

**Institution :** Institut Grasset  
**Professeur :** Christian Labelle  
**Cours :** Contrôle de la qualité I (420-PA4-AG)  
**Session :** Automne 2018

---

### Descripteur Travail pratique 3

**Objectif :** Tester les améliorations fonctionnelles d'une application

**Pondération :** 25% de la note finale

---

### Description :

Effectuer les tests fonctionnels de l'application web « [JPetStore](#) » à l'aide de classes JUnit 4 et de Selenium, et faire l'intégration avec TestLink.

#### Étapes à réaliser:

1 <sup>e</sup> partie	
1.1	Créez un projet dans TestLink.
1.2	Créez un plan de test dans le projet TestLink créé à l'étape précédente.
1.3	Créez un « Build/Release » dans TestLink pour effectuer les tests fonctionnels de la version <u>originale</u> de l'application « JPetStore ».
2 <sup>e</sup> partie	
2.1	Créez les suites de tests dans TestLink regroupant les cas de tests couvrant l'ensemble des fonctionnalités de l'application « JPetStore ».
2.2	Créez les cas de tests dans TestLink pour tester chaque fonctionnalités de la version <u>originale</u> de l'application « JPetStore ».
2.3	Assignez les cas de tests créés dans TestLink au plan de test.
2.4	Créez les spécifications de l'application « JPetStore » dans TestLink à partir du document fourni avec le projet Eclipse.
2.5	Assignez les spécifications créées à l'étape précédente au cas de tests dans TestLink.
2.6	Générez le code Selenium permettant de réaliser les cas de tests spécifiés dans TestLink pour la version <u>originale</u> de l'application « jPetStore ».
2.7	Créez un « package » Java dans le projet Eclipse fourni et y ajouter toutes les classes JUnit contenant le code Selenium généré pour la version <u>originale</u> de l'application « jPetStore ». Assurez-vous que le code de chaque classe JUnit met à jour le statut de l'exécution des tests dans TestLink pour le « Build / Release » approprié.
2.8	Exécutez les tests sur la version <u>originale</u> de l'application « jPetStore ».
2.9	Générez les rapports suivants une fois l'exécution des tests complétées: <ul style="list-style-type: none"><li>• « Requirement Specification Document » en format « Pseudo MS Word »</li><li>• « Test Plan Report » en format « PDF »</li></ul> <b>Important : Ces rapports doivent être remis pour évaluation.</b>

3 <sup>e</sup> partie	
3.1	Modifiez la version originale de l'application « jPetStore » en intégrant les fichiers fournis.
3.2	Exécutez les tests (sans modification) sur la version <u>modifiée</u> de l'application « jPetStore ».
3.3	<p>Générez les rapports suivants une fois l'exécution des tests complétées:</p> <ul style="list-style-type: none"> <li>• « Requirement Specification Document » en format « Pseudo MS Word »</li> <li>• « Test Plan Report » en format « PDF »</li> </ul> <p><b>Important : Ces rapports doivent être remis pour évaluation.</b></p>
4 <sup>e</sup> partie	
4.1	Créez un nouveau « Build/Release » dans TestLink pour effectuer les tests fonctionnels de la version <u>modifiée</u> de l'application « JPetStore ».
4.2	Modifiez les cas de tests existants ou ajoutez les cas de tests manquants dans TestLink pour tester chaque fonctionnalités de la version <u>modifiée</u> de l'application « JPetStore ».
4.3	Assignez les cas de tests ajoutés dans TestLink au plan de test si nécessaire.
4.5	Générez le code Selenium permettant de réaliser les cas de tests spécifiés dans TestLink pour la version <u>modifiée</u> de l'application « jPetStore ».
4.6	Créez un « package » Java dans le projet Eclipse fourni et y ajouter toutes les classes junit contenant le code Selenium généré pour la version <u>modifiée</u> de l'application « jPetStore ». Assurez-vous que le code de chaque classe junit met à jour le statut de l'exécution des tests dans TestLink pour le « Build / Release » approprié.
4.7	Exécutez les tests sur la version <u>originale</u> de l'application « jPetStore ».
4.8	<p>Générez les rapports suivants une fois l'exécution des tests complétées:</p> <ul style="list-style-type: none"> <li>• « Requirement Specification Document » en format « Pseudo MS Word »</li> <li>• « Test Plan Report » en format « PDF »</li> </ul> <p><b>Important : Ces rapports doivent être remis pour évaluation.</b></p>

### Détails de remise :

Fournir un fichier compressé (zip) contenant les éléments ci-dessous. **Le fichier doit être déposé dans un dossier sur votre Google Drive et le dossier partagé avec l'utilisateur 4774687@gmail.com**

- Le fichier OVA de la machine virtuelle sur laquelle vous avez créé le projet TestLink et exécuté vos tests.
- Tous les rapports « Requirement Specification Document » en format « Pseudo MS Word » générés à partir de TestLink.
- Tous les rapports « Test Plan Report » en format « PDF » générés à partir de TestLink.
- Le projet Eclipse (Photon) contenant le code source des tests JUnit automatisant l'ensemble des jeux d'essai à l'aide de Selenium et l'intégration à TestLink (pour la version originale de l'application « jPetStore » et la version modifiée).

---

### Critères d'évaluation :

- |   |          |
|---|----------|
| 1. Les cas d'utilisation sont documentés selon les normes dans TestLink   | 3 points |
| 2. Les activités clés sont planifiées de façon réaliste pour la version originale de l'application et documentée selon les normes dans TestLink                                   | 2 points |
| 3. Les activités clés sont planifiées de façon réaliste pour la version modifiée de l'application et documentée selon les normes dans TestLink                                    | 2 points |
| 4. Les cas de tests sont liés aux cas d'utilisation dans TestLink   | 3 points |
| 5. Les jeux d'essai créés sont appropriés aux tests à effectuer et documentés selon les normes dans TestLink  | 3 points |
| 6. Le code source est formaté selon les normes spécifiées, respecte l'ensemble des normes dictées par « checkstyle » et utilise efficacement les possibilités de l'outil Selenium | 3 points |
| 7. Les tests de la version originale de l'application sont fonctionnels, complets et pertinents   | 3 points |
| 8. Les tests de la version modifiée de l'application sont fonctionnels, complets et pertinents  | 3 points |
| 9. Le statut des cas de tests définis dans TestLink est mis à jour automatiquement par les tests jUnit  | 3 points |

**Total :** 25 points (qui seront ramenés sur 25 %)

---

**Remise :** Cours #15

**Politique de retard :**

À partir de la PIEA, pour l'enseignement aux adultes, la politique de retard dans les remises de travaux a été précisée de la manière suivante :

- L'étudiant qui remet un travail en retard est pénalisé
- Le professeur enlèvera 10% du maximum des points par jour de retard, incluant les jours de fin de semaine.
- Après la remise des travaux corrigés, les retards ne sont plus admissibles. L'étudiant obtient la note zéro.

**Politique de plagiat**

Sera considéré comme du plagiat toute utilisation ou reproduction d'idées ou de matériel existants en format papier, analogique ou numérique, incluant tous les contenus sonores ou vidéos disponibles sur internet sous forme de montage ou de produit fini. L'utilisation de productions existantes sera permise seulement pour des fins d'analyse. L'utilisation de sources communes d'enregistrement sonore par deux ou plusieurs élèves entraînera aussi une sentence pour plagiat. Ce dernier implique la note zéro pour tous les protagonistes impliqués. L'utilisation de banques de sons fournies en classe n'est pas considéré comme du plagiat. L'utilisation de sons provenant d'internet, sous la forme non-édités et destinés à un montage par l'étudiant est permise, sauf si ces sons font partie de la même oeuvre que celle proposée pour le travail

**Politique linguistique**

Le travail doit être produit en langue française (Rédaction, tournage, montage, habillage infographique, etc).

**Politique de contenu**

Sont à proscrire tous les contenus à caractère irrespectueux, diffamatoire ou explicite, qu'ils soient sexuels, racistes, religieux ou autres. Tout contenu devra être validé par le professeur avant d'être diffusé.