I-CH 326+151 - Examen

L'examen de fin de premier semestre pour les modules "I-CH 151 Intégrer des bases de données dans des applications Web" et "I-CH 326 Développer et implémenter orienté objets" se présente sous la forme d'un mini-projet à réaliser seul.

L'idée est de réaliser, chacun pour soi, un projet du même acabit que Epsic.Rpg sur lequel nous avons travaillé tous ensemble durant le premier semestre.

Donnée

Le projet devra porter sur le domaine de votre choix (e-shop, jeu, blog, etc.). Une fois le thème choisi il faudra modéliser puis créer un service web permettant la gestion du domaine en question.

Par exemple: si le domaine est la gestion des assurés dans une caisse d'assurance, il faudra créer une service web permettant de gérer les assurés, les contrats d'assurance et le paiement des primes.

Pour ce faire, vous disposerez de 3 matinées de cours et du temps à la maison si nécessaire.

Contraintes

- Service web ASP.NET Core 5 écrit en C# et .NET 5
- Base de données pour persister les données (SQLite accepté et recommandé)
- EF Core pour effectuer les opérations de CRUD en base de données
- Interface Swagger pour la documentation du service
- Minimum 3 tables dans la base données
- Minimum 2 use cases à implémenter (exemple pour un e-shop gestion des articles et création d'une commande par un client)

Notation

La notation du projet se basera sur les éléments suivants :

- **(20%)** Respect des contraintes ci-dessus
- **(10%)** Diagrammes de use-case représentent bien les actions possibles pour l'utilisateur
- **(10%)** Diagramme d'activité décrivant en détail le comportant de l'action principale du domaine (exemple pour un e-shop la commande d'articles par un client)
- **(20%)** Bonne mise en application des concepts de POO appris durant le cours

    - Séparation des responsabilités (une classe s'occupe d'une partie précise de l'application)
    - Utilisation de concepts "avancés" tel que héritage, composition, interfaces, encapsulation, injection de dépendances, etc.
    - Code découplé et testable facilement
- **(10%)** Utilisation des bons verbes HTTP pour chaque action (GET pour la lecture, POST pour l'écriture etc)
- **(10%)** Utilisation des bons status codes HTTP pour chaque action (200 OK, 404 NotFound, etc.)
- **(10%)** Tests automatisés, au moins une dizaine de **tests utiles**
- **(10%)** Au premier checkout, les commandes `dotnet restore`, `dotnet build` et `dotnet test` devront fonctionner et les tests devront passer

Chaque élément vaut 3 points qui seront attribués seront la règle suivante :

- 3 points = fonctionnalité attendue
- 2 points = fonctionnalité acceptable avec quelques erreurs
- 1 point = fonctionnalité incomplète
- 0 point = fonctionnalité inexistante

À fournir

D'ici le dimanche 31 janvier 2021 23h59, me faire parvenir par e-mail à selmir.hajruli@eduvaud.ch les éléments suivants :

- Lien vers le code source sur GitHub accessible en public ou à l'utilisateur "Selmirrrrr"
- Brève documentation avec les différents diagrammes demandés et autres explications si vous jugez nécessaire d'expliquer par écrit certains éléments ou choix techniques de l'application

Chaque tranche de 12h de retard entrainera une pénalité de 0.5 point sur la note finale.

Pour finir, une présentation du projet aura lieu le 4 février 2021 devant la classe (15 minutes).