

Architecture générale

Le projet Caserne est une application Windows Forms en C# utilisant SQLite pour la gestion des missions de pompiers. L'application suit une architecture mixte utilisant :

- Mode connecté : Pour les opérations de lecture et mise à jour directes
 - Mode déconnecté : Avec DataSet pour les opérations complexes
 - Base de données SQLite : Tables Pompier, Mission, Engin, Caserne, etc.
 - Interface Windows Forms : Avec UserControls réutilisables
-

FormTableauBord.cs

Fonction principale : Tableau de bord central pour la gestion et supervision des missions

Fonctionnalités clés

Affichage des missions :

- Liste dynamique des missions dans un FlowLayoutPanel
- Chaque mission affichée via le UserControl UcMission
- Tri par ID décroissant (missions les plus récentes en premier)
- Filtre "missions en cours" via checkbox

Gestion des missions :

- Terminer une mission :
 - Ouverture d'un formulaire de saisie de compte-rendu
 - Mise à jour automatique de la date/heure de retour
 - Marquage comme terminée dans la base
- Génération de rapports PDF :
 - Création automatique de rapports détaillés
 - Inclusion des pompiers affectés et engins utilisés
 - Sauvegarde avec nom automatique (format : Rapport_Mission_ID_Date.pdf)

Navigation vers autres modules :

- Création de nouvelles missions
- Gestion du personnel

- Consultation des engins
- Statistiques
- Bouton de sortie avec confirmation

Points techniques

- Utilisation d'iTextSharp pour la génération PDF
 - Requêtes SQL complexes avec jointures multiples
 - Gestion des événements dynamiques sur les UserControls
 - Méthode statique de sauvegarde en base pour le mode déconnecté
 - Rafraîchissement automatique après modifications
-

FormAjoutPompier.cs

Fonction principale : Formulaire pour ajouter un nouveau pompier dans la base de données

Fonctionnalités clés

- Validation stricte : Contrôle de tous les champs obligatoires (nom, prénom, téléphone, bip, date de naissance, sexe, situation, caserne, grade)
- Génération automatique : Calcul du prochain matricule disponible
- Interface ergonomique : ComboBox pour jours/mois/années, casernes, grades
- Contrôles de saisie : Validation en temps réel des champs texte et numériques
- Gestion des habilitations : Attribution via CheckedListBox
- Insertion coordonnée : Mise à jour des tables Pompier et Affectation

Points techniques

- Mode connecté avec SQLite
 - Validation métier avancée
 - Interface utilisateur intuitive avec contrôles personnalisés
-

FormCompteRendu.cs

Fonction principale : Saisie simple du compte-rendu de mission

Fonctionnalités clés

- Interface minimale : Zone de texte principale avec boutons Valider/Annuler
- Validation optionnelle : Confirmation si compte-rendu vide
- Ergonomie : Raccourci Ctrl+A pour sélection totale
- Intégration : Retour via propriété CompteRendu

Points techniques

- Formulaire modal simple
 - Gestion propre des DialogResult
-

FormCreationMission.cs

Fonction principale : Création complète d'une nouvelle mission d'intervention

Fonctionnalités clés

- Saisie complète : Informations mission (motif, adresse, nature du sinistre)
- Attribution intelligente :
 - Sélection automatique des engins selon le type de sinistre
 - Attribution automatique des pompiers qualifiés disponibles
- Gestion des contraintes : Proposition de caserne alternative si ressources insuffisantes
- Validation métier : Contrôles complets avant création
- Interface progressive : Processus en deux étapes (infos → ressources)

Points techniques

- Architecture mode déconnecté avec DataSet
 - Logique métier complexe pour l'attribution automatique
 - Sauvegarde coordonnée via méthode statique
-

FormGestionPompier.cs

Fonction principale : Gestion complète des pompiers (consultation, modification)

Fonctionnalités clés

- Sélection par caserne : Filtre d'affichage par caserne

- **Affichage détaillé** : Informations personnelles, grade, dates, contact
- **Modifications autorisées** :
 - Changement de grade avec authentification
 - Gestion des affectations (changement de caserne)
 - Mise en congé
- **Interface organisée** : Navigation par onglets (infos de base/carrière)
- **Sécurité** : Authentification requise pour les modifications sensibles

Points techniques

- Gestion d'images pour les grades
 - Système de connexion intégré
 - Affichage conditionnel selon les droits
-

FormLogin.cs

Fonction principale : Authentification des administrateurs

Fonctionnalités clés

- **Saisie sécurisée** : Champs identifiant et mot de passe masqué
- **Validation** : Vérification contre la table Admin
- **Ergonomie avancée** :
 - Bouton "œil" pour révéler/masquer le mot de passe
 - Validation par touche Entrée
 - Protection contre copier/coller sur le mot de passe
- **Retour propre** : DialogResult.OK si succès

Points techniques

- Formulaire modal standard
 - Gestion des événements clavier personnalisée
 - Sécurité basique anti-copie
-

FormParcoursEngins.cs

Fonction principale : Navigation et consultation des engins par caserne

Fonctionnalités clés

- **Sélection contextuelle : Liste déroulante des casernes**
- **Navigation fluide : Boutons Début/Précédent/Suivant/Fin**
- **Affichage riche :**
 - **Numéro formaté (Caserne-Type-Numéro)**
 - **Date de réception**
 - **État de disponibilité (mission/disponible)**
 - **État technique (panne/opérationnel)**
- **Visualisation : Images des engins selon leur type**
- **Indicateurs : Compteur de position (ex: 3/12 engins)**

Points techniques

- **BindingSource pour navigation dans les données**
 - **Classe interne Engin pour structuration**
 - **Gestion d'images avec fallback**
 - **Navigation circulaire**
-

FormStats.cs

Fonction principale : Affichage de statistiques détaillées

Fonctionnalités clés

Statistiques par caserne :

- **Top 5 engins : Engins les plus utilisés en mission**
- **Heures d'utilisation : Temps total d'utilisation par engin**

Statistiques globales :

- **Interventions par type : Répartition des missions par nature de sinistre**
- **Top 10 habilitations : Habilitations les plus sollicitées**
- **Répartition du personnel : Nombre de pompiers par habilitation + non-habilités**

Points techniques

- Interface avec ComboBox de sélection caserne
 - Requêtes SQL complexes avec agrégations
 - StringBuilder pour optimisation d'affichage
 - Calculs de durées automatiques
 - Gestion des données manquantes (LEFT JOIN, COALESCE)
-

UcMission.cs

Fonction principale : Contrôle utilisateur pour affichage des missions

Fonctionnalités clés

- Affichage structuré : ID, date, caserne, description, nature du sinistre
- Propriétés publiques : Interface simple pour alimentation des données
- Boutons intégrés : Accès direct aux actions Terminer et PDF
- Réutilisabilité : Composant modulaire pour intégration

Points techniques

- UserControl Windows Forms standard
 - Interface publique claire
 - Conçu pour intégration dans FlowLayoutPanel
-

Flux de données et intégration

Architecture de données

- Tables principales : Mission, Pompier, Engin, Caserne, NatureSinistre
- Tables de liaison : Mobiliser (pompiers/missions), PartirAvec (engins/missions)
- Gestion d'état : Champs enMission pour pompiers et engins

Patterns utilisés

- Mode connecté : Lectures directes, mises à jour immédiates
- Mode déconnecté : DataSet pour opérations complexes (création mission)
- Validation : Contrôles de saisie et validation métier
- Sécurité : Authentification pour modifications sensibles

Fonctionnalités transversales

- **Gestion d'erreurs uniforme avec try-catch et MessageBox**
- **Confirmations utilisateur pour actions critiques**
- **Génération automatique d'identifiants**
- **Sauvegarde coordonnée des données liées**
- **Interface cohérente avec navigation intuitive**