Andrew Deeley

Cégep de Chicoutimi  Hiver 2023

Épreuve Synthèse de Programme – Changements au cahier des charges

Table des matières

[Contexte 2](#_Toc129615970)

[Changements dans la conception de la solution 2](#_Toc129615971)

[Caisse Enregistreuse 2](#_Toc129615972)

[SGI 2](#_Toc129615973)

[Diagrammes 2](#_Toc129615974)

[Autre 3](#_Toc129615975)

[Modifications aux plans de tests 3](#_Toc129615976)

[Tests d’intégration 3](#_Toc129615977)

[Tests unitaires 3](#_Toc129615978)

[Tests fonctionnels 3](#_Toc129615979)

[Modifications des interfaces 4](#_Toc129615980)

# Contexte

Ce document fera le tour des modifications apportés à la conception du logiciel selon le cahier des charges émis le 6 février. Il sera divisé en trois parties : changements dans la conception de la solution proposée, modifications aux plans de tests et ajouts d’interfaces.

# Changements dans la conception de la solution

## Caisse Enregistreuse

|  |  |
| --- | --- |
| **Énoncé original** | **Implémentation réel** |
| *Pour certains articles, le prix du produit est déterminé par son poids ($/lb)* | Il les produits unitaires et les produits en poids partageant la même valeur car c’est l’unité qui gère le prix de vente. Exemple :  $13.99 l’unité  Unité = 1 boite -> $13.99/boite  Unité = 1 lb -> $13.99/lb |
| *Pour les produits ayant ce genre de caractéristiques [de poids], une fois le produit balayé, la caisse permet de peser le produit* | Aucune fonctionnalité n’a été implémenter pour permettre de peser un produit. Seulement une quantité peut être entrée manuellement. |
| Non-mentionnée :  La gestion des transactions | Un code-barre ajoute une ligne dans la transaction. Cependant, la fonction a été programmée de façon à permettre la possibilité future d’ajouter des codes personnalisés tel que des coupons-rabais facilement. |

## SGI

|  |  |
| --- | --- |
| **Énoncé original** | **Implémentation réel** |
| *Les rapports seront générés sur demande par le gestionnaire utilisant le logiciel.* | **Précision :** les rapports sont générés à même le logiciel, dans un menu « Rapports ». À l’instant, il n’est pas possible d’exporter les rapports, mais la fonction a été implémentée de sorte à rendre ceci possible facilement. |
| Non-mentionnée :  L’ajout et la modification de départements et fournisseurs | Lors de l’ajout et la modification d’un produit, il est possible de créer sur le moment un nouveau département ou fournisseur directement dans le champ de sélection. Aucun besoin d’un sous-menu pour la gestion des départements et fournisseurs, ils sont gérés à même le produit. Si le département ou fournisseur n’existe pas, il sera créé et affecté correctement. |

## Diagrammes

Pour les classes, tout les DTO et le DTOConverter ont été retirés en faveur d’utiliser directement les modèles fournis par le cadriciel.

## Autre

Lors du cahier, il était mentionné que le support pour un balayeur de codes-barres serait implémenté si le temps le permet. Cette fonctionnalité a été ajoutée et le logiciel supporte les balayeurs de codes-barres.

L’application a du support pour le français et l’anglais grâce a un fichier de localisation. Si l’utilisateur ouvre le logiciel sur un ordinateur avec un système d’exploitation en anglais, tout les textes de l’application seront en anglais, et vice-versa pour le français.

# Modifications aux plans de tests

À première vue, les plans de tests avait comme objectif de vérifier l’intégrité des données entre la bases de données et le cadriciel, assurer le bon fonctionnement de la création des transactions et le lien entre l’interface et le code de gestion. Au cours du projet, le but du plan de tests c’est avérée peu pratique.

## Tests d’intégration

Les tests d’intégration ont été éliminés du projet. Le plan initial était de créer les modèles via le cadriciel et d’effectuer des migrations pour la création de la base de données. Cependant, le livrable #2 nous a imposé de créer les scripts de création de base de données. Ceci a renverser l’idée a l’envers; j’ai opté de créer mes modèles a partir de la base de données. Donc il n’y avait aucune possibilité que les modèles ne concordent pas. Les tests étaient n’était donc plus nécessaires.

## Tests unitaires

Suivant le même principe que les tests d’intégration, vu que les produits sont gérées par le cadriciel, il est devenu inutile d’implémenter des tests. Les développeurs du cadriciel ont déjà faites leur preuves et tout problème liée a la gestion des modèles ne seraient pas capturés par les tests unitaires. Entre autre, il reste un test qui vérifie l’ajout d’un produit à la facture (et donc vérification de quantité en inventaire, etc.)

## Tests fonctionnels

Initialement, je voulais effectuer des tests sur la logique de changement de pages. Par contre, après quelques recherches, il est difficile de tester ce genre de comportement car l’implémentation est très encapsulée. Il aurait fallu que je modifie le code de production simplement par principe de pouvoir implémenter des tests. Ceci avait comme impacte de compliquer le code et le rendre moins maintenable par des développeurs futurs.

# Modifications des interfaces

Certains ajouts et modifications ont eu lieu, notamment :

* SGI
  + Filtres n’ont pas été implémentés
  + Un bouton pour accéder au menu des rapports a été ajoutée
  + La pagination n’a pas été implémentée
* Caisse
  + L’endroit qui affiche la facture a été rapetissée vers la gauche
  + L’espace libérée à droite affiche maintenant :
    - Sous-total
    - Total de la TPS
    - Total de la TVQ
    - Total après taxes
  + La barre d’entrée du CUP a été déplacée à gauche, en haut des modes.
  + Un bouton « annuler la facture » a été ajoutée à gauche du bouton « Terminer »