Université d'Ottawa · University of Ottawa

SEG3503 ASSURANCE DE LA QUALITÉ LOGICIELLE
Été 2020

Devoir 3

Date d'échéance: le 24 Juillet à 23h59 (remis à travers BrightSpace)

Le devoir est à faire INDIVIDUELLEMENT ou en équipes de deux étudiants MAXIMUM. Vous devez remettre une archive .ZIP contenant vos réponses aux questions ainsi que votre script de test et autres éléments de code qui le supportent (stubs, mocks, scripts d'installation de base de donnée, etc.).

Objectif

Le but de ce devoir est de concevoir et d'exécuter le test fonctionnel d'une application de base donnée avec interface graphique, puis d'appliquer ces tests à différents niveaux d'intégration. L'application à tester est un système d'information pour la gestion des ventes, de l'inventaire et des clients d'un supermarché (voir le document attaché pour une description). Vous allez développer des cas de test basés sur des cas d'utilisation et implémenter les stubs/mocks nécessaires pour tester les composantes indépendamment.

Une partie importante de ce travail consiste à comprendre un système logiciel qui n'a pas été conçu et implémenté par vous. Toutes les documentations disponibles vous ont été données. Il est de votre responsabilité d'enrichir votre compréhension du programme en navigant son code source et en l'exécutant.

Partie 1 Conception des cas de test

Question 1.1. (10%)
Fournissez un graphe de scénario correspondant à chacun des cas d'utilisation du système.

Question 1.2. (10%)
Fournissez un ensemble de scénarios tel que toutes les branches sont couvertes dans les graphes de scenarios. Énumérez les scénarios dans une table au format suivant :

Id du scénario	Évènements	Description

Question 1.3. (10%)
Fournissez la specification d'un ensemble de cas de test correspondant aux scénarios. La spécification de chaque cas de test devrait inclure :

- Un identificateur de cas de test
- Une référence à l'Id du scénario
- Les étapes de préparation du test (setup)
- La séquence des évènements de l'utilisateur et réponses attendues du système
- Le résultat final attendu

Partie 2 Testing indépendant de la couche GUI (30%)

On vous demande de tester la couche GUI (la classe **CounterFrame**) indépendamment du reste de l'application. Vous allez devoir développer des implémentations stubs (ou mocks) de **MarketProxy** ainsi qu'un nouveau programm de test principal similaire à **TestSuperMarket** qui utilise ce stub. Exécutez vos cas de test de la question 1.3 en utilisant ces stubs et rapportez vos résultats ainsi que l'information de couverture. Vous pouvez utiliser EclEmma ou un autre outil de couverture de code (veuillez inclure des captures d'écran dans votre rapport).

Partie 3 Intégration des couches GUI et Application (30%)

Dans cette partie, vous allez intégrer les couches GUI (CounterUI) et Application (packages **Counter** et **Market**)**.** Vos tests utiliseront les vraies classes de la couche Application (**SocketProxy**, **BasicServer** et **MarketServer**) mais vous devrez développer des implémentations stubs (ou mocks) de **DBManager**; ainsi qu'un nouveau programme de test principal similaire à **TestSuperMarket** qui utilise ce stub. Exécutez vos cas de test de la question 1.3 en utilisant ces stubs et rapportez vos résultats ainsi que l'information de couverture. Vous pouvez utiliser EclEmma ou un autre outil de couverture de code (veuillez inclure des captures d'écran dans votre rapport).

Partie 4 Intégration complète (10%)

Dans cette partie, vous allez intégrer l'application entière, incluant la base de données. Vous allez devoir insérer des données de test dans la base de données. Exécutez vos cas de test de la question 1.3 en utilisant ces stubs et rapportez vos résultats ainsi que l'information de couverture. Vous pouvez utiliser EclEmma ou un autre outil de couverture de code (veuillez inclure des captures d'écran dans votre rapport).