

**SESSION Août 2020**

***Bachelier en Informatique de Gestion***

*Catégorie économique*

Nom : UE : 301

Prénom :

Groupe : Bloc : 3

Nom de l’enseignant : PALERMO O.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Surveillance de la consommation d'aliments

1. Objectif

L’objectif de cet examen sera de développer une application Web permettant à l’utilisateur de surveiller son alimentation.

L'utilisateur pourra enregistrer ce qu'il consomme selon une date qu'il aura lui-même précisée. Après avoir encodé une date, il devra sélectionner les aliments qu'il a ingérés.

La personne aura un certain contrôle sur ce qu'il encode : il pourra rajouter des aliments consommés et en supprimer. S'il dépasse les 400 points sur une journée, il recevra un indicateur visuel qui l'avertira.

Un aliment possède un nombre précis de points. Ainsi, le logiciel sera donc capable de révéler le nombre total de points qui ont été épuisés.

La création de l'application se déroulera en deux phases : un serveur développé en C# et ASP.NET Core proposera un Web API pour exposer ou modifier les informations contenues de la base de données (en SQL Server). Quant à l’interface graphique client, celle-ci sera codée avec l’aide du framework Angular.

1. Consignes pour rendre les projets

Deux projets seront créés : l’un en C#, ASP.NET Core, SQL Server[[1]](#footnote-1) et l’autre en Angular. Ceux-ci devront respectivement se trouver dans deux dossiers compressés "examenASPNOMPrenom" et "examenAngularNOMPrenom".

Avant de mettre en ligne vos fichiers sur la plateforme ConnectED, vous prendrez soin de compresser vos dossiers et **d’exclure le dossier node\_modules** de votre projet Angular.

1. Caractéristiques de l’abstraction des données

Un aliment possède un identifiant, un nom et un nombre de points (entier).

Une date de surveillance possède un identifiant et intègrera naturellement la notion de dates.

Un aliment consommé possède un identifiant et fera le lien entre les deux éléments précédemment cités.

1. Cas d'utilisation à produire

L'utilisateur devra être capable de :

* Consulter le nombre total de points selon une date sélectionnée ;
* Recevoir une alerte s'il dépasse les 400 points sur une journée ;
* De gérer les dates :
  + Pouvoir en ajouter ;
  + Pouvoir les lire.
* De gérer les aliments consommés selon une date sélectionnée :
  + Pouvoir en ajouter ;
  + Pouvoir les lire ;
  + Pouvoir en supprimer.
* De gérer les aliments :
  + Pouvoir en ajouter ;
  + Pouvoir les lire ;
  + Pouvoir les modifier.

1. Points techniques à réaliser

C# & ASP.NET Core

* **Création d’un fichier SQL qui contiendra les instructions pour créer les tables et les remplir de données fictives (ce dernier ne doit pas être généré automatiquement par SSMS);**
* Conception des différentes classes ;
* Développement de DAO pour faciliter l’accès aux informations de la base de données ;
* Proposition d’un Web API qui exposera les points d'entrées suivants[[2]](#footnote-2) :
  + GET sur /api/aliments : permet de lister tous les aliments ;
  + POST sur /api/aliments: permet de créer un aliment ;
  + PUT sur /api/aliments/{id} : permet de modifier un aliment (son nom ou ses points) ;
  + GET sur /api/dates : permet de lister toutes les dates ;
  + POST sur /api/dates : permet de créer une date ;
  + POST sur /api/dates/{id}/aliments : permet d'enregistrer les aliments qui ont été consommés ;
  + DELETE sur /api/dates/{id}/aliments/{id} : permet de supprimer un aliment consommé appartenant à une date précise.

1. Angular

* Création d’une interface graphique donnant la possibilité à l’utilisateur d'opérer toutes les fonctionnalités citées plus haut.
* Élaboration d’un ou plusieurs « services » pour opérer les appels sur la base de données et des « components » pour ficeler le tout ;
* Utilisation d’une hiérarchie évolutive : « smart + dumb component » accompagnée d’un respect des bonnes pratiques de la gestion des différents dossiers ;
* Intégration de l’ergonomie : recevoir des feedbacks lors des suppressions, ajouts, etc. Le rafraîchissement automatique des données doit être exercé. À savoir : le rafraîchissement manuel est pénalisé !

1. **Le fichier SQL se trouvera dans le dossier compressé "examenASPNOMPrenom"** [↑](#footnote-ref-1)
2. Le non-respect de la structure des routes sera pénalisé. [↑](#footnote-ref-2)