DOCUMENT DE DÉFINITION

KORALLIA FRENETTE  
WILLIAM LEMIRE

TRAVAIL PRÉSENTÉ À JEAN-CHRISTOPHE DEMERS  
DANS LE CADRE DU COURS **420-C61-IN – PROJET SYNTHÈSE**

TECHNIQUES DE L’INFORMATIQUE  
CÉGEP DU VIEUX-MONTRÉAL  
08/09/2022

# Présentation générale

Nous développons une application web pour aider les particuliers qui pratiquent des hobbies à gérer la vente ou l’échange de leurs créations à d’autres particuliers. La vision pour ce projet serait de créer une plateforme où le monde peuvent échanger leurs produits directement à d’autres particuliers, mais à cause des contraintes de temps, nous nous limiterons à développer un conteneur qui peut contenir des modules indépendants les uns des autres ainsi que les modules détaillés ci-bas.

# Présentation détaillée

Dans le cadre de la relance économique post-Covid au Québec, nous voulons créer une application web, en Java, qui vise à aider les gens à vendre ou échanger leurs produits ou services avec d’autres particuliers, grâce à une application web faite avec le framework Spring.

L’application TrOqc comportera plusieurs modules, dont un module de création de compte, un module de lobby, un module de finances, un module de gestion d’inventaire, et un module de ventes/échanges. Dans l’optique de faire du développement Agile, puisque le temps est plutôt restreint, nous allons développer le module de création de compte, le moule de lobby, le module de finances, ainsi que le module de gestion d’inventaire. Même si le module de ventes/échanges ne sera pas créé, l’application restera tout de même utilisable puisque l’usager peut manuellement entrer des factures qui seront comptabilisées dans les finances et qui soustrairont les produits/matériaux de l’inventaire.

## Module de Connection/Création de compte

Ce module est le point d’entrée de l’application. Après avoir accédé à la page d’accueil et d’avoir sélectionné le bouton « Créer compte », l’usager sera envoyé au module de création de compte. De là, il ou elle devra entrer ses informations, dont :

* Nom et prénom
* Adresse courriel
* Mot de passe
* Nom de compte
* Adresse
* Catégorie des produits ou services vendus
* Lien vers le site web (optionnel)
* Logo de compagnie (optionnel)

Après la vérification de l’information dans les champs, si tout passe, l’usager sera ajouté à la table Usagers dans notre base de données, permettant d’accéder au reste de l’application grâce au courriel ou au nom de compte et le mot de passe insérés plus tôt.

## Module de lobby

Après l’authentification, l’usager est redirigé vers le module lobby, qui comprend des boutons qui permettent de rediriger vers les autres modules. Il y aura aussi une fonctionnalité de bloc-notes, qui affichera dans un menu déroulant des notes (Post-it) que l’usager peut se laisser et qui seront sauvées dans la base de données. Il pourra aussi les modifier ou les supprimer.

## Module de finances

En un coup d’œil, nous voulons permettre à l’usager de voir ses dépenses/revenus pour une période spécifiée afin de savoir s’il ou elle est dans le positif. Afin de faire ce calcul, nous allons comptabiliser plusieurs informations dont :

* Le coût des produits et matériaux insérés dans la base de données lors de la journée
* Les ventes de produits (insérées manuellement)
* Toutes autres dépenses (insérées manuellement)

Puisque nous n’aurons pas encore développé le module de ventes, il est important de pouvoir manuellement insérer des ventes de produits qui sont dans l’inventaire afin de considérer ces revenus. L’ajout d’une vente dans ce module déduira la quantité appropriée du produit dans l’inventaire.

En calculant ces chiffres, nous allons déterminer le profit net de chaque journée et le sauver dans la base de données pour pouvoir afficher à l’écran les totaux pour la période de temps spécifiée. Grâce à ces points, nous pourrons faire des projections sur les ventes et proposer des projection pessimiste et optimistes des ventes. Un algorithme déterminera quel modèle de régression linaire et non-linéaire s’applique le mieux grâce au coefficient de détermination, et la projection sera faite avec la régression appropriée. L’usager pourra, aussi, exporter dans un fichier Excel tous les résultats mentionnés.

## Module de gestion d’inventaire

Selon la perspective de développement Agile, ce module est le pilier de notre logiciel, puisqu’on ne peut vendre de produits en ligne sans inventaire. Le module aura plusieurs fonctionnalités, dont :

* L’insertion de produits
* L’insertion de matériaux (servant à faire des produits)
* La modification/suppression de produits/matériaux
* La génération automatique de SKU (code de référence) si l’usager n’en a pas
* La génération automatique d’un code barre grâce à des librairies externes
* Des alertes si certains produits ou matériaux descendent sous une certaine quantité spécifiée par l’usager
* Des alertes lors de suppressions/modifications
* La création de templates de produits (le template spécifie la quantité de matériaux requis, donc l’insertion de ce produit va déduire les matériaux associés automatiquement)
* La modification/suppression de templates
* L’exportation en fichier Excel ou .csv de l’inventaire

Nous allons utiliser une base de données no-SQL avec neo-4j pour laisser l’utilisateur créer ses propres champs descriptifs pour ses produits. Par exemple, il peut créer un champs couleur pour son produit chandail et, alors notre base de données va rajouter ce champ pour décrire ce produit en particulier et l’afficher.

## Module de Configurations

Ce module sert à modifier les informations relatives au compte de l’usager. Tous les champs seront modifiables, sauf le nom du compte. Le mot de passe nécessite l’insertion de deux strings identiques qui respectent une expression régulière.

# Contraintes applicatives

La plus grande contrainte quant à la fonctionnalité de notre projet est que nous voulons développer un logiciel complètement modulaire, c’est-à-dire, un conteneur qui permet d’ajouter n’importe quel module puisqu’ils sont autonomes. Donc, le lobby de notre application est essentiel, puisqu’il pourra contenir nos modules et en rajouter à n’importe quel moment.

Dans la même ligne d’idées, nous voulons que les modules se gèrent eux-mêmes, qu’ils n’aient à se concentrer que sur leur logique, donc tout ce qui est externe aux modules, comme l’accès à la base de données, sera géré par des classes utilitaires.

# Plateforme ciblée

La plateforme ciblée pour notre logiciel est le web.