches aAFiches_ { get; set; }
        1 référence
        public ALogDeConnexionss aALogDeConnexionss_ { get; set; }
        8 références
        public ASousTypes aASousTypes_ { get; set; }
        8 références
        public ATypes aATypes_ { get; set; }
        5 références
        public AUtilisateurs aAUtilisateurs_ { get; set; }
        1 référence
        public AMotsClefs aAMotsClefs_ { get; set; }
        11 références
        public Passerelle aPasserelle { get; set; }
        4 références
        public AFicheMotsClefs aAFicheMotsClefs_ { get; set; }

        1 référence
        public AmtFAQ(Passerelle pPasserelle)
        {
            aPasserelle = pPasserelle;
            aACategories_ = new ACategories(this);
            aAFiches_ = new AFiches(this);
            aALogDeConnexionss_ = new ALogDeConnexionss(this);
            aASousTypes_ = new ASousTypes(this);
            aATypes_ = new ATypes(this);
            aAUtilisateurs_ = new AUtilisateurs(this);
            aAMotsClefs_ = new AMotsClefs(this);
            aAFicheMotsClefs_ = new AFicheMotsClefs(this);
        }
    }
}
```

1.3.2 Sous-classes

Les classes incluses dans AmtFAQ sont elles-mêmes des classes parentes, voici ci-contre un exemple avec la classe ACategories, parente de ACategorie.

Ces sous-classes permettent de créer, manipuler, filtrer et modifier plus facilement les listes d'objets.

```
namespace AmtFAQ
{
    4 références
    public class ACategories
    {
        // Référence Parent
        // CK - 15/03/2024
        public AmtFAQ _aAmtFAQ;

        //Liste qui contient toute les catégories
        // CK - 15/03/2024
        private List<ACategorie> mListeACategorie = new List<ACategorie>();

        1 référence
        public ACategories(AmtFAQ pAmtFAQ)
        {
            //Type : Constructeur
            //Paramètres : Référence Parent
            _aAmtFAQ = pAmtFAQ;
            mListeACategorie = _aAmtFAQ.aPasserelle.FSqlACategorieListe();
        }

        //Ajout d'un paramètre optionnel pour que la méthode nous renvoie la 1ère catégorie de la liste (indice 0)
        // CK - 19/03/2024
        public ACategorie aACategorie_(string pCode)
        {
            return tListe(pCode)[0];
        }

        3 références
        public List<ACategorie> tListe(string pCode = null)
        {
            //Type : Liste
            //Fonction : retourne la catégorie recherché dans la liste, à l'aide de son code
            //CK - 15/03/2024

            List<ACategorie> mListeACategorie = new List<ACategorie>();
            if (pCode == null)
                return mListeACategorie;
            else
                foreach (ACategorie iACategorie in mListeACategorie)
                    if (pCode == iACategorie.aCode)
                        mListeACategorie.Add(iACategorie);
            return mListeACategorie;
        }
    }
}
```

1.3.3 Sous-sous-classes

Les sous-sous-classes sont utilisées pour créer les objets et récupérer leurs informations, comme ici avec ACategorie qui permet de construire un objet de type ACategorie avec pour attributs aCode et aLibelle.

```
namespace AmtFAQ
{
    20 références
    public class ACategorie
    {
        // Référence Parent
        public ACategories _aACategories;

        // Propriétés publiques de la classe ACategorie
        11 références
        public string aCode { get; set; }
        11 références
        public string aLibelle { get; set; }

        //Constructeur de la classe ACategorie
        1 référence
        public ACategorie(string pCode, string pLibelle)
        {
            aCode = pCode;
            aLibelle = pLibelle;
        }
    }
}
```

2. RÉDACTION D'UN CAHIER DES CHARGES

Une fois les explications données, je me suis mis à rédiger un cahier des charges afin de mieux imagier le logiciel et sa structure pour répondre au mieux aux besoins et à la demande des utilisateurs.

2.1 FONCTIONNEMENT

- Soit par recherche avec les tags / titres / questions
 - Rechercher tous les tags correspondants et si le texte ne correspond plus alors afficher les derniers résultats
- Soit par navigation dans les treeview
 - Créer des menus déroulants pour chaque grande catégorie qui contiennent leurs sous-catégories

2.2 INTERFACE

- Barre de menu
 - Bouton de retour à la première page
 - naviguer entre les pages
 - Les flèches de navigations contiennent l'historique / le chemin des pages parcourues
 - rechercher
 - Nom et prénom de l'utilisateur

2.3 ECRANS

- Recherche:
 - Afficher l'entrée de la recherche
 - Afficher les tags / catégories associées à la recherche
 - Afficher les résultats de la recherche
- Navigation à partir des treeview:
 - Afficher le nom de la catégorie
 - Afficher les tags / catégories associées à la recherche
 - Afficher les résultats de la recherche
- Consultation de fiche:
 - Afficher la question

- Afficher la catégorie
 - Afficher la réponse à la question
 - Afficher les tags / catégories associées à la recherche
 - Afficher la date de création / modification
 - Afficher l'auteur
 - Afficher un bouton pour demander une modification / vérification de la fiche
 - Afficher le bouton d'activation de la modification
 - Afficher l'icône
 - Afficher le lien du fichier associé
- Les écrans se superposent pour conserver les écrans précédents
 - les flèches permettent de naviguer dans les écrans consultés

2.4 LOGS

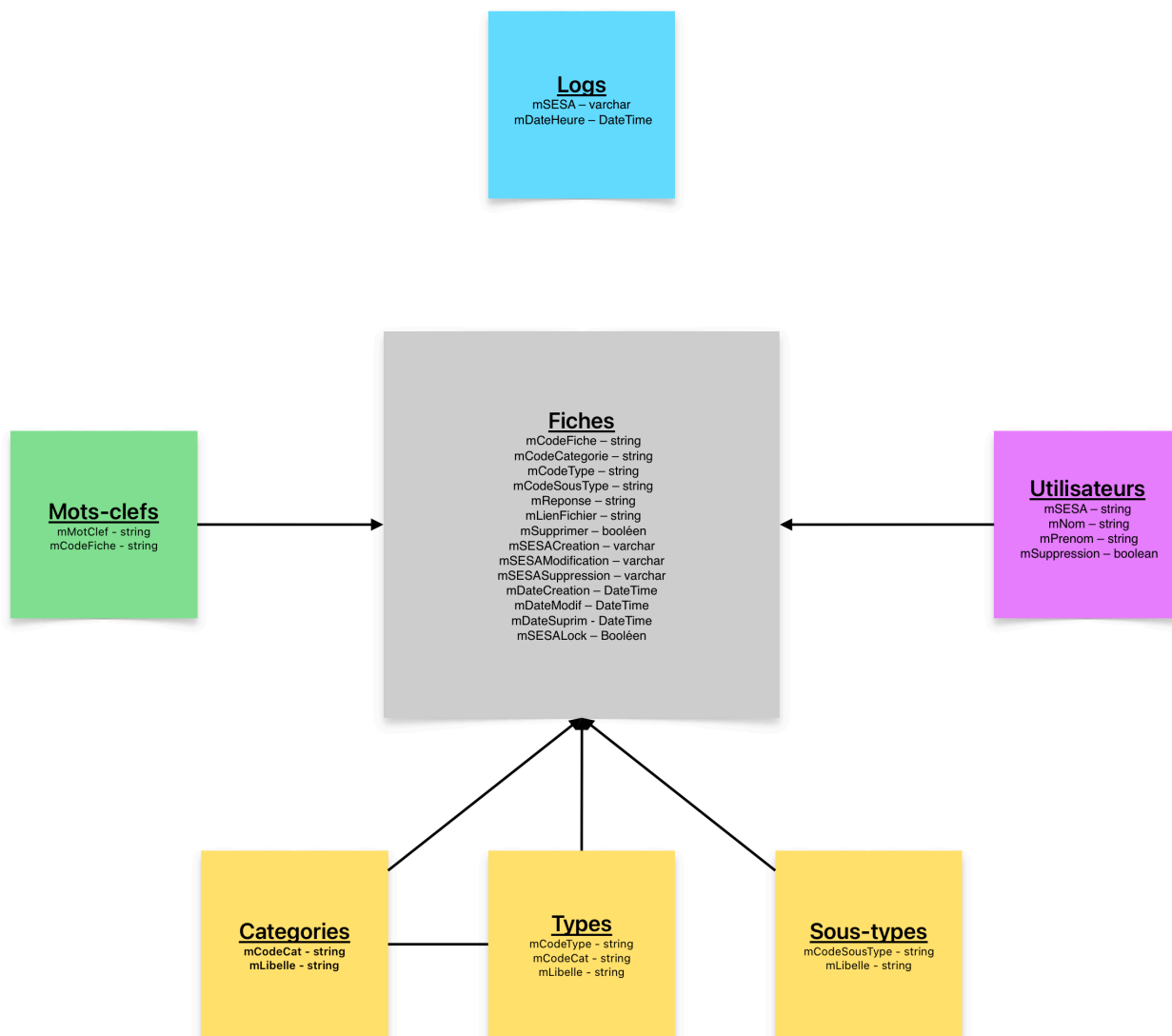
- connexion
- déconnexion
- modification du profil
- modification d'une fiche

2.5 ETAPES

1. Créer l'interface
 1. Bouton d'accueil
 2. Flèches de navigation
 3. Barre de recherche
 4. Bouton de recherche
2. Créer les fenêtres
 1. Ecran d'accueil
 2. Ecran des types
 3. Ecran des sous-types
 4. Ecran des fiches
 5. Ecran des recherches
3. Finalisation
 1. Enregistrement des logs
 2. Création de fiches
 3. Modification de fiches
 4. Autocomplétion des recherches

2.6 ANALYSE DES DONNÉES

2.6.1 Base de données



La base de données tourne autour de la table contenant les fiches.

Chaque fiche est rangée dans une grande catégorie, puis un type et un sous-type, les types dépendent des catégories mais les sous types sont communs à tous les types car ils peut y avoir le même sous-type dans deux types différents.

Lors de l’affichage d’une fiche, on propose de voir d’autres fiches qui possèdent le même mot clef dans la table Mots-clefs.

La table utilisateurs contient les utilisateurs administrateurs de l’application, stockés par leur SESA (moyen d’identification interne lors de la connexion à la

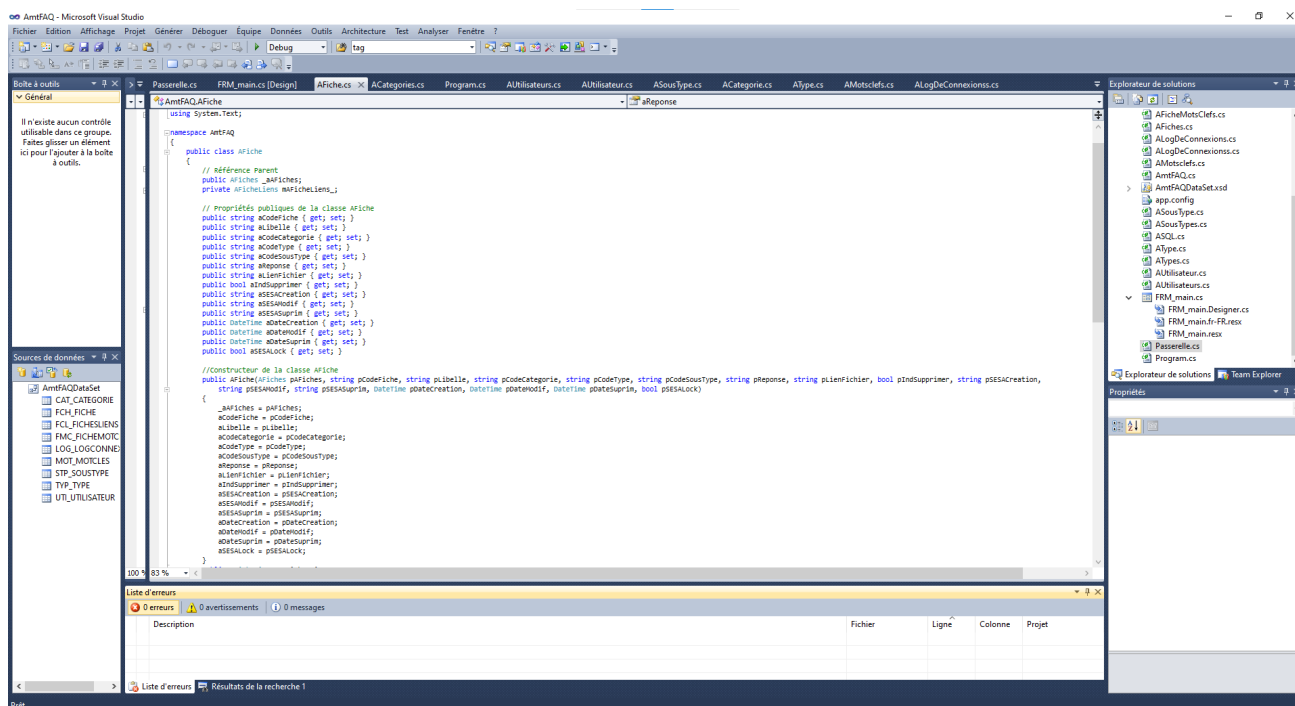
session Windows), leur nom et leur prénom.

La table de logs enregistre les connexions et déconnexions des utilisateurs.

2.6.2 Classes

aAmtFAQ

- aACategories
 - aACategorie
- aATypes
 - aAType
- aASousTypes
 - aASousType
- aAUtilisateurs
 - aAUtilisateur
- aAMotsClefs
 - aAMotClef
- aAFiches
 - aAFiche
- aALogDeConnexions
 - aALogDeConnexionss
- aPasserelle

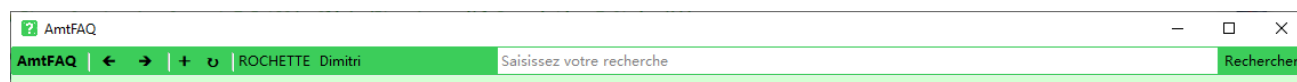


3. DÉVELOPPEMENT

3.1 CRÉATION DE L'INTERFACE

Afin de rendre son utilisation la plus simple et rapide possible, le logiciel sera présenté à la manière d'un site web, c'est-à-dire avec une interface minimaliste mais avec des pages qui changent et qui nous guident vers ce que nous cherchons.

Je crée donc une barre horizontale contenant plusieurs éléments:



3.1.1 Bouton d'accueil

Le bouton affiche 'AmtFAQ' lorsque la fenêtre d'accueil est affichée et rafraîchit les éléments à partir de la base de données, sinon il affiche 'Accueil' et envoie sur la fenêtre d'accueil lors du clic.

3.1.2 Flèches de navigation

Des flèches pour naviguer sur les pages précédentes / suivantes, elles affichent l'historique des fenêtres parcourues et renvoient sur la fenêtre sélectionnée en cas de clic sur un élément de la liste.

3.1.3 Bouton d'ajout d'une fiche

Pour les personnes autorisées (utilisateurs authentifiés dans la base de données), il y a en plus un bouton '+' qui est affiché dans la barre horizontale qui lorsqu'on clique dessus ouvre une interface pour ajouter une fiche à la base de données.

3.1.4 Bouton pour rafraîchir la page

Le bouton permet de rafraîchir toutes les données de l'application tout en restant sur la page courante.

3.1.5 Identifiant de l'utilisateur

Affiche le nom de l'utilisateur associé à la session Windows courante en utilisant une barre de recherche accompagnée d'un bouton de validation.

3.1.6 Entrée textuelle pour la recherche

Le textbox permet de faire des recherches parmi les fiches présentes dans la base de données et de les retrouver grâce à leur contenu, titre, catégorie, type, sous type ou mot clef. Lors de la saisie, un menu déroulant s'affiche pour proposer à l'utilisateur de compléter la fin de sa recherche grâce aux éléments connus dans la base de données.

3.2 PRISE EN MAIN DES ÉLÉMENTS DÉJÀ CRÉÉS

Le projet a déjà été commencé par une stagiaire avant moi, une partie du travail est fait mais reste inachevé.

Les méthodes des classes étant faite, j'ai dû recevoir des explications pour l'utilisation des nombreuses classes afin de pouvoir les exploiter au mieux.

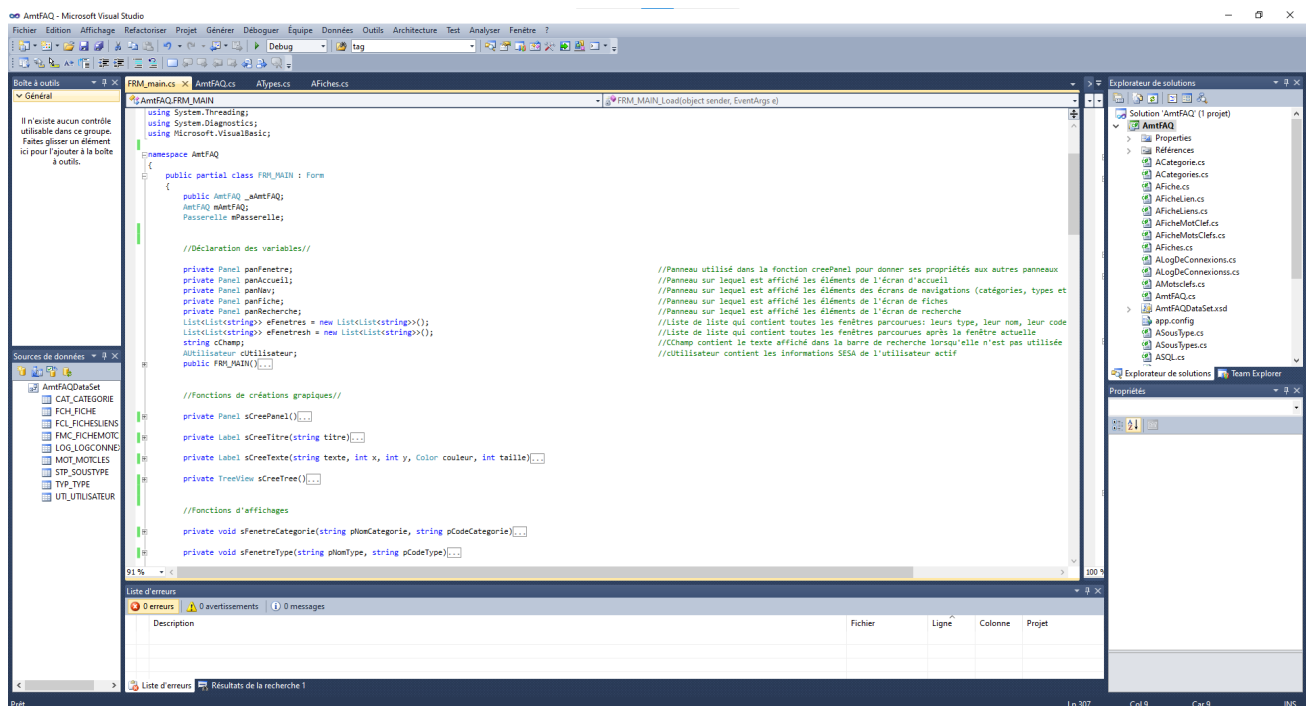
La base de données est créée mais est incomplète pour certaines lignes.

3.3 CRÉATION DES FENÊTRES

3.3.1 Méthodes

Pour permettre la navigation facile entre les pages et pour créer une harmonie entre les différentes pages, j'ai décidé de créer les pages avec du code, par le biais de fonctions, plutôt que de les créer graphiquement sur la fenêtre avec le Designer.

J'ai donc créé des fonctions qui créent les éléments graphiques (les controls) ainsi que des fonctions qui créent des des fenêtres grace aux fonctions précédemment citées, cela me permet par exemple d'afficher une fiche réponse en appelant simplement ma fonction `private void sCreeFenetreFiche(AFiche pFiche)` avec comme paramètre `pFiche`, objet de type `AFiche`.



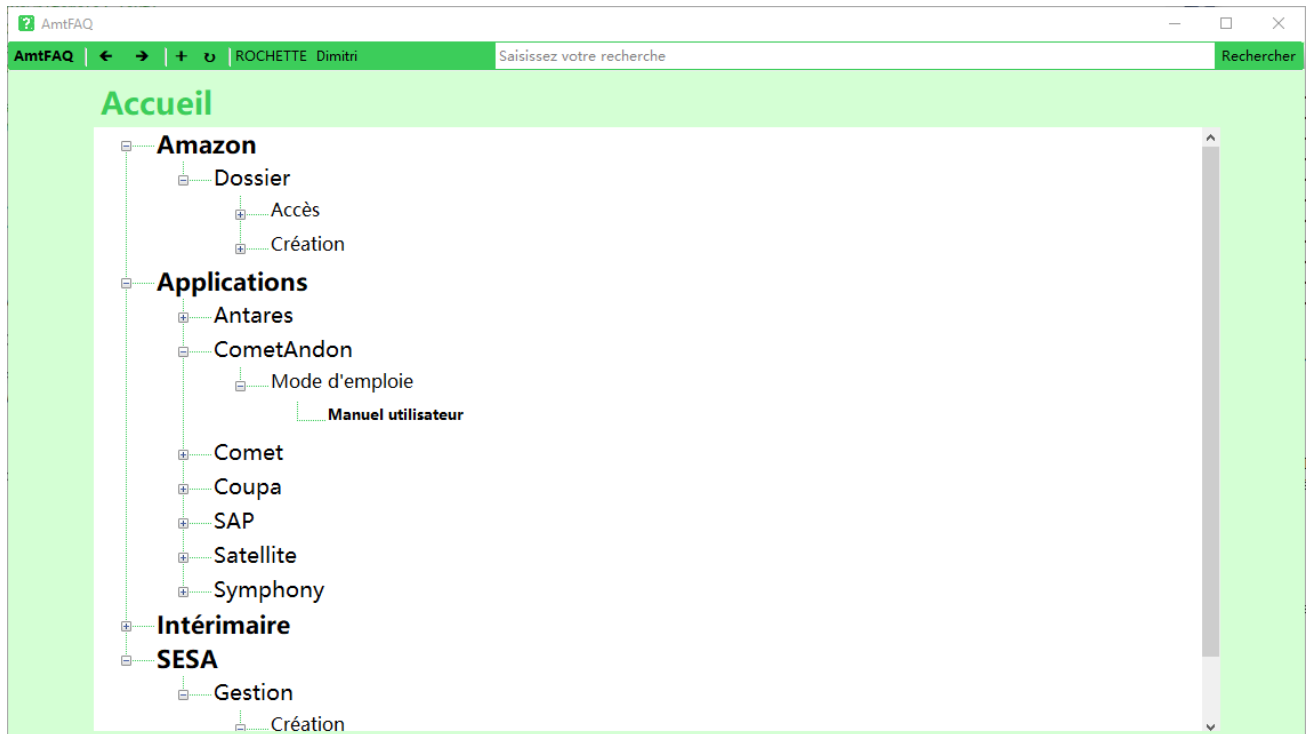
Capture d'écran de Visual Studio montrant le programme principale

3.3.2 Création de compte

Pour déterminer les droits de l'utilisateur, une vérification sur la base de données est faite au démarrage de l'application: on utilise le compte Windows de l'utilisateur pour faire le lien avec son nom et son prénom enregistré dans la base de données, si le compte n'est pas identifié, la page de création de compte s'affiche et demande le nom et le prénom de l'utilisateur pour l'enregistrer dans la base de données.

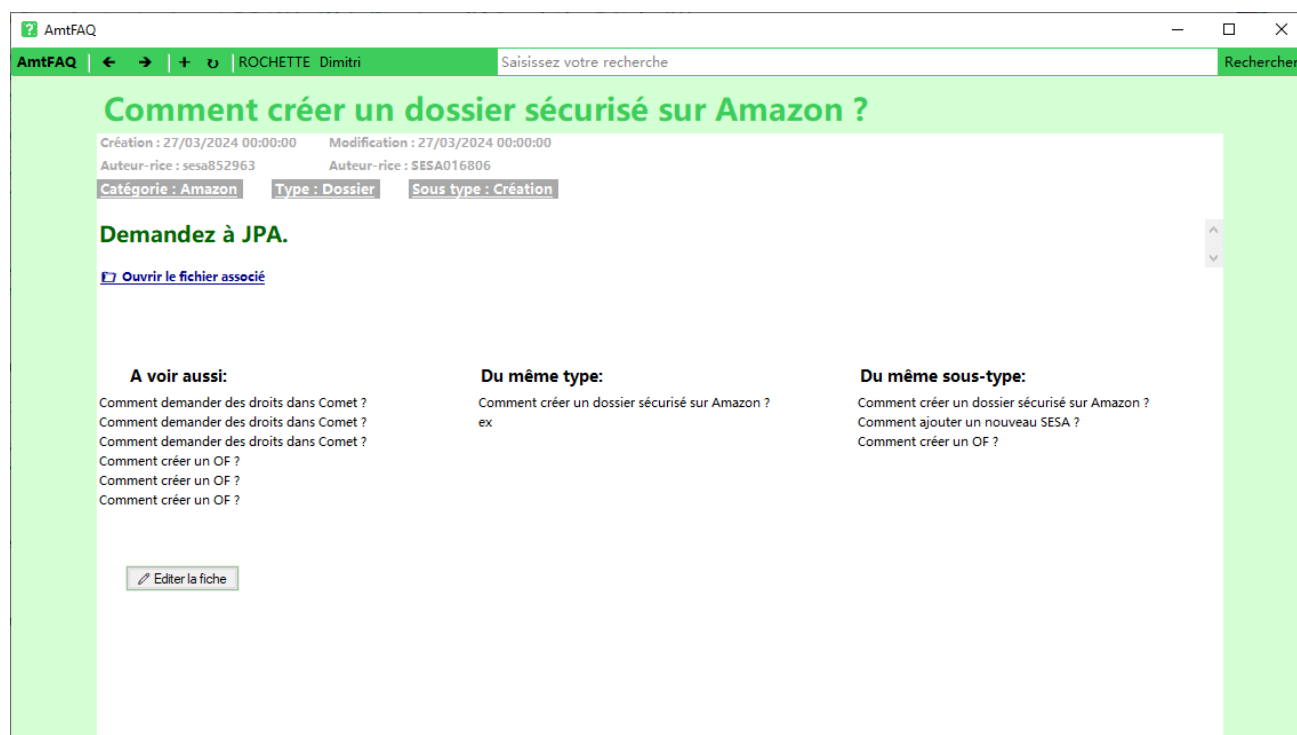
3.3.3 Accueil, Catégorie, Type, Sous type

La page d'accueil contient le treeview principal, il affiche les grandes catégories de fiches, il permet d'accéder au types contenus dans la catégories sélectionnée par un clic ou en dépliant le noeud, les fenêtres sont récursives jusqu'aux fiches.



3.3.4 Consultation de fiche

La fenêtre des fiches affiche la question, la réponse, la date de création, la date de dernière modification, la catégorie et les fiches associées afin de prévoir de potentielles autres questions des utilisateurs et leur éviter des manipulations supplémentaires. Pour les personnes autorisées (utilisateurs authentifiés dans la base de données), il y a en plus un bouton pour modifier la fiche, avec lequel il est possible de modifier n'importe quel élément de la fiche.



3.3.5 Création et édition de fiches

La fenêtre d'édition et de création des fiches permet de la manipulation/gestion/catégorisation/modification/suppression/création des fiches.

Les textes modifiables sont dans des textbox, pré-remplis lors de modifications, qui permettent de mettre le texte souhaité pour chaque propriété de la fiche.

Un bouton permet d'ouvrir une fenêtre de l'explorateur de fichiers pour sélectionner un fichier et l'associer à la fiche en cours d'édition. On peut aussi le consulter ou le supprimer.

Les checkboxes permettent de sélectionner la catégorie, le type et le sous-type de la fiche, ou bien d'en créer si besoin.

Un checkbox contenant le titre de chaque fiche permet de sélectionner les fiches associées qui seront affichées aux côtés de celle-ci.

Enfin, il y a un bouton pour enregistrer la fiche et un pour supprimer.

The screenshot shows a web application window titled 'AmtFAQ'. The browser's address bar shows 'Applications'. The page title is 'Modification: Manuel utilisateur'. The form contains the following sections:

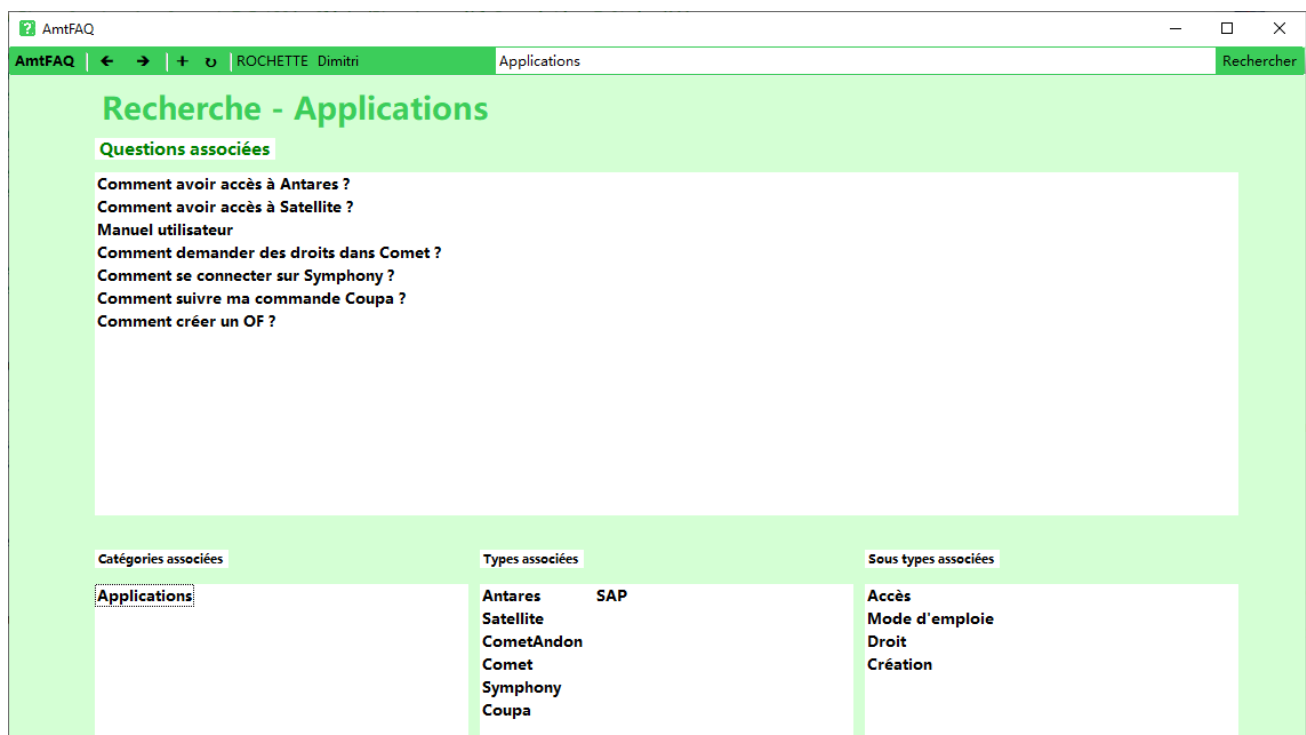
- Question:** A text input field containing 'Manuel utilisateur'.
- Réponse à la question:** A text input field containing 'Installation et utilisation de Comet Andon'.
- Fichier associé:** A section with a button 'Changer de fichier', a link 'Fichier actuel', and a button 'Supprimer le fichier associé'.
- Catégories:** A section with three columns of checkboxes:
 - Column 1: ☐ Amazon, ☒ Applications, ☐ Intérimaire, ☐ SESA.
 - Column 2: ☐ Antares, ☒ CometAndon, ☐ Comet, ☐ Coupa, ☐ SAP, ☐ Satellite, ☐ Symphony.
 - Column 3: ☐ Accès, ☐ Création, ☐ Droit, ☒ Mode d'emploi.

3.3.6 Recherche

Lors de la recherche d'un ou plusieurs termes à l'aide de la barre de recherche, des propositions d'auto-complétion sont faites pour guider l'utilisateur dans sa recherche. Une fois l'entrée de la recherche validée, une fenêtre de recherche apparait et propose tous les résultats de la recherche dans quatre listview:

- un treeview contenant les fiches associées à la recherche
- un treeview contenant les catégories associées à la recherche
- un treeview contenant les types associés à la recherche
- un treeview contenant les sous-types associés à la recherche

On peut alors sélectionner la fiche que l'on cherchait, ce qui ouvrira la fenêtre de la fiche en question, sinon le click sur un résultat dans un treeview affiche tout les résultats liés à l'élément sélectionné.



3.4 SYSTÈME DE LOGS

Pour enregistrer les logs dans la base de données, j'ai créé une fonction qui récupère comme paramètre du texte qui définit le type de logs, la date, l'heure et l'identifiant s'enregistrent automatiquement dans la base de données: la date et l'heure s'ajoutent comme valeur par défaut dans la base de données et l'identifiant se rajoute grâce à *Environment.UserName* qui récupère le nom d'utilisateur de la session.

Les logs sont enregistrés pour la connexion à l'application et la création/modification/suppression de fiches.

3.5 À FAIRE / FINIR

- Optimisation du code / éviter les répétitions
- Lien entre les fiches plus précis

4. BILANS

4.1 SEMAINE 1

J'ai pu découvrir l'entreprise et le projet à réaliser, avec la prise en main de méthodes internes et des classes déjà faites.

4.2 SEMAINE 2

J'ai pu commencer l'interface et les fonctions de création graphiques permettent la navigation des fenêtres plus simple sans utiliser le designer mais le code.

4.3 SEMAINE 3

J'ai avancé sur les fenêtres de fiches et de recherches, refait la base de données et l'architecture des catégories, types et sous-types.

4.4 SEMAINE 4

J'ai créé les fenêtres de création de compte, de création et de modification des fiches.

4.5 SEMAINE 5

J'ai pu approfondir les fonctionnalités de l'application et améliorer l'expérience utilisateur tout en ajoutant des boutons pour rafraîchir la fenêtre, ajouter les logs.