Prosit 2 – Restructuration

Prosit Aller

[15/11/21 – xx/11/21]

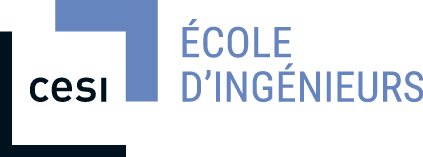


Table des matières

[1. Analyse du contexte 2](#_Toc83824561)

[2. Mots clés 2](#_Toc83824562)

[3. Problématiques 2](#_Toc83824563)

[4. Contraintes 2](#_Toc83824564)

[5. Livrables 2](#_Toc83824565)

[6. Généralisation 2](#_Toc83824566)

[7. Pistes de solution 2](#_Toc83824567)

[8. Plan d’action 2](#_Toc83824568)

[9. Réalisation du plan d’action 2](#_Toc83824569)

[10. Conclusion et retour sur les objectifs 2](#_Toc83824570)

[11. Bilan et synthèse du travail effectué 2](#_Toc83824571)

[12. Réferences bibliographiques 2](#_Toc83824572)

[12.1. Prosit 2](#_Toc83824573)

[12.2. Complémentaires 2](#_Toc83824574)

# Analyse du contexte

Après les modifications apportées, Mathilde prend en charge la partie technique du projet.

# Mots clés

- Architecture logicielle : L’architecture logicielle décrit d’une manière symbolique et schématique les différents éléments d’un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions.

- UML : Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

- C dièse : C# est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET.

- Conception orientée objet : La programmation orientée objet, ou programmation par objet, est un paradigme de programmation informatique.

- (Pattern) MVVM : Le modèle MVVM (Model-View-ViewModel) est un modèle de conception architecturale d’interface utilisateur permettant de découpler l’interface utilisateur et le code qui ne lui est pas associé. Avec MVVM, vous définissez votre interface utilisateur de façon déclarative en XAML et vous utilisez le balisage de liaison de données pour la lier à d’autres couches contenant des données et des commandes. L’infrastructure de liaison de données fournit un couplage faible qui maintient l’interface utilisateur et les données liées synchronisées, et route l’entrée utilisateur vers les commandes appropriées.

- Schéma d’architecture : Un schéma architectural est une représentation visuelle qui cartographie l'implémentation physique des composants d'un système logiciel. Il montre la structure générale du système logiciel et les associations, limites et frontières entre chaque élément.

- Plan de formation : Le plan de formation est un document établi par l'employeur au sein de chaque entreprise et qui a pour objet de recenser l'ensemble des actions de formation que l'employeur doit et entend mettre en œuvre l'année suivante (N+1), que ce soit à son initiative ou avec son accord.

# Problématiques

- Comment réaliser un schéma d’architecture type ainsi qu’un plan de formation pour la   
POO ?

- Comment cloisonner la propagation de modification lors du développement ?

# Contraintes

- Utiliser Visual Studio

- .Net core

- Git

# Livrables

- Plan de formation

- Schéma d’architecture type (MVVM)

- Diagrammes UML

# Généralisation

- Apprendre les bases du C#

- Apprendre MVVM

- Approfondir UML

- Comprendre et effectuer des schémas d’architecture

# Pistes de solution

- Utiliser une architecture MVVM

- Schéma d’architecture permettra de normaliser la rédaction du code de l’entreprise

- Effectuer un diagramme de cas d’utilisation

# Plan d’action

- Réaliser les diagrammes UML

- Faire le schéma d’architecture type

- Apprendre les bases du C# + POO

- Effectuer le plan de formation

# Réalisation du plan d’action

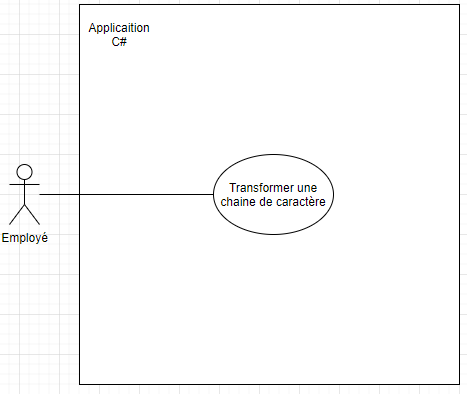


Figure 1: Diagramme Use Case

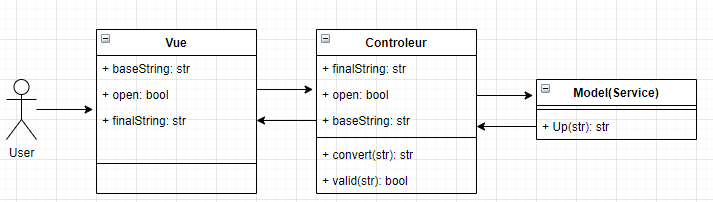


Figure 2:Diagramme de Classe

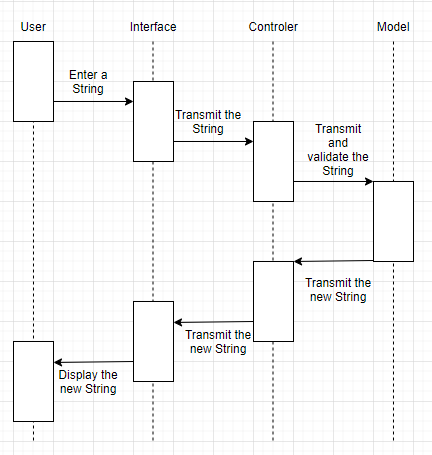
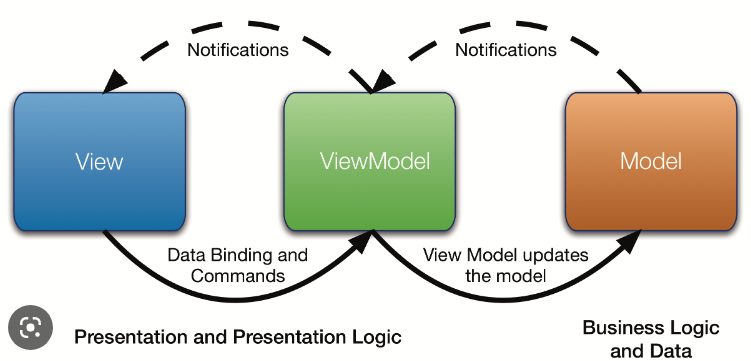


Figure 3: Diagramme de Séquence



L'idée centrale du modèle MVVM est de séparer l'interface utilisateur de la présentation et de la logique métier. L'interface utilisateur est représentée par des vues qui contiennent uniquement du code pour présenter les informations d'état telles qu'elles sont conservées et fournies par ViewModels. Les ViewModels contiennent une logique de présentation pour récupérer les données des modèles et déléguer l'exécution de la logique métier à leurs modèles.

en interface utilisateur peuvent se concentrer sur le développement d'une excellente expérience utilisateur et n'ont pas nécessairement besoin de connaissances en programmation. Les programmeurs peuvent se concentrer sur la mise en œuvre de la logique métier et de présentation.

Les View sont échangeables du fait de leur indépendance. Cela simplifie notamment le développement d'application multi-plateforme. Les tests deviendront plus faciles, car les tests peuvent se concentrer sur les viewmodel.

Le modèle MVVM a été inventé par les employés de Microsoft en 2005 avec les technologies WPF et XAML.