Guide d'étapes clés : Testez l'implémentation d'une nouvelle fonctionnalité .NET

Comment utiliser ce document ?

Ce guide vous propose un découpage du projet en étapes. Vous pouvez suivre ces étapes selon vos besoins. Dans chacune, vous trouverez :

- des recommandations pour compléter la mission ;
- les points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet (attention, celui-ci peut varier d'un apprenant à l'autre).

Suivre ce guide vous permettra ainsi :

- d'organiser votre temps ;
- de gagner en autonomie ;
- d'utiliser les cours et ressources de façon efficace ;
- de mobiliser une méthodologie professionnelle que vous pourrez réutiliser.

Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et dépendra de votre vitesse de progression.

Recommandations générales

Ce projet se fait en trois temps :

- Une partie d'analyse, de compréhension et d'initialisation (étapes de 1 à 4).
- Une partie développement par l'implémentation des tests (étapes 5 et 6).
- Une partie de vérification et de finalisation (étapes 7 et 8).

Il est nécessaire d'apporter une grande importance à l'analyse du code, la compréhension du "workflow" de l'application, avant de débuter l'implémentation des tests.

En tant que développeur, vous constaterez que le codage n'occupe qu'une petite fraction de votre temps. La majorité de votre travail consiste à analyser les spécifications et le code, à poser des questions et à effectuer des recherches.

Concernant ce dernier point, il est vivement recommandé de faire vos recherches en anglais pour obtenir des résultats plus pertinents. Plus vos termes de recherche seront précis, plus les résultats correspondront à vos attentes. Évitez d'utiliser de longues phrases dans les moteurs de recherche, privilégiez plutôt l'utilisation de mots-clés.

Un bon développeur ne prétend pas tout savoir. Il possède une solide base de connaissances, mais ne cherche pas à réinventer la roue pour chaque problème. Au lieu de cela, il sait comment rechercher rapidement et efficacement une solution à un problème que quelqu'un d'autre a probablement déjà rencontré.

Cependant, il ne faut pas simplement copier-coller des solutions. Il est essentiel de comprendre ce que vous codez. Sinon, si un bug survient en production, cela peut conduire à une catastrophe.

Pour vous aider dans votre projet, il est conseillé de suivre les cours suivants, en plus des différents tutoriels et cours proposés tout au long des étapes :

- Déboguez vos application C# ;
- Testez votre application C# ;
- Initiez-vous au test et à la qualité logicielle.

Étape 1 : Initialisez le projet et installez les dépendances nécessaires

20 % progression

Avant de démarrer cette étape, je dois avoir :

- Un ordinateur avec .NET et SQL Server Management Studio (SSMS) installés. Le projet cible .NET 6 ; si cette version du Framework n'est pas installée sur votre machine, vous pouvez la télécharger sur Download .NET 6.0 (Linux, macOS, and Windows).
- Lu le scénario et les notes de Louis, et pris connaissance des maquettes.

Une fois cette étape terminée, je devrais avoir :

- Le dépôt cloné localement sur ma machine.
- Lancé l'application .NET.

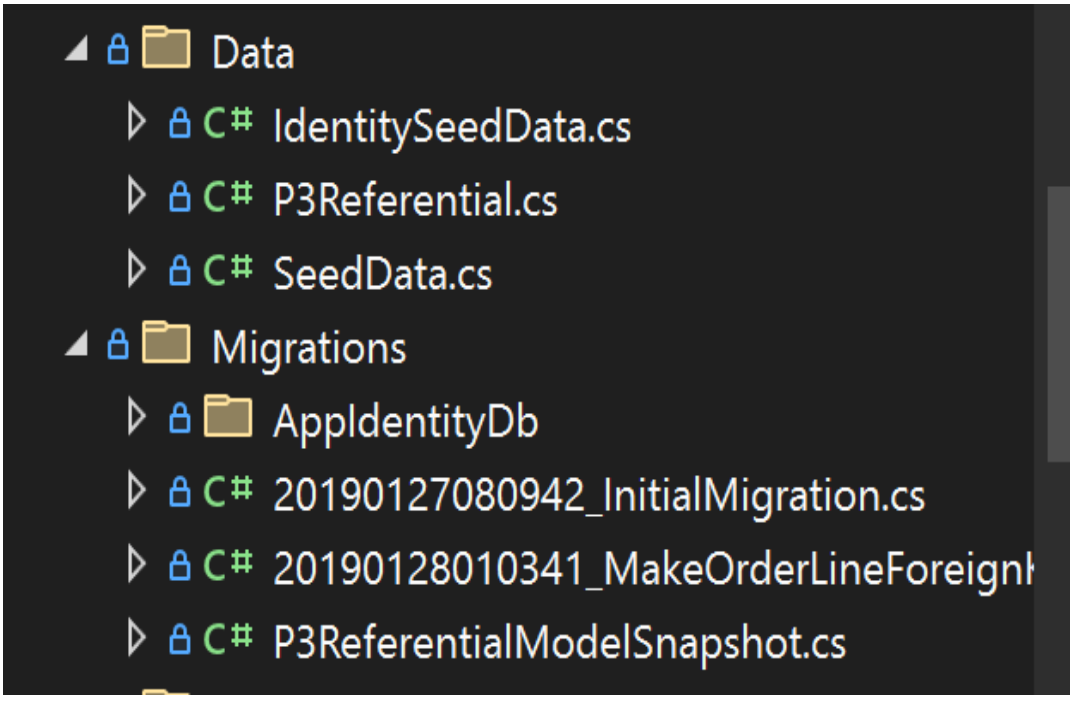
Recommandations :
Cette première étape concerne la mise en place de l'environnement de développement.

- Sur ce projet, vous allez travailler avec deux logiciels principalement :
    - Visual Studio, pour le code et le versionning sur Git ;
    - SQL Server Management Studio (SSMS), pour la gestion de votre base de données SQL.

- Démarrez le projet :
    - Clonez le repository sur votre machine depuis Visual Studio (accessible depuis la page d'accueil de VS).
    - Ouvrez le projet.
    - Suivez le README pour installer le projet et lancer l'application.
    - Lancez l'application.

Points de vigilance :

- Ne pas modifier la configuration de la base de données SQL, que ce soit via SSMS ou dans ces fichiers ci-dessous :
    - IdentitySeedData.cs ;
    - P3Referential.cs ;
    - SeedData.cs ;
    - les fichiers qui se trouvent dans le dossier "Migrations" (avec structure du nom de fichier Une date_Un nom, par exemple 20190127080942_InitialMigration.cs, qui peut être amené à changer).



Ressources :

- Le cours Gérez du code avec Git et GitHub sur OpenClassrooms vous offre un aperçu complet de l'utilisation de Git.
- Pour un guide complet sur l'utilisation de SSMS, consultez SQL Server Management Studio Overview (SSMS) (rédigé en anglais).

Étape 2 : Comprenez les principes de validation des champs du formulaire

30 % progression

Partons sur de bonnes bases. Avant de modifier le code, il est primordial de le comprendre – et de comprendre les fonctionnalités que vous devez mettre en place !

Avant de démarrer cette étape, je dois avoir :

- Le projet cloné localement et toutes les dépendances nécessaires installées.

Une fois cette étape terminée, je devrais avoir :

- Une compréhension claire de la validation des champs du formulaire dans l'application.

Recommandations :

- Vous allez travailler avec deux concepts principaux :
  - les attributs de validation en .NET pour imposer des règles sur les champs des formulaires ;
  - les messages d'erreur personnalisés pour informer l'utilisateur lorsqu'une règle n'est pas respectée.
- Voyez ou revoyez les principes de la validation des champs de formulaire en .NET.
- Identifiez où ces principes sont actuellement appliqués dans le code du projet. Posez des questions au mentor si vous en avez.
- Découvrez et servez-vous des objets prêts à l'emploi de Microsoft qui permettent de tester la validation des contraintes d'une propriété (donc les champs du formulaire) côté Back-End.

Points de vigilance :

- La validation est un aspect crucial de toute application pour assurer le bon fonctionnement des opérations entre le Front-End et le Back-End. Même si des contrôles sont exécutés côté Front-End, il est impératif d'avoir des contrôles côté Back-End. À terme, votre Back-End pourrait être appelé par différents types de Front-End que vous ne maîtrisez pas, d'où la nécessité de ce double contrôle. Par exemple, vous développez un Back-End pour gérer un site e-commerce qui est consulté depuis une application Android.

Admettons que le développeur de ce Front-End ait oublié de mettre un contrôle pour empêcher de taper des lettres dans un Input représentant un prix... ce qui induit des erreurs. Si vous avez des questions, demandez à votre mentor afin de bien comprendre les principes avant de passer à l'étape suivante.

Ressources :

- Consultez l'article Validation du modèle dans ASP.NET Core MVC et Razor Pages pour vous familiariser avec la validation du modèle.

Étape 3 : Comprenez le code existant

40 % progression

Avant de démarrer cette étape, je dois avoir :

- Une compréhension de la validation des champs du formulaire.

Une fois cette étape terminée, je devrais avoir :

- Une compréhension du code existant et des modifications nécessaires à apporter.

Recommandations :

- Explorez l'application. Cette phase est très importante, elle vous permet de vous familiariser avec le code existant.
- N'hésitez pas à déboguer l'application en mettant des points d'arrêt, puis à manipuler l'application. Cela vous aidera à vous situer dans le code en fonction des actions que vous effectuez sur le Front-End (vous cliquez sur un bouton sur le Front-End, vous voyez où vous arrivez dans le Back-End).

Points de vigilance :

- Assurez-vous de comprendre chaque partie du code avant de tenter d'y apporter des modifications. N'hésitez pas à poser des questions à votre mentor ou à effectuer des recherches sur Google.
- Toute modification du code doit être faite en conservant les conventions de codage et de nommage existantes.

Ressources :

- Les chapitres Mettez en place des points d'arrêt et Définissez des points d'arrêt conditionnels du cours Déboguez vos applications C# proposé dans les recommandations générales.

- Pour compléter vos compétences en débugging, vous pouvez consulter First look at the Visual Studio Debugger sur le site de Microsoft (rédigé en anglais).

Étape 4 : Comprenez la façon dont est géré l'inventaire

50 % progression

Avant de démarrer cette étape, je dois avoir :

- Une compréhension du code existant et des modifications nécessaires à apporter pour compléter les TODO des notes de Louis.

Une fois cette étape terminée, je devrais avoir :

- Une compréhension du fonctionnement de la gestion de l'inventaire et des critères de validation pour les produits.

Recommandations :

- Vous devrez comprendre :
  - comment les produits sont gérés dans l'application ;
  - quels sont les critères de validation pour l'ajout ou la modification d'un produit dans l'inventaire.
- Étudiez la façon dont l'inventaire est géré en parcourant l'application dans le Front-End, et également en analysant les tables dans SSMS.
- Essayez de comprendre comment cette gestion de l'inventaire est mise en œuvre dans le code actuel (quelles règles sont imposées sur ce que l'utilisateur entre dans les champs).
- Identifiez les parties du code qui gèrent l'inventaire (grâce aux différentes fonctions, en ajoutant des points d'arrêt pour déboguer, par exemple).

Étape 5 : Implémentez des tests unitaires

70 % progression

Nous voilà dans le vif du sujet ! Nous avons à présent une bonne compréhension du code nous permettant d'aborder les tests de façon plus sereine.

Avant de démarrer cette étape, je dois avoir :

- Le projet cloné localement et toutes les dépendances nécessaires installées.
- Une compréhension claire :
  - de la validation des champs du formulaire ;
  - du code existant et des modifications nécessaires à apporter ;
  - de la gestion de l'inventaire.

Une fois cette étape terminée, je devrais avoir :

- Des tests unitaires exécutés avec succès sur la base des notes de Louis.
- Une capture d'écran avec les tests réalisés avec succès.
- Le fichier des protocoles de test complété dans le template fourni.

Recommandations :

Cette étape implique l'écriture de tests unitaires pour chaque fonctionnalité décrite dans les notes de Louis. Vous devrez :

- Écrire des tests pour tous les cas de figure présents dans les notes de Louis.
- Vérifier que chaque test unitaire passe avec succès : le code fonctionne quand on lui envoie les bonnes informations… et il renvoie bien une erreur quand on lui envoie les mauvaises informations.
- Remplir le document Protocoles de test avec toutes les étapes pour mener chaque test.

Points de vigilance :

- Les tests doivent couvrir tous les cas de figure (par exemple, prenez en compte les différents langages dans lesquels l'application est disponible).
- Pensez à documenter les tests que vous mettez en place, pour faciliter la compréhension du code.
- Vérifiez que votre code respecte les normes de codage de C#.

Ressources :

- Le cours Testez votre application C#.
- L'article Conventions de codage C# courantes de Microsoft.

Étape 6 : Implémentez des tests d'intégration

80 % progression

Avant de démarrer cette étape, je dois avoir :

- Des tests unitaires exécutés avec succès.

Une fois cette étape terminée, je devrais avoir :

- Des tests d'intégration exécutés.
- Une capture d'écran avec les tests réalisés.
- Le fichier des protocoles de test complété dans le template fourni.

Recommandations :

- Cette étape concerne l'écriture de tests d'intégration pour l'application. Vous devez :
    - écrire des tests qui vérifient l'intégrité et la cohérence de l'application (attention aux notes de Louis, on est en mode Admin ici) ;
    - vous assurer que chaque test d'intégration passe avec succès.

- Identifiez les fonctionnalités de l'application qui nécessitent des tests d'intégration (par exemple, les méthodes Add, Create, Delete, etc., dans les Controllers).
- Écrivez les tests en vous assurant qu'ils couvrent toutes les interactions possibles entre les composants demandés dans les notes de Louis.
- Exécutez les tests et assurez-vous qu'ils passent tous avec succès.

Ressources :
- L'article Tests d'intégration dans ASP.NET Core sur le site de Microsoft abordant les tests d'intégration.

Étape 7 : Vérifiez et corrigez les erreurs

90 % progression

Avant de démarrer cette étape, je dois avoir :
- Des tests d'intégration exécutés avec succès.

Une fois cette étape terminée, je devrais avoir :
- Corrigé toutes les erreurs et assuré le bon fonctionnement de l'application.

Recommandations :
Cette étape consiste à vérifier le fonctionnement correct de l'application, à détecter les erreurs et à les corriger.

- Exécutez l'application et recherchez les erreurs en essayant d'ajouter des produits, en les supprimant, etc. Il faut manipuler le Front-End.
- Utilisez les outils de débogage de Visual Studio pour localiser les erreurs.
- Corrigez les erreurs et réexécutez l'application pour confirmer que tout fonctionne correctement, autant de fois que nécessaire.

Ressources :
- Le chapitre Réalisez des tests End-to-End du cours *Testez vos applications Front-End avec JavaScript*.

Étape 8 : Revue de code et préparation de la soutenance

95 % progression

Avant de démarrer cette étape, je dois avoir :
- Une application fonctionnelle sans erreurs, et produit les livrables demandés.

Une fois cette étape terminée, je devrais avoir :

- Un code propre et bien organisé, prêt à être soumis à évaluation.
- Préparé et pratiqué une présentation orale, et vérifié qu'elle répond à tous les critères communiqués dans le scénario.

Recommandations :
Cette étape finale implique la revue de code pour assurer la qualité et la propreté du code. Vous devrez :

- Passer en revue le code pour assurer sa qualité et sa lisibilité.
- Passer en revue toute la documentation associée à votre projet, y compris les notes de Louis, les commentaires dans le code, etc.
- Vous assurer que votre livrable correspond à toutes les attentes contenues dans ces documents.
- Travailler la présentation de vos tests, savoir expliquer les différents outils/objets que vous avez utilisés dans vos méthodes de test.

Vous devrez aussi préparer votre soutenance :

- Pratiquez votre présentation devant un miroir, ou devant des amis/mentors (vous pouvez passer une soutenance test avec votre mentor) qui peuvent vous donner des commentaires constructifs.
- Vous pouvez également suivre, si ce n'est pas déjà fait, le cours Prenez la parole en public.
- Assurez-vous de maîtriser votre discours, d'être clair et convaincant. Vous avez terminé ce projet, c'est que vous le maîtrisez, no stress !

Points de vigilance :

- Veillez à ne pas introduire de nouvelles erreurs lors de la modification du code. Faites un dernier check environ 30 minutes avant votre soutenance pour vous assurer que tout fonctionne.
- Assurez-vous que le code est bien documenté et facile à comprendre.
- Repassez vos tests unitaires et d'intégration après chaque refactorisation, pour vous assurer qu'il n'y a aucune régression.

Ressources :

- L'article Conventions de codage C# courantes de Microsoft sur les normes de codage.
- L'article C# Best Practices sur Medium (rédigé en anglais).
- Le cours Prenez la parole en public.

Projet terminé !