

# Cahier de recette

<b>Version</b>	1.1
<b>Date</b>	28 février 2013
<b>Rédigé par</b>	Charles Ango – Jean-Baptiste Souchal
<b>Relu par</b>	Julien Legras
<b>Approuvé par</b>	Magali Bardet

## MISES À JOUR

Version	Date	Modifications réalisées
0.1	23/11/12	Création
0.2	30/11/12	Ajout des procédures de test
0.3	07/12/12	Modifications et suppression de procédures de test
1.1	28/02/13	Mise à jour après retours

## Table des matières

1	Objet	4
2	Documents applicables et de référence	4
3	Terminologie et sigles utilisés	4
4	Environnement de test	4
5	Responsabilités	4
6	Stratégie de tests	4
7	Gestion des anomalies	5
8	Procédures de test	5

## 1 Objet

Ce document a pour but de présenter une série de scénarios décrivant avec précision les démarches à suivre dans le cadre de l'utilisation du logiciel «chat sécurisé». Les différents tests s'exécuteront dans un environnement physique et virtuel (des machines virtuelles seront utilisées pour tester les différents modules clients de l'application).

## 2 Documents applicables et de référence

- Spécification technique des besoins
- Document d'architecture logicielle
- RFC 2810 à 2813 d'avril 2000

## 3 Terminologie et sigles utilisés

- **AC** : Autorité de certification
- **DAL** : Document d'architecture logicielle

## 4 Environnement de test

Les tests se feront dans un réseau local et par internet (pour tester les protocoles déjà existants ie. IRC, XMPP). Nous disposerons de machines dont l'utilisation sera faite comme suit :

- un serveur sécurisé (machine virtuelle)
- un serveur local (pour les tests sur réseau local, machine virtuelle)
- 5 machines clients (machines physiques ou virtuelles)
- une connexion internet (pour les tests sur internet)

La configuration matérielle utilisée a été détaillée dans le DAL. Les tests seront effectués sur le site de la faculté.

## 5 Responsabilités

- Testeur : Charles Ango
  - ↔ Le testeur sera chargé de faire les tests des différentes itérations des développeurs.
- Client : Jean-Baptiste Souchal
  - ↔ Le client valide les tests effectués par le testeur.

## 6 Stratégie de tests

Le testeur met en place l'architecture de test de l'application. Il débute les tests dans l'ordre qui était destiné d'importance des exigences fonctionnelles.

Un test est validé par le testeur lorsqu'il répond à l'exigence fonctionnelle à laquelle il est lié. Si le test n'est pas validé, alors les tests sont arrêtés. Un document sera rempli afin de gérer l'historique des tests qui n'ont pas été validés.

Les tests non validés seront renvoyé aux développeurs. Quand les développeurs auront terminé les corrections, les tests reprendront là où ils s'étaient arrêtés.

Après avoir effectué tous les tests, les résultats des tests seront envoyés au client pour une validation.

## 7 Gestion des anomalies

Lors des tests, chaque anomalie découverte sera consignée dans un tableur. Ce tableur contiendra :

- le cas de test
- la tâche impliquée
- la description de l'erreur
- les développeurs assignés à cette tâche
- le statut de l'erreur

Le responsable de la tâche à laquelle se rattache l'anomalie détectée, se verra chargé de résoudre l'anomalie. Il devra résoudre l'anomalie dans un maximum de 1 jour. Si au bout de ce délai, l'anomalie n'est toujours pas résolue, le nombre de développeur chargé de gérer l'anomalie sera augmenté en fonction de la complexité de l'anomalie et de la difficulté qu'a le responsable de la tâche à résoudre le problème.

## 8 Procédures de test

cf. Document Excel annexe