

CAHIER DE RECETTE

Version :	0.1
Date :	14/12/16
Rédigé par :	L'équipe SmartLogger
Relu par :	L'équipeSmartLogger

Objectif : Ce document est consacré à la définition des moyens et des procédés mis en œuvre pour assurer la recette du produit logiciel développé.

L'objectif de la recette ou validation est de vérifier que le logiciel est conforme à la spécification référencée au chapitre 2 et qu'il répond aux attentes exprimées dans le cahier des charges initial.

Le cahier de recette recense les objectifs de tests de validation et les moyens nécessaires pour les atteindre. Il précise

- les conditions à satisfaire préalablement à l'exécution des tests,*
- les moyens matériels requis (plate-forme de tests),*
- la logique de leur déroulement (étapes successives),*
- les conditions d'arrêt.*

Le document de validation fait partie des documents contractuels du projet, au même titre que les spécifications techniques.

MISES A JOUR

Version	Date	Modifications réalisées
0.1	11/12/16	Création

1. Introduction :

Présenter les informations générales de la recette:

- *Fonctionnalités du logiciel (liste des cas d'utilisation)*
- *Liste des objets à tester (versions, lots, modules, etc.)*
- *Contexte d'exécution des tests*
- *Choix technologiques et dispositions particulières.*

2. Documents applicables et de référence

- *Le document de Spécifications Techniques des Besoins : STB.pdf.*
- *Le Glossaire des termes techniques et des acronymes : Glossaire.pdf*

3. Terminologie et sigles utilisés

- *Confer Glossaire.pdf*

4. 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 test :

- *Réalisation des tests sur nos machines personnelles.*
- *Potentiellement un serveur présent directement sur le site client.*

Configurations matérielles utilisées :

- *Système Linux (Fedora, Ubuntu)*

Outils de test mis en œuvre :

- *Junit pour les tests unitaires de la partie Scala et Java du projet.*

Jeu de données et/ou bases de données de test :

-

Contraintes à prendre en compte :

-

Présenter une description la plus précise possible des plates-formes utilisées pour les tests en indiquant leur degré de représentativité par rapport à la plate forme cible :

- *site où seront réalisés les tests*
- *configurations matérielles utilisées*
- *outils de test mis en œuvre (éventuellement)*
- *jeu de données et/ou bases de données de test*
- *contraintes à prendre en compte (disponibilité, accessibilité, limites d'utilisation, etc.)*

5. Responsabilités

Définir l'organisation des tests en terme de responsabilités et d'attributions (qui conçoit les tests ? qui les exécute ? qui fournit les données de test ? Etc.)

6. Stratégie de tests

Décrire la démarche générale mise en œuvre pour réaliser les tests

- *description de l'approche et des phases de tests*
- *campagne de test*
- *ordre d'exécution des tests*
- *critères d'arrêt des tests*

Test Fonctionnel : [UC-001] Fonctionnement principal				
Objectif :		Vérifier le fonctionnement principal de l'application		
Éléments à tester :		Réception des données externe Conversion des données dans le bon format Vérification de la validité des données Stockage des données dans la base de données Analyse du module analytique fonctionnelle Adaptation du comportement en fonction des résultats Envoie potentiel d'alerte		
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'installe tout seule.		
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On lance le script de démarrage.		Le système se lance et le module analytique effectue correctement son travail.	
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
Commentaire :			Approbation :	
Fiches d'anomalies émises :				

Test Fonctionnel : [UC-002] Détection de données erronées				
Objectif :		Vérifier que le système détecte correctement les données erronées.		
Éléments à tester :		Réception des données externe 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		
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On envoie des données erronées depuis la vue de l'application ou par le biais d'un flux HTTP.		Le système reçoit une données depuis l'application elle-même ou depuis un flux HTTP.	
2	Le système traite et converti le flux en données exploitable.		Le système à converti les données dans le format du système.	
3	Le système effectue la vérification des données		Le système vérifie la validité des données et prévient l'utilisateur que les données sont invalides (si celle-ci sont invalides).	
4	Le système stocke les données afin de les traiter ultérieurement.		Le système stocke les données dans une base de données dédiée aux stockages des données non traitable ou non acceptable.	
5	Le système envoie une alerte à l'utilisateur pour le prévenir d'une données erronées.		Le système envoie une alerte via un canal choisi.	
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
Commentaire :			Approbation :	
Fiches d'anomalies émises :				

Test Fonctionnel : [UC-003] Réutilisation des données				
Objectif :	Vérifier que le système puisse accepter des données utilisateur.			
Éléments à tester :	Réception des données interne Conversion des données dans le bon format (si nécessaire) Vérification de la validité des données Analyse des données Apprentissage du module analytique Sauvegarde des données dans la base de données			
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			
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre.		L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web.	
2	L'opérateur clique sur un lien permettant d'afficher la page des données qu'il souhaite refaire traiter par le système.		L'application web affiche une page contenant les données que le système peut traiter afin que celui-ci les re traite.	
3	L'opérateur choisie les données qu'il souhaite refaire travailler au système.		L'opérateur choisie les données qu'il souhaite faire traiter à nouveaux par le système.	
4	Le système reçoit les données a retravailler.		Le système reçoit les données a retravailler pour modifier son comportement ou ajouter de nouveaux flux d'entrée.	
5	Le système effectue l'analyse des données afin de modifier son comportement ou d'apprendre de nouveaux flux d'entrée.		Le système exécute l'analyse des données que lui envoie l'opérateur de l'application.	
6	Le système apprend des données reçus.		Le système apprend des nouvelles données reçues et modifie son comportement en conséquence.	
7	Le système sauvegarde les modification de son comportement dans la base de données correspondante.		Le système stocke le modification de son comportement dans la base de données correspondante.	

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	
Commentaire :		Approbation :	
Fiches d'anomalies émises :			

Test Fonctionnel : [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 interne 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		
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre.		L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web.	
2	L'entraîneur clique sur un lien permettant d'afficher la page d'envoi de jeux de données.		L'application web affiche une page permettant à l'entraîneur d'envoyer des jeux de données.	
3	L'opérateur prépare les jeux de données à envoyé et les envoies depuis l'application web.		L'opérateur envoie les données qu'il souhaite faire traiter par le système.	
4	Le système traite et converti le flux en données exploitable.		Le système à converti les données dans le format du système.	
5	Le système effectue la vérification des données		Le système vérifie la validité des données et prévien l'utilisateur que les données sont invalides (si celle-ci sont invalides).	
6	Le système effectue l'analyse des données afin de modifier son comportement ou d'apprendre de nouveaux flux d'entrée.		Le système exécute l'analyse des données que lui envoie l'opérateur de l'application.	
7	Le système sauvegarde les modification de son comportement dans la base de données correspondante.		Le système stocke le modification de son comportement dans la base de données correspondante.	

8	Le système renvoie le résultat de son apprentissage sur l'application web		Le système envoie le résultat de son apprentissage sous une forme lisible afin que l'entraîneur puisse voir les modifications de comportement effectuées.	
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
Commentaire :			Approbation :	
Fiches d'anomalies émises :				

Test Fonctionnel : [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 présente dans le même réseau interne.		
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre		L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web.	
2	L'entraîneur clique sur un lien affichant les données		L'entraîneur arrive sur la page de consultation des différentes tables	
3	L'entraîneur choisi les données qu'il souhaite consulté depuis une liste		L'entraîneur choisi les données et la page charge le contenu à afficher.	
4	Le système interroge la table mentionnée ci-dessus.		Le système consulte la base de donnée souhaitée par l'entraîneur	
5	Le système renvoie le contenu de la table.		Le système renvoie le résultat de la requête vers la page de consultation.	
6	Le résultat est affiché sur la page de consultation.		La page affiche le contenu de la table.	
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
Commentaire :			Approbation :	
Fiches d'anomalies émises :				

Test Fonctionnel : [UC-006] Ajustement du comportement				
Objectif :		Vérifier que le système détecte correctement les données erronées.		
Éléments à tester :		Réception des données externe 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 présente dans le même réseau interne.		
Scénario :				
Id	Démarche	Donnée s	Comportement attendu	OK ?
1	On se connecte au système depuis un navigateur web et la page principal de l'application s'ouvre		L'application web s'ouvre et l'opérateur se retrouve sur la page principal de l'application web.	
2	Le testeur clique sur un lien permettant la modification du comportement de la machine		Le testeur arrive sur une page comportement l'ensemble des options de modification du comportement de la machine.	
3	Le testeur envoie les modifications de comportement de la machine.		La page de modification du comportement se mets en attente de confirmation du comportement de la machine d'apprentissage.	
4	Le module d'analyse répercute les modifications sur son comportement en lançant une nouvelle phase d'apprentissage		Le système exécute une phase complète d'analyse et d'apprentissage afin de corriger son comportement.	
5	Le module d'analyse envoie une confirmation de la modification de son comportement.		La page de modification du comportement affiche le résultat de la modification.	
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

Commentaire :

Approbation :

Fiches d'anomalies émises :

7. Gestion des anomalies

Afin de définir les modalités de gestions des anomalies, nous allons utilisé un logiciel de Bug Tracking de type (*MantisBT*) afin de pouvoir référencer facilement les différents bugs que nous allons rencontrer durant le développement du projet. De plus, il permettra au responsable qualité d'affecter le correctif de bug aux membres les plus aptes à les corriger.

Nous allons donc gérer le journal des tests ainsi que le suivi des faits techniques directement dans le logiciel de bug tracking.

Décrire les modalités de gestion des anomalies découvertes pendant les tests (référencement et suivi des faits techniques, journaux de tests, etc.)

8. Procédures de test

CF Partie 6

Pour chaque cas d'utilisation, décrire les différentes procédures imaginées pour vérifier le respect des exigences de la spécification technique.

On pourra utiliser un tableau comme celui proposé ci-après :

Objet testé : <logiciel testé>		Version : <version du logiciel testé>		
Objectif de test :				
Procédure n° <numéro et titre de la procédure>				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	La première action que doit réaliser le testeur	Ce que le logiciel doit fournir comme résultat.	Exigences vérifiées	
2	La deuxième action que doit réaliser le testeur	Ce que le logiciel doit fournir comme résultat.	Exigences vérifiées	
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

9. Jeux de données de test :

Format des logs :

Décrire précisément les données utilisées pour les tests.

Chaque jeu de données sera constitué d'un ensemble de valeurs explicites et cohérentes.

Chaque jeu de données pourra être associé à un identifiant et ainsi être référencé dans les procédures de tests.

Par exemple :

[J1] : Nom : MARTIN, Prénom : Paul ; Date de naissance : 19/07/1987, Adresse : Rue Alexandre Dumas 75000 PARIS, ...

Pour une procédure : Entrer les données [J1]

10. Couverture de test

CF Partie 6

Reprendre les exigences de la STB et préciser, pour chacune d'entre elles, la méthode de vérification (démonstration / tests) et, éventuellement, les procédures de tests mises en œuvre. Ces informations pourront être présentées dans un tableau du type suivant :

<i>Id Exigence STB</i>	<i>Méthode de vérification</i>	<i>Procédures utilisées</i>	<i>Commentaire</i>

Ce tableau doit permettre de vérifier que chaque exigence a fait l'objet d'une vérification.