

CAHIER DE RECETTE

Version :	1.3
Date :	18/05/17
Rédigé par :	L'équipe SmartLogger
Relu par :	L'équipe SmartLogger

HISTORIQUE DE LA DOCUMENTATION

Version	Date	Modifications réalisées
0.1	11/12/16	Création
0.2	05/01/17	Modifications de la partie 3. Stratégie de tests
1.0	12/01/17	Correctifs et restructuration du document
1.1	26/01/17	Corrections liées aux remarques professorales
1.2	28/03/17	Ajout des procédures de tests sur les itérations 1 et 2.
1.3	18/05/17	Validation des procédures de tests définies

1. Documents applicables et de référence

- Le document de Spécifications Techniques des Besoins : STB.pdf.
- Le document d'architecture du logiciel : DAL.pdf
- Le plan de développement : PdD.pdf
- Le Glossaire des termes techniques et des acronymes : Glossaire.pdf

2. Environnement de test

La plate-forme cible du projet est un serveur Linux, ainsi dans l'optique d'effectuer des tests sur un système le plus proche possible de l'original, nous allons travailler sur des Systèmes Linux uniquement, sachant que les systèmes Microsoft ont été écartés par le client.

Site de réalisation des tests :

- Réalisation des tests sur nos machines personnelles.
- Potentiellement un serveur présent directement sur le site client.
- Utilisation d'un serveur identique à celui du client afin d'avoir un environnement de travail identique pour le produit, en production ou durant son développement.

Configurations matérielles utilisées :

- Système Linux (Fedora, Ubuntu)
- Serveur basé sur un système Linux.

Outils de test mis en œuvre :

- JUnit pour les tests unitaires de la partie Java du projet.

Outils de gestion des anomalies :

- Utilisation d'un logiciel de Bug Tracker tels que MantisBT ou encore Redmine.

Jeux de données utilisés durant les phases de tests :

- Données de la forme : *niveau de criticité – informations importantes*
 - *niveau de criticité* : Valeur numérique permettant d'affilier un niveau de criticité à un format de données précis.
 - *Informations importantes* : Toutes les données traitées par le système sont considérées comme importantes, c'est pourquoi nous stockons l'ensemble des données sans distinction.

3. Stratégie de test des cas d'utilisation

3.1. Solution de test

Les solutions de test que nous avons choisis de définir consistent à rédiger les tests en parallèle du développement. Cela implique qu'une partie de l'équipe se chargera de la rédaction et de l'automatisation complète des tests, pendant que le reste de l'effectif sera chargé de développer le contenu de l'itération.

Cela permettra ensuite, via automatisation des tests de pouvoir rejouer à chaque fin d'itération, les tests unitaires de chacun des modules afin d'accélérer le processus de test, et d'empêcher tout phénomène de régression. Cet investissement servira à allouer plus de temps aux tests d'intégration.

En cas d'anomalie durant le développement, l'utilisation d'un outil de *bug tracking* permettra de détecter et de fournir des informations complémentaires sur des défauts, afin d'en faciliter la correction par l'équipe.

3.2. Description des scénarios de test correspondant aux cas d'utilisation

[UC-001] Fonctionnement principal				
Objectif	Vérifier le fonctionnement principal de l'application			
Éléments à tester	Réception des données externes Conversion des données dans le bon format Vérification de la validité des données Stockage des données dans la base de données Capacité d'analyse fonctionnelle Adaptation du comportement en fonction des résultats Envoi des d'alertes potentielles			
Pré-requis	Un serveur fonctionnel, avec une connexion internet permettant de recevoir des flux HTTP depuis d'autres services web.			
Initialisation	On lance le script de démarrage de l'application afin qu'elle s'initialise toute seule.			
Scénarios :				
ID	Démarche	Modules	Comportement attendu	Exigences Vérifiées
1	On lance le script de démarrage.	Tous	Le système se lance et le module analytique effectue correctement son travail.	OP-1 OP-4
Rapport de test		<input type="checkbox"/> Testé par :		Le :
Fonctionnalité :		Conformité :		Ergonomie :
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible

[UC-002] Détection de données erronées				
Objectif	Vérifier que le système détecte correctement les données erronées.			
Éléments à tester	Réception des données externes Conversion des données dans le bon format Vérification de la validité des données			
Pré-requis	Un serveur fonctionnel, avec une connexion internet permettant de recevoir des flux HTTP depuis d'autres services web La réception de données erronées depuis le flux HTTP			
Initialisation	On envoie des données invalides depuis le flux HTTP			
Scénarios :				
ID	Démarche	Modules	Comportement attendu	Exigences Vérifiées
1	On envoie des données erronées depuis la vue de l'application ou par le biais d'un flux HTTP	Module E/S	Le système reçoit une donnée depuis l'application elle-même ou depuis un flux HTTP	OP-1 IN-1
2	Le système traite et convertit le flux en données exploitables	Module C/V	Le système a converti les données dans le format du système	OP-5 IN-1
3	Le système effectue la vérification des données	Module C/V	Le système vérifie la validité des données et prévient l'utilisateur que les données sont invalides (si celles-ci sont invalides)	OP-5 IN-1
4	Le système stocke les données afin de les traiter ultérieurement	Module de Gestion des Données	Le système stocke les données dans une base de données dédiée au stockage des données non traitables ou non acceptables	OP-3
5	Le système envoie une alerte à l'utilisateur pour le prévenir d'une donnée erronée	Module E/S	Le système envoie une alerte via un canal choisi	OP-2 IN-2
Rapport de test		☐ Testé par :		Le :
Fonctionnalité :		Conformité :		Ergonomie :
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible

[UC-004] Entraînement du système				
Objectif	Vérifier que le système puisse être entraîné depuis l'application web par un entraîneur.			
Éléments à tester	Réception des données internes Conversion des données dans le bon format Vérification des données valides Analyse des données Apprentissage du module analytique Sauvegarde des données dans la base de données Affichage des résultats du traitement sur l'application web			
Pré-requis	Le système doit être opérationnel, l'opérateur doit avoir accès au service web de l'application afin d'envoyer les données à la machine d'apprentissage			
Initialisation	L'opérateur se connecte au serveur web via un navigateur web, et possède un jeu de données qu'il souhaite envoyer à la machine			
Scénarios :				
ID	Démarche	Modules	Comportement attendu	Exigences Vérifiées
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre	Module d'Interface Utilisateur	L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web	OP-1 IN-1 IN-3
2	L'entraîneur clique sur un lien permettant d'afficher la page d'envoi de jeux de données	Module d'Interface Utilisateur	L'application web affiche une page permettant à l'entraîneur d'envoyer des jeux de données	OP-5 IN-1 IN-3
3	L'opérateur prépare les jeux de données à envoyer et les envoie depuis l'application web	Module d'Interface, Utilisateur / Module E/S	L'opérateur envoie les données qu'il souhaite faire traiter par le système	OP-5 IN-1 IN-3
4	Le système traite et convertit le flux en données exploitables	Module de C/V	Le système a converti les données dans le format du système	OP-5
5	Le système effectue la vérification des données	Module de C/V	Le système vérifie la validité des données et prévient l'utilisateur que les données sont invalides (si celles-ci sont invalides)	OP-5
6	Le système effectue l'analyse des données afin de modifier son comportement ou d'apprendre de nouveaux flux d'entrée	Module Analytique	Le système exécute l'analyse des données que lui envoie l'opérateur de l'application	OP-5
7	Le système sauvegarde les modifications de son comportement dans la base de données correspondante	Module de Gestion des Données	Le système stocke la modification de son comportement dans la base de données correspondante	OP-3
Rapport de test		☐ Testé par :		Le :
Fonctionnalité :		Conformité :		Ergonomie :
☐ Excellente ☐ Bonne ☐ Moyenne ☐ Faible		☐ Excellente ☐ Moyenne ☐ Faible		☐ Excellente ☐ Moyenne ☐ Faible

[UC-005] Consultation des données				
Objectif	Vérifier qu'un utilisateur puisse consulter les données du module analytique			
Éléments à tester	Accès au système depuis un navigateur web Consulter l'état de la base de données			
Pré-requis	Le système doit être en fonctionnement Un navigateur web est requis, ainsi que la présence de la base de données			
Initialisation	On se connecte à l'application depuis un navigateur web présent dans le même réseau interne			
Scénarios :				
ID	Démarche	Modules	Comportement attendu	Exigences Vérifiées
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre	Module d'Interface Utilisateur	L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web	OP-1 IN-1 IN-3
2	L'entraîneur clique sur un lien affichant les données	Module d'Interface Utilisateur	L'entraîneur arrive sur la page de consultation des différentes tables	OP-5 IN-1 IN-3
3	L'entraîneur choisit les données qu'il souhaite consulter depuis une liste	Module d'Interface Utilisateur	L'entraîneur choisit les données et la page charge le contenu à afficher	OP-5 IN-1 IN-3
4	Le système interroge la table mentionnée ci-dessus	Module E/S, Module de Gestion des Données	Le système consulte la base de donnée souhaitée par l'entraîneur	OP-3
5	Le système renvoie le contenu de la table	Module d'E/S	Le système renvoie le résultat de la requête vers la page de consultation	OP-3
6	Le résultat est affiché sur la page de consultation	Module d'Interface Utilisateur	La page affiche le contenu de la table	OP-3 IN-3
Rapport de test		☐ Testé par :		Le :
Fonctionnalité :		Conformité :		Ergonomie :
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible

[UC-006] Ajustement du comportement				
Objectif	Vérifier que le système détecte correctement les données erronées			
Éléments à tester	Réception des données externes Conversion des données dans le bon format Vérification des données valides			
Pré-requis	Un serveur fonctionnel, avec une connexion internet permettant de recevoir des flux HTTP depuis d'autres services web La réception de données erronées depuis le flux HTTP			
Initialisation	On se connecte à l'application depuis un navigateur web présent dans le même réseau interne			
Scénarios :				
ID	Démarche	Modules	Comportement attendu	Exigences Vérifiées
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre	Module d'Interface Utilisateur	L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web	OP-1 IN-1 IN-3
2	Le testeur clique sur un lien permettant la modification du comportement de la machine	Module d'Interface Utilisateur	Le testeur arrive sur une page comportant l'ensemble des options de modification du comportement de la machine	OP-5 IN-1 IN-3
3	Le testeur envoie les modifications de comportement de la machine	Module d'Interface, Utilisateur Module E/S	La page de modification du comportement se met en attente de confirmation du comportement de la machine d'apprentissage	OP-5 IN-1 IN-3
4	Le module d'analyse répercute les modifications sur son comportement en lançant une nouvelle phase d'apprentissage	Module Analytique	Le système exécute une phase complète d'analyse et d'apprentissage afin de corriger son comportement.	OP-3
5	Le module d'analyse envoie une confirmation de la modification de son comportement	Module Analytique, Module E/S, Module d'Interface Utilisateur	La page de modification du comportement affiche le résultat de la modification	IN-1 IN-3
Rapport de test		☐ Testé par :		Le :
Fonctionnalité :		Conformité :		Ergonomie :
☐ Excellente ☐ Bonne ☐ Moyenne ☐ Faible		☐ Excellente ☐ Moyenne ☐ Faible		☐ Excellente ☐ Moyenne ☐ Faible

4. Stratégie de test appliquée

Cette partie va détailler les différentes applications des moyens de test, afin d'assurer la qualité des fonctionnalités produites depuis le début du développement.

Conformément aux solutions de tests sélectionnées précédemment, le pré-requis de toute validation d'une itération consistera à répéter les tests associés aux itérations précédentes. L'exécution de ces procédures de test, seront conçues par les différents membres de l'équipe, en respectant les principes généraux des tests unitaires et d'intégration. Leur réalisation et leur exécution sera alors assurée par le responsable qualité.

Le but de ces stratégies est d'offrir à l'équipe de test, un outil organisationnel simplifié, dans la mise en place de ces procédures. Ces mêmes stratégies décriront chacun un point technique crucial du projet, concernant un module du système ou concept général, en énumérant les différentes classes de test qui lui sont associés et les différents points de contrôle.

Module Analytique					
Objectif		Vérifier le fonctionnement et les capacités de traitement de l'analyseur.			
Éléments à tester		<ul style="list-style-type: none">Fonctionnalités de base de l'analyseRobustesse du modèle au format de donnéesPerformance (temps d'exécution et résultats)			
Description de la stratégie :					
ID	Mécanisme	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	Analyseur	SmartAnalyzerTest	Fonctionnalités de la classe SmartAnalyzer, qui constitue le noyau de l'analyseur. Robustesse du module	Cas de mauvaise instanciation	OK
				Changement de Modèle pendant l'analyse	OK
				Changement de modèle avant l'analyse	OK
				Prédiction depuis un objet non défini	OK
		SmartAnalyzerBenchmark	Tester, et comparer les performance de différents algorithmes de classification	Vérifier l'identité de deux prédictions successives.	OK
				NaiveBayes	OK
				DecisionTreeClassifier	OK
				LogisticRegression	OK
				RandomForestClassifier	OK
		Testé par : Grégoire Pommier			Le : 15/05/2017
Fonctionnalité :			Conformité :	Ergonomie :	100 %
<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>			<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	<div>[] Excellente</div> <div>[] Bonne</div> <div>[X] Moyenne</div> <div>[] Faible</div>	

Module E/S – Mécanismes d'alerte					
Objectif		Assurer la propagation d'une alerte, levée lors de l'analyse d'un log témoignant d'un état critique.			
Éléments à tester		<ul style="list-style-type: none">Les différentes API de sortie.Le mécanisme de levée des alertes			
Description de la stratégie :					
ID	Mécanisme	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	Levée d'alerte - Gestion des alerteurs	AlerterTest	Mise en place de l'alerteur, dans les cas d'utilisation potentiels. Implantation correcte du pattern Observer.	Ajout d'un alerteur	OK
				Ajout simultané de plusieurs alerteurs	OK
				Ajouts d'alerteurs identiques	OK
				Retrait d'un alerteur	OK
				Retrait non autorisé	OK
				Retrait du premier alerteur associé	OK
				Retrait du dernier alerteur associé	OK
				Utilisation avec un alerteur	OK
				Utilisation avec plusieurs alerteurs	OK
				Utilisation cas réel (avec alerte mail et Slack)	OK
				Utilisation sans alerteurs	OK
2	Alerteur Mail	MailNotifierTest	Vérification que les mail partent bien, que l'erreur est bien gérée. Vérification manuelle de la réception.	Envoi à plusieurs destinataires	OK
				Envoi à un destinataire non identifié	OK
				Envoi à un destinataire sans adresse	OK
3	Alerteur Slack	SlackNotifierTest	Vérification du bon fonctionnement des envois sur Slack	Envoi avec clé d'api de test	OK
				Envoi avec clé d'api standard	OK
				Envoi avec clé d'api erronée	OK
Testé par : Grégoire Pommier				Le : 15/05/2017	Couverture
Fonctionnalité :			Conformité :	Ergonomie :	100 %
<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>			<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	

Module E/S – Mécanisme de réception des données					
Objectif		Assurer la réception des logs et leur stockage dans le batch.			
Éléments à tester		<ul style="list-style-type: none">• L'entrée fonctionne correctement, elle est résistante à la montée en charge, au stress, à la durée.• Les logs sont correctement interprétés (parsing)• Les logs sont bien stockés			
Description de la stratégie :					
ID	Mécanisme	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	Gestion des requêtes d'entrée	InputManagerTest	Le gestionnaire d'entrée gère correctement les requêtes, surtout en cas de stress	Fonctionnement normal	OK
				Stress Test	OK
				Test de requête erronée	OK
				Test de fermeture de l'entrée pendant l'envoi	OK
2	Stockage des données reçues	LogBatchTest	Le batch ne perd pas les données qu'on lui fournit	Test du Batch vide	OK
				Test d'ajout d'un log	OK
				Test d'ajout de 2 logs	OK
3	Interprétation des logs reçus	LogParserTest	Le parser fonctionne correctement	Test du cas nul	OK
				Test du cas standard	OK
				Test d'un cas limite	OK
				Test mauvais entête	OK
				Test mauvais format	OK
Testé par : Grégoire Pommier				Le : 15/05/2017	Couverture
Fonctionnalité :			Conformité :	Ergonomie :	100 %
<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>			<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	<div>[] Excellente</div> <div>[X] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	

Module Utilitaire					
Objectif		Gérer les propriétés définies par les fichiers de configuration			
Éléments à tester		<ul style="list-style-type: none">La manipulation des fichiers (lecture/écriture) et de leurs propriétés associéesL'encryptage des propriétés dans les fichiers.			
Description de la stratégie :					
ID	Mécanisme	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	Gestionnaire de propriétés	PropertyManagerTest	Vérifier que les propriétés sont manipulés correctement, et que les fichiers associés correspondent aux propriétés du gestionnaire	Chargement de fichier inexistant	OK
				Lecture d'un fichier existant	OK
				Sauvegarde des propriétés du gestionnaire	OK
				Lecture d'un fichier vide	OK
2	Cryptage des propriétés	EncrypterPropertyManagerTest	Vérifier que l'encryptage se fait correctement.	Sauvegarde de donnée dans un fichier	OK
				Chargement de donnée depuis un fichier	OK
				Chargement de donnée depuis un fichier vide	OK
Testé par : Grégoire Pommier				Le : 15/05/2017	Couverture
Fonctionnalité :			Conformité :	Ergonomie :	100 %
<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>			<div>[X] Excellente</div> <div>[] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	<div>[] Excellente</div> <div>[X] Bonne</div> <div>[] Moyenne</div> <div>[] Faible</div>	

Module de Gestion des Données					
Objectif		Assurer la persistance des données reçues par SmartLogger			
Éléments à tester		La bonne manipulation de la base de données, ainsi que son contenu			
Description de la stratégie :					
ID	Mécanisme	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	Représentation d'unLog	LogTest	Le bon fonctionnement de l'objet Log	Validité du modèle par rapport aux informations à stocker	OK
2	DAO	DAOTest	Test fonctionnels CRUD sur la base de donnée. Test de réactions en cas d'erreur.	Tentative d'accès sans base de données associée	OK
				Procédure d'initialisation de la table de données	OK
				Procédure d'insertion	OK
				Accès avec des identifiants d'accès	OK
				Accès avec identifiant inexistant ou incorrects	OK
				Procédure de mise à jour d'une valeur	OK
3	Connecteur	DBConnectorTest	Tester la réalisation de la connexion à la base de données distante	Réalisation de la connexion	OK
				Fermeture d'un connecteur ouvert	OK
				Fermeture d'un connecteur déjà fermé	OK
				Ouverture d'un connecteur déjà ouvert	OK
Testé par : Grégoire Pommier				Le : 15/05/2017	Couverture
Fonctionnalité :			Conformité :	Ergonomie :	100 %
[X] Excellente [] Bonne [] Moyenne [] Faible			[X] Excellente [] Bonne [] Moyenne [] Faible	[] Excellente [X] Bonne [] Moyenne [] Faible	

Architecture Globale - Stratégies d'intégration					
Objectif			Représenter l'application globale		
Éléments à tester			Tester l'intégration entre les différents modules.		
Description de la stratégie :					
ID	Classes	Classe de test	Ce qui est vérifié	Cas de test	Validé ?
1	SmartLogger (Version itération 1)	SmartLoggerIntegrationML_I	Test d'intégration des modules 1 et 2	Test Standard	KO
2	SmartLogger (Version itération 2)	SmartLoggerIntegrationML_I_O	Test d'intégration des module 1 2 et 3	Test Standard	KO
3	SmartLogger (Version itération 3)	SmartLogger	Test de déploiement sur la plateforme	Test Standard	OK
Testé par : Grégoire Pommier				Le : 15/05/2017	Couverture
Fonctionnalité :			Conformité :	Ergonomie :	33%
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input checked="" type="checkbox"/> Moyenne <input type="checkbox"/> Faible			<input checked="" type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input checked="" type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	