

**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE



LOG3430

MÉTHODES DE TEST & DE VALIDATION DU LOGICIEL

Laboratoire 1

Tests Unitaires

Sanghyuk Lee 1589242

François-Xavier Legault 1876882

17 Septembre 2019

Introduction

L'objectif de ce travail pratique est d'implémenter des tests unitaires afin de s'assurer de la conformité des spécifications. Il est essentiel de faire des tests unitaires pour valider que chaque unité d'un logiciel fonctionne comme prévu. Il existe plusieurs infrastructures logicielles (*Framework*); JUnit pour Java, unittest pour Python, RSpec pour Ruby, etc. Au cours de ce travail pratique, unittest est utilisé pour faire des tests unitaires. Nous avons dans un premier lieu complété les classes "TestContactDAO" et "TestContactService", puis nous nous sommes assuré de la conformité des spécifications du programme et de la validité des tests unitaires.

Structure

Le tableau suivant démontre chaque classe et la description brièvement :

NOM	DESCRIPTION
Contact	Classe d'entité avec les attributs : id, first_name, last_name, phone, mail, updated, updated_date
ContactDAO	Classe de DAO (Data Access Object) pour se connecter à la base de données, créer la table Contact, exécuter des opérations élémentaires
ContactService	Classe de service qui rassemble les fonctions métiers qui peuvent être utilisés .

Conformité des spécifications (réponse à la question 4)

Selon les spécifications fonctionnelles de l'application, nous croyons que "ContactService" est conforme à celles-ci. La classe service "ContactService" rassemble les fonctions métiers, pouvant être utilisées directement par une interface par exemple. Elle est dépendante de la classe "ContactDAO" pour sauvegarder les traitements dans la base de données. On peut ainsi ajouter, mettre à jour et supprimer des contacts dans cette base de données. Les contacts se caractérisent par leur identifiant personnel, nom, prénom, numéro de téléphone et leur Email.

Selon les spécifications fonctionnelles, le couple d'attributs: nom et prénom, est considéré comme identifiant unique d'un contact (deux contacts ayant les mêmes noms et prénoms impossible). Un attribut booléen sur chacun des contacts indique si le contact est à jour, et un autre attribut contient la date de la dernière mise à jour du contact. L'application peut donc se servir de ces informations pour effectuer plusieurs traitements sur les contacts. De plus, "TestContactService" permet de tester correctement la conformité aux spécifications énoncées. Effectivement, cette classe contient tous les tests nécessaires pour les méthodes de la classe "ContactService" et a une couverture adéquate.

Conclusion

Nous avons donc mis en pratique nos connaissances théoriques sur les tests unitaires afin de valider la fonctionnalité de chaque unité du programme. Nous nous sommes familiarisé avec les tests unitaires qui impliquent un accès à une base de donnée. Finalement, en considérant les spécifications fonctionnelles de l'application, nous nous sommes assuré que le programme est conforme aux spécifications et que les tests unitaires permettent de tester correctement la conformité de ces spécifications.