

Tests du programme

1 Introduction

Nous allons consacrer dans ce dossier l'étude des fonctionnalités de notre application *Royal* dans la recherche d'éventuels bogues. Il s'agit d'une application de gestion de *BDs* personnelle écrite en Java. Nous nous basons sur une application déjà existante¹, nous devons donc veiller à vérifier que les fonctionnalités qui existaient déjà restent fonctionnelles, qu'on ne les a pas « cassées », en plus de la vérification des fonctionnalités réellement ajoutées.

L'application consiste globalement en une fenêtre principale contenant une liste de toutes les *BDs* déjà passées entre les mains de l'utilisateur. Quand un ouvrage est sélectionné, nous pouvons voir les informations détaillées le concernant à droite. Il est aussi possible de modifier ces informations, et, bien sûr de rajouter des nouveaux ouvrages. L'ajout peu s'effectuer soit manuellement, c'est à dire que l'utilisateur rentre lui même les *ISBNs* des livres dans l'application, soit automatiquement via un programme Android qui permet d'ajouter simplement les livres en prenant en photo leur code-barres. Cette application est aussi effectuée par nos soins, et demandera des tests complémentaires. L'application est aussi en mesure de pouvoir aller rechercher sur internet les informations relatives à un livre, et l'utilisateur pourra définir si le livre a été acheté ou emprunté dans une bibliothèque. Si le livre a été emprunté, une alarme de retour peu être configuré.

2 Stratégie de test

Pour notre projet, nous effectuerons différents types de test su la base de la typologie présentés en cours (*Fonctionnels, IHM/Navigation, Opérabilité, Performances, Charges, Sécurité*).

2.1 Fonctionnels

Les tests Fonctionnels seront tres nombreux pour notre projet. En effet veiller à l'existence ainsi que le bon fonctionnement des fonctions décrite dans le cahier des charges est une priorité dans notre projet. De plus nous aurons à assurer le bon fonctionnement de notre application bien hors de son utilisation type. Cependant ayant une utilisation mono-utilisateur et utilisant une base de donnée virtuelle local nous ne traiterons pas de test sur les accès concurrent.

2.2 IHM / Navigation

Notre projet étant utilisé par des personnes de tout type pour une utilisation personnel il faudra veiller à l'ergonomie de celui-ci ainsi qu'à son homogénéité. Cette homogénéité concernera aussi les différent message d'erreurs présent dans notre projet. Afin de vérifier ces points il faudra donc effectuer de nombreux test sur l'IHM ainsi que la Navigation.

2.3 Opérabilité

Notre projet comportant une application pc ainsi qu'une application android, il faudra assurer l'interopérabilité des deux applications. De plus l'application pc étant développé en Java il faudra vérifier la compatibilité avec les divers système d'exploitation. L'application créé peu de fichier temporaire, de plus si l'application s'arrête inopinément au prochain lancement ces fichier seront écrasés. C'est pourquoi nous effectuerons des test sur l'opérabilité de notre projet.

1. La version de base de Royal : www.royal-projet.org

2.4 Performances / Charges

Notre projet étant mono-utilisateur et n'utilisant pas de serveur, les tests de charge ne seront pas effectué. Tout-fois des tests de performance seront effectué pour les fonctions d'importation d'ISBN scanné.

2.5 Sécurité

A nouveau, notre projet ne contenant d'habilitation et n'enregistrent pas d'information (*notre projet n'utilise pas de serveur et la base de données est virtuelle*) il n'y aura pas de tests de sécurité. De plus seul la coupure de courant pourrait être tester pour le mode dégrader c'est pourquoi nous n'effectuerons pas de test de sécurité. Pour la sauvegarde, la base de données est virtuel et recréé à chaque lancement (*hibernate*) c'est pourquoi il n'y a pas besoin de sauvegarde.

3 Matrice fonctionnelle

Fréquence	Importance	Capitale	Forte	Faible
Très souvent		1A – 1PC – 5PC – 6PC	3A – 4PC	4A
Moyennement souvent		2PC – 7PC		
Rarement		2A – 8PC	5A – 3PC	

Les tests obligatoires sont en blanc, recommandés en gris clair, et optionnels en gris foncés.

Applications PC :

- 1PC — Ajout des informations de base.
- 2PC — Liens entre les albums.
- 3PC — Ajout d'une image de couverture.
- 4PC — Affichage selon critères.
- 5PC — Recherche des informations grâce au code barres.
- 6PC — Importation des *ISBNs* via courriel.
- 7PC — Ajout automatique des albums.
- 8PC — Configuration d'une adresse email.

Application Android :

- 1A — Capture du code barres.
- 2A — Configuration d'une adresse email.
- 3A — Validité du code barres.
- 4A — Enregistrement des albums.
- 5A — Synchronisation avec l'application PC.