

Génie logiciel

Projet de développement

Professeur : HO Tuong Vinh

Date énoncée: 10/08/2017

Date de remise : 20/08/2017, avant 8h00

Nom de l'équipe: « Équipe GSR », ABOUBACAR DJIBO, MEDOU Daniel, ANDRE Perrault

Nom proposé pour le projet / logiciel: Mise en place d'une application de gestion de stock : Cas d'un restaurant. « GSR » qui veut dire **Gestion de Stock d'un Restaurant**

1. Une courte description du logiciel :

Notre application devra être capable de gérer un stock c'est-à-dire contrôler les entrées et sorties de stock, de signaler ou d'alerter la baisse du stock selon un seuil minimal déterminé, de générer les inventaires, de gérer la caisse, gérer le personnel, générer les factures, imprimer la liste des produits en stock, des ventes, des états de la comptabilité, et du personnel.

2. Pourquoi voulez-vous développer ce logiciel ?

La raison du développement du logiciel sera présentée en deux volets. Côté client ou utilisateur et côté développeur.

- ✓ Côté utilisateur, l'application permettra à l'utilisateur (client) de faire une gestion optimale et plus facile de son restaurant. C'est-à-dire avoir une vue globale et détaillée des activités son restaurant, faire un suivi journalier par exemple de son stock ce qui lui permettra de prendre des meilleures décisions pour améliorer la qualité de service et même des activités de son restaurant.
- ✓ Du côté développeur, l'implémentation du logiciel permettra à l'équipe de développement de mieux faire assoir ces connaissances en programmation Orientée Objet avec comme langage de programmation Java EE, l'utilisation des Framework comme Spring, Hibernate, etc., l'utilisation d'un IDE libre et Open-Source comme Éclipse ou Netbeans.

3. Les logiciels similaires existants

Il existe plusieurs logiciels de gestion de stock mais, deux (**GSM** et **SoftCaisse-Restaurant**) ont attiré notre attention de par leur simplicité et des fonctionnalités qui nous seront nécessaire pour la réalisation de notre projet. Nous avons :

- ✓ **Gestion de Stock Modulaire (GSM)** propose à l'utilisateur de gérer les stocks de marchandises en toute simplicité. Le logiciel gère les entrées/sorties directes des stocks mais peut également les planifier. Il s'occupe également de la planification des fabrications et de la valorisation CUMP. Enfin, on retrouve dans **Gestion de Stock Modulaire** une organisation classique avec référence/quantité/valeur et lorsqu'un seuil minimum de quantité s'en voit atteint, une alerte apparaît en colorisant la case en question. Disponible au : www.logiciel-gestionstock.fr.
- ✓ **SoftCaisse-Restaurant**, est une application qui a pour but de gérer les articles, fournisseurs, ventes, livraisons et client d'un restaurant. Il dispose de plusieurs fonctionnalités nous permettant de réaliser notre projet. Disponible au www.ami-web.com

4. Marché / utilisateurs cible et potentiel

Les potentiels utilisateurs du logiciel sont principalement les chefs d'entreprise de restauration et même les gestionnaires des établissements hôteliers peuvent également utiliser cette application pour le suivi de leur service de restauration. Avec la maintenance évolutive que pourra subir notre application, les clients aussi pourront utiliser cette dernière via leur mobile pour passer des commandes

5. Technologies: outils / technologies à utiliser

Pour la réalisation du logiciel, les outils à utiliser sont les suivants :

- ✓ Un ordinateur portable avec système d'exploitation Linux installé version 16.04;
- ✓ Eclipse comme IDE Open-Source nous permettra d'écrire notre programme;
- ✓ Plate-forme Java EE pour assurer le côté web de notre application;
- ✓ Framework libre Spring pour construire et définir l'architecture de notre application;
- ✓ MySQL & PhpMyAdmin pour la gestion de la base de données ;
- ✓ Framework libre JUnit pour des tests unitaires de notre application ;
- ✓ Framework libre Hibernate pour gérer la persistance des objets en base de données relationnelle

6. Risques potentiels

Du point de vue des risques, ils en sont nombreux mais nous pouvons énumérer quelques un tels que :

- ✓ Les risques d'ordre techniques : L'utilisation d'un nouvel environnement de programmation (Framework cités dans la section 5.) pas véritablement maîtrisé par tous les co-auteurs du projet;
- ✓ Les risques d'ordre humains : Éventuel incompétences des co-auteurs par rapport aux tâches qui leur ont été assignées. Nous mentionnons aussi le cas de maladie ou de l'indisponibilité de co-auteur en plein avancé du projet;
- ✓ Les risques liés aux délais : La mauvaise estimation liée au temps d'exécution des tâches peut nous conduire au non-respect des contraintes de délai.