

Travail pratique 3

Application WPF et présentation

Consignes

- Le travail doit se faire individuellement
- Le travail doit être déposé sur **GitLab**.
 - o L'utilisateur **fsthilaire-dm** doit avoir le rôle **Developer**.
- Sur **LÉA**, il faut remettre un fichier **TEXTE** qui contient les éléments ci-dessous.
 - o Lien **GitLab**
 - o Nom, Prénom, DA
- Une présentation/discussion du projet avec l'enseignant pendant la semaine d'examen.
 - o La date est à déterminer (En attente de l'horaire des examens)
- La remise finale du projet sera la **veille de la présentation à 23:59:59**.
- Le fichier **TEXTE** doit être remis dans **LÉA** au plus tard la **veille de la présentation à 23:59:59**.
- La date du dernier **COMMIT** dans **GitLab** sera la date de remise du projet.

Énoncé du travail

Vous devez créer une application WPF pour faire la gestion de votre collection de jeux vidéo.

Votre application doit respecter l'architecture présentée en classe.

Le DEA de la base de données est fourni avec l'énoncé.

1- Base de données

Vous devez créer la base de données au complet par la technique **Code First**. Il est important de respecter le DEA.

Vous devez également créer vos données initiales pour toutes les tables de la base de données par des fichiers de migration. Vous devez avoir au moins 2 migrations.

Vous devez créer 2 comptes utilisateurs. Un utilisateur avec le rôle « Normal » et un utilisateur avec le rôle « Administrateur ».

L'application doit avoir un fichier **Readme** qui contient les commandes pour la migration et les comptes utilisateurs de bases.

2- Authentification et autorisation

Vous devez faire une application sécurisée.

L'application doit avoir une vue de connexion.

L'algorithme de **hachage** doit être **bcrypt**.

Chaque module doit autoriser ou non l'utilisateur en validant le rôle de l'utilisateur.

3- Modules obligatoires

Vous devez réaliser obligatoirement les modules ci-dessous.

- Utilisateur
 - Seulement le rôle administrateur a accès
- Jeu
 - Seulement le rôle administrateur a accès
- Jeu de l'utilisateur connecté
 - Tous les rôles ont accès. L'utilisateur doit être authentifié.

Il faut appliquer toutes les fonctionnalités du **CRUD**. Il doit avoir la liste et la gestion.

La modification et la création de données doivent avoir de la validation.

La suppression doit vérifier s'il y a des dépendances.

La liste doit avoir les commandes suivantes : ajouter, éditer, supprimer, rafraichir. Il faut s'assurer que les commandes peuvent s'exécuter iniquement si les critères sont réalisés.

Pour la gestion, il faut avoir les commandes suivantes : enregistrer (nouveau/modifier), rafraichir et supprimer. Pour la suppression, il faut retourner à la liste du module.

Pensez à utiliser les bons composants **WPF** pour la saisie des données.

4- Module assigné

Vous devez réaliser l'un des modules ci-dessous. **La vue vous sera assignée.**

- Type
- Plateforme

Il faut appliquer toutes les fonctionnalités du **CRUD**. Il doit avoir la liste et la gestion.

La modification et la création de données doivent avoir de la validation.

La suppression doit vérifier s'il y a des dépendances.

La liste doit avoir les commandes suivantes : ajouter, éditer, supprimer, rafraichir. Il faut s'assurer que les commandes peuvent s'exécuter iniquement si les critères sont réalisés.

Pour la gestion, il faut avoir les commandes suivantes : enregistrer (nouveau/modifier), rafraichir et supprimer. Pour la suppression, il faut retourner à la liste du module.

5- Navigation

L'application doit avoir un menu pour passer d'une liste à l'autre.

Il faut être en mesure de passer de la liste à la vue de gestion pour chacun des modules.

6- Localisation

L'application doit supporter 2 langues. La langue par défaut doit être le français. La 2^e langue est l'anglais. Il faut traduire la vue pour la liste et la vue pour la gestion. Les messages de validations doivent être traduits.

Vous devez faire la traduction du module assigné.

Vous devez également faire la traduction du menu principal de l'application.

7- Utilisation d'un API externe (Bonus)

Cette fonctionnalité est maintenant facultative.

Vous devez exporter les données de la liste des jeux de l'utilisateur connecté vers la plateforme **AirTable**. Chaque utilisateur a sa propre information de connexion à Air Table.

Il faut créer une nouvelle table à chaque exportation.

La fonctionnalité doit être dans la liste des jeux de l'utilisateur connecté. Elle doit être disponible uniquement si l'information **Air Table** est présente pour l'utilisateur.

8- Tests automatisés

Pour les tests, vous devez effectuer un test unitaire pour une fonctionnalité pour chacun des types de classes ci-dessous. **Vous devez faire les tests avec le module assigné.**

— ~~Repository~~ À ne plus faire.

- Service
- Validation
- ViewModel

Vous devez également effectuer 2 tests d'intégration. (Du **ViewModel** à la base de données).

Vous devez utiliser la technique présentée et la librairie présentée en classe.

9- Présentation/Discussion

Vous avez 20 minutes pour présenter les fonctionnalités du projet à l'enseignant.

La présentation se fait individuellement avec l'enseignant. L'heure de la rencontre vous sera assignée.

Préparez vos scénarios pour être efficace.

L'enseignant vous demandera ce qui a bien été dans la réalisation du projet et également ce qui a moins bien été.

Il faudra également justifier le choix de vos tests.

Critères d'évaluation

Le travail pratique vaut pour 45% de la note finale du cours. Ce travail est évalué sur 100 points.

| Évaluation | Pondération |
|---|-------------|
| Création de la base de données <ul style="list-style-type: none"> - Utilisation de l'approche Code First - Respect du DEA - <i>Seed</i> des données initiales - Migration fonctionnelle | 8 |
| Module Utilisateur <ul style="list-style-type: none"> - Respect des concepts présentés en classe - Utilisation adéquate des couches (Vue, MVVM, Service, Repo) - Fonctionnalités présentes et validation des données - Navigation entre la liste et la gestion - Utilisation de synchrone et asynchrone | 10 |
| Module Jeu <ul style="list-style-type: none"> - Respect des concepts présentés en classe - Utilisation adéquate des couches (Vue, MVVM, Service, Repo) - Fonctionnalités présentes et validation des données - Navigation entre la liste et la gestion - Utilisation de synchrone et asynchrone | 15 |
| Module Jeu de l'utilisateur connecté <ul style="list-style-type: none"> - Respect des concepts présentés en classe - Utilisation adéquate des couches (Vue, MVVM, Service, Repo) - Fonctionnalités présentes et validation des données - Navigation entre la liste et la gestion - Utilisation de synchrone et asynchrone | 15 |
| Module assigné <ul style="list-style-type: none"> - Respect des concepts présentés en classe - Utilisation adéquate des couches (Vue, MVVM, Service, Repo) - Fonctionnalités présentes et validation des données - Navigation entre la liste et la gestion - Utilisation de synchrone et asynchrone | 10 |
| Authentification et autorisation <ul style="list-style-type: none"> - Fenêtre de connexion - Application de la bonne autorisation selon la vue. | 7 |
| Localisation <ul style="list-style-type: none"> - Vue et message de validation - Utilisation des bonnes librairies - Les éléments réutilisables sont dans un fichier global | 5 |
| Tests automatisés <ul style="list-style-type: none"> - Pertinence des tests - Couverture adéquate du test - Nombre de tests respectés | 15 |

| | |
|--|-------------------------------------|
| Élément général <ul style="list-style-type: none"> - Documentation - Mise en forme du code - Configuration et structure de l'application - Fichier de configuration - Menu | 5 |
| Présentation et discussion <ul style="list-style-type: none"> - Déroulement des scénarios - Explication de ce qui bien été - Explication de ce qui a moins bien été - Justification des tests | 5 |
| Total | 95 |
| Utilisation d'un API externe (Bonus) <ul style="list-style-type: none"> - Utilisation adéquate de l'API Air Table - Utilisation de la technique présentée en classe Disponibilité de la commande | 5 |
| Français : Pénalité accordée au français. -1% par faute | 1 point par faute jusqu'à 15 points |
| Retard : 10% par jour de retard pour un maximum de 10 jours ou 31 mai. | -10 points par jour |

Assignation du module

Les modules ont été assignés au hasard avec <https://www.randomlists.com/>

| Étudiant | Module assigné |
|---|----------------|
| Amorim Garcia Marques Santos, Cristiane | Plateforme |
| Brodeur, Tommy | Plateforme |
| Croitoru, Vlad | Type |
| Daunais, Mathyas | Type |
| Desrosiers, Jérémy | Type |
| Forget, Eric | Plateforme |
| Letendre, Jasmin | Plateforme |
| Lizotte, Philipe | Plateforme |
| Maltais, Alec | Type |
| Marquis, Marie-Christine | Type |
| Martel-Lafleur, Alexandre | Type |
| Plouffe, Mathis | Type |
| Tardif, Etienne | Plateforme |

Description de la base de données

| Table | Description |
|----------------|--|
| Utilisateur | Les utilisateurs du système. - Les champs AirTableJeton et AirTableBaseId sont pour la communication avec AirTable. |
| Jeu | Les jeux vidéo. - La date de publication ne peut pas être dans le futur. |
| UtilisateurJeu | Table pivot qui contient la liste des jeux d'un utilisateur. |
| Plateforme | Les plateformes pour les jeux. Exemple : - Playstation 5 - Nintendo Switch - PC - Android |
| Type | Les types de jeux. Exemple : - Plateforme - FPS - MMO - RPG |
| Role | Les rôles pour la sécurité. - Administrateur - Normal |