

# Rapport de tests

Bancomat Web  
Java EE / QDL

## Introduction

Dans le cadre du cours de la HE-ARC « Java Entreprise Édition » il nous est demandé, à titre éducatif, de développer une application web à l'aide du Framework JSF (Java Server Faces). Ce projet doit être suivi à l'aide de procédures de test pour assurer la qualité du logiciel. Ce document relate la stratégie, planification, spécification ainsi que le bilan des tests.

Le site web est un système de bancomat disposant d'une interface, destinée à des clients, de retrait / dépôt / transfert d'argent à partir de compte bancaire qu'ils peuvent demander à créer. Le programme disposera également d'une interface spéciale d'administration des comptes et des utilisateurs.

## Stratégie de test

La partie suivante aborde les différents aspects de la stratégie des tests.

## Qualité du produit

Le produit doit fournir 2 rôles bien distincts qui n'ont pas les même droit et fonctionnalités :

Le client doit pouvoir :

- Demander l'ouverture d'un compte
- Retirer
- Déposer
- Transférer de l'argent

L'administrateur doit pouvoir :

- Accepter les demandes d'ouverture de compte
- Gérer les comptes clients
- Gérer les personnes inscrites

## Objectifs de test

Les choses importantes à tester sont :

- Les accès utilisateurs. Il ne faut pas qu'on puisse accéder au compte d'un autre cela entraînerait un gros problème de sécurité. Entre autres, il serait idéal de tester toute la sécurité et d'effectuer des transferts cryptés en HTTPS au vu de l'importance des données.
- La procédure d'ouverture de compte car cela entraînera le blocage de l'utilisation de l'application.
- Le bon fonctionnement des retraits, dépôts, transferts clients. Autrement cela rendrait également l'utilisation obsolète du site.
- Tester les différents types de comptes bancaires.

Éventuellement la gérance des clients et comptes clients en cas de bug lors de la création. Les autres fonctionnalités sont prioritaires car cela n'empêche pas le déploiement de l'application.

## Périmètre de test

La partie test du projet sera défini uniquement par des tests systèmes (performances et sécurité) ainsi que par des tests de validation défini par les fonctionnalités. Les programmes d'automatisation de tests tel que Selenium permettant un travail réduit et d'éviter de tester les fonctionnalités au niveau unitaire et intégration.

## Plan de test

### Niveau de test

#### Tests système

- **La sécurité** : Comme expliqué précédemment, l'application dispose de connexion ayant des droits attribués. Il s'agit ici de définir la procédure de test pour garantir des accès autorisés. Pour cela nous allons créer 3 profils et premièrement tester un accès public à une grande de l'application avec 1 utilisateur non-inscrit. Les 2 derniers profils sont un client loggué et un administrateur loggué qui tenteront d'accéder à 2 pages, l'une faisant partie de la partie cliente du site et l'autre de la partie administration.
- **Les performances** : Étant donné que le site web ne sera pas déployé et que seule la version locale est implémentée, nous ne ferons pas réellement de test de performances. Cependant des test utilisateurs lors du développement évalueront globalement si le temps de réponse est acceptable.

## Tests de validation

La liste des fonctionnalités primordiales à tester sont les suivantes:

- **Login / logout** : Connecter l'utilisateur quel que soit son rôle du moment qu'il rentre des coordonnées correspondantes à un compte adéquate stocké sur le serveur. La déconnexion doit faire revenir l'utilisateur à un statut non-connecté.
- **Création de compte / Refus de compte** : Faire une demande de création de compte de chaque type (client). Voir les demandes de création et valider, ou refuser, la création d'un compte client (administrateur).
- **Gestion de compte (client)** : Voir ses différents comptes et leurs soldes respectifs, déposer, retirer de l'argent et effectuer des transferts vers des comptes du même site.
- **Gestion de compte (admin.)** : Voir les comptes d'un client défini, supprimer un compte.
- **Gestion des clients (admin.)** : Voir la liste des clients, supprimer un client.

## Organisation

Le travail est réparti dans l'équipe de test qui est composé de 3 testeurs :

- **Omband Ndo Charles** : Tests de performances
- **Lovis Thomas** : Rédaction de la documentation, tests système
- **Vulliemin Kevin** : Rédaction de la documentation, tests de validation

## Infrastructure

### Environnement de test

Tous les tests sont effectués en local sur l'ordinateur du testeur associé. La configuration du serveur est tel que suivi :

- **Omband Ndo Charles** : Wamp Server 3.0.6, Windows 10
- **Lovis Thomas** : Wamp Server 3.0.6, Windows 8.1
- **Vulliemin Kevin** : Wamp Server 3.0.6, Windows 10

## Outils de test

Les outils sont les programmes d'automatisations de test par scripts suivants :

- **Katalon Recorder (Selenium IDE) :**  
[https://www.seleniumhq.org/download/#selenium\\_ide](https://www.seleniumhq.org/download/#selenium_ide)  
Extension disponible pour la plupart des navigateurs récents, elle permet d'enregistrer une suite d'action sous forme de script afin de l'exécuter sur demande.
- **Apache JMeter :** <https://jmeter.apache.org/>  
Programme permettant la simulation massive de requêtes sur un site web. Un certain scénario peut être enregistrée afin de simplifier et automatiser les lancements.

# Spécifications détaillées des tests

## Clusters

Les clusters sont au nombres de 6 est sont défini comme suit :

<b>ID</b>	1	<b>ID</b>	2
<b>Cluster</b>	Login / Logout	<b>Cluster</b>	Création / Refus de compte
<b>Version</b>	0.1	<b>Version</b>	0.1
<b>Date</b>	25.04.2018	<b>Date</b>	25.04.2018
<b>Auteur</b>	Vulliemin Kevin	<b>Auteur</b>	Vulliemin Kevin

<b>ID</b>	3	<b>ID</b>	4
<b>Cluster</b>	Gestion de compte client	<b>Cluster</b>	Gestion de compte admin.
<b>Version</b>	0.1	<b>Version</b>	0.1
<b>Date</b>	25.04.2018	<b>Date</b>	25.04.2018
<b>Auteur</b>	Vulliemin Kevin	<b>Auteur</b>	Vulliemin Kevin

<b>ID</b>	5	<b>ID</b>	6
<b>Cluster</b>	Gestion des clients	<b>Cluster</b>	Vérification des accès
<b>Version</b>	0.1	<b>Version</b>	0.1

<b>Date</b>	25.04.2018	<b>Date</b>	25.04.2018
<b>Auteur</b>	Vulliemin Kevin	<b>Auteur</b>	Vulliemin Kevin

## Conditions de test

ID	Type	Description	Priorité
1.1	Fonctionnel	L'identification de l'utilisateur est réussi s'il rentre des informations de connexion associé à un utilisateur dans la base de données.	2
1.2	Fonctionnel	L'identification échoue si l'utilisateur rentre un nom de compte inexistant.	2
1.3	Fonctionnel	L'identification échoue si l'utilisateur rentre un mot de passe qui n'est pas associé au compte rentré.	2

ID	Type	Description	Priorité
2.1	Fonctionnel	Le choix de type de compte doit être au format texte et pouvoir être modifiée avant de faire une demande	2
2.2	Fonctionnel	Le client doit recevoir une alerte contenant un texte de confirmation de sa demande de création de compte	1
2.3	Fonctionnel	La liste de demandes des clients dans l'interface administrateur doit être au format texte	2
2.4	Fonctionnel	L'administrateur doit pouvoir accepter une demande client à l'aide d'un bouton explicite	2
2.5	Fonctionnel	L'administrateur doit pouvoir refuser une demande client à l'aide d'un bouton explicite	2

ID	Type	Description	Priorité
3.1	Fonctionnel	Le nom des comptes doit être au format texte	1
3.2	Fonctionnel	Le montant des comptes doit être au format nombre	1
3.3	Fonctionnel	Le bouton de dépôt doit permettre d'augmenter la somme sur le compte bancaire	1

3.4	Fonctionnel	Le bouton de retrait doit permettre de diminuer la somme sur le compte bancaire	1
3.5	Fonctionnel	Le bouton de transfert doit permettre de transférer de l'argent vers un autre compte	1
3.6	Fonctionnel	Les fonctions de retrait et de transfert ne peuvent être utilisées si cela engendre le passage du montant du compte en dessous de 0. Cela n'est pas valable pour les comptes de type crédit	1

ID	Type	Description	Priorité
4.1	Fonctionnel	Le nom des comptes clients doivent s'afficher au format texte	1
4.2	Fonctionnel	Le montant des comptes clients doivent s'afficher au format nombre	1
4.3	Fonctionnel	Le bouton supprimer compte doit entraîner la suppression du compte bancaire associé dans la base de données	1

ID	Type	Description	Priorité
5.1	Fonctionnel	Le nom des clients dans la liste doit être au format texte	1
5.2	Fonctionnel	Le bouton supprimer doit entraîner la suppression du compte client associé dans la base de données	1

ID	Type	Description	Priorité
6.1	Fonctionnel	Tous les liens et pages doivent être accessibles	2
6.2	Fonctionnel	Un utilisateur non-connecté (public) à le droit d'accéder uniquement à la page d'accueil et à la page d'inscription	2
6.3	Fonctionnel	Un utilisateur connecté en tant que client à le droit d'accéder à son espace de compte et aux actions associées (dépôt, retrait, transfert)	2
6.4	Fonctionnel	Un utilisateur connecté en tant qu'administrateur à le droit d'accéder à la liste des clients et de leurs compte ainsi que de supprimer ceux-ci.	2

# Rapport d'exécution des tests

Certains script ont pu être créé mais pas exécuté.

## Liste des bugs

Pas de gestion de sécurité côté serveur Glassfish.