

CAHIER DE RECETTE

Version : 1.5

Date : 19 Avril 2022

Rédigé par : MENUDE / TOURE

Relu par : BELABDOUN / DURAND / FONTENIL / MENUDE / TOURE

Approuvé par : BARDET MAGALI

Signature :

MISES À JOUR

Version	Date	Modifications réalisées
0.1	28/11/2021	Création du document
0.2	04/12/2021	Mise à niveau
1.0	26/01/2022	Version rendue pour la revue de projet.
1.1	25/02/2022	Ajout de validations de tests
1.2	08/03/2022	Ajout de validations de tests
1.3	01/04/2022	Ajout de validations de tests
1.4	09/04/2022	Ajout de validations de tests
1.5	19/04/2020	Ajout de validations de tests

1. Introduction :

Dans le cadre de la réalisation du projet DataMatrix, ce document permettra à la cliente de valider les différentes fonctionnalités de 2DMatrix avant sa mise en production. Ainsi, il présentera tous les tests qui seront appliqués sur ce dernier. Il vérifiera la conformité entre 2DMatrix et les fonctionnalités citées dans la Spécification Technique des Besoins (STB). Pour ce faire, nous établirons un plan de test en partant d'abord d'une liste de tests unitaires utilisés pendant le développement des fonctionnalités. Ensuite nous exposerons les tests fonctionnels après le développement de chaque fonctionnalité et allons finir par les tests de sécurité en se basant sur les exigences de sécurité définies dans le Référentiel Général de Sécurité de l'ANSSI.

Pour terminer, les différents tests fonctionnels seront réalisés sur les fonctionnalités suivantes :

Id	Intitulé
EF-STB-1.1	Scan d'un 2D-Doc interne
EF-STB-1.2	Scan d'un pass sanitaire
EF-STB-2	Création d'un 2D-Doc interne
EF-STB-3	Accès à 2DMatrix-manage
EF-STB-4	Révocation de 2D-Doc interne
EF-STB-5.1	Révocation d'un Organisme
EF-STB-5.2	Suppression d'un Organisme
EF-STB-6	S'authentifier à 2DMatrix-Create
EF-STB-7.1	Ajouter un utilisateur d'organisme
EF-STB-7.2	Supprimer un utilisateur d'organisme
EF-STB-8	Ajout d'un Organisme
EF-STB-9	Modification des données brutes d'un DataMatrix scanné

Les tests unitaires seront effectués par le développeur en charge de la fonctionnalité à développer. Ils serviront principalement à tester le Back-End de chaque outil pour s'assurer qu'ils ont un comportement conforme aux attentes décrites dans les précédents documents, à savoir la spécification technique des besoins et le document d'architecture technique.

Les tests fonctionnels seront opérés pour vérifier cette fois-ci la conformité du front-end des outils de manière à s'assurer que l'ensemble des sous composants de nos outils produisent les résultats attendus. Ces derniers seront à la charge du développeur du sous-composant.

Les tests de sécurité permettront de réduire tous les risques qui menacent le projet. Alors l'objectif principal de ces tests sera d'écarter les principales failles de sécurité exploitables sur une application web (Injection SQL, XSS) qui pourraient nuire au bon fonctionnement de 2DMatrix.

2. Documents applicables et de référence

Dans la réalisation de ce document, nous avons eu recours à certains documents références importants qui sont listés ci-dessous :

Spécification technique des besoins (STB) : Ce document permet de connaître les besoins du projet et donc les conclusions auxquelles les tests doivent nous amener.

Document d'architecture technique (DAT) : L'utilisation de ce document nous permet de connaître les différentes fonctionnalités à tester, car ce dernier décrit l'architecture technique du projet répondant aux besoins énoncés dans le document ci-dessus. (STB)

3. Terminologie et sigles utilisés

Test unitaire : Les tests unitaires consistent à tester individuellement les composants de l'application et vérifier sa conformité vis-à-vis des différentes spécifications.

Test fonctionnel : Vérifier la conformité de l'application développée en se basant sur les spécifications fonctionnelles.

Test de sécurité : Vérifier la conformité du projet en termes de sécurité (robustesse des programmes, du matériel d'hébergement).

EF-STB-X : Exigence fonctionnelle numéro X du document de spécification technique du besoin.

2DMatrix : L'ensemble des solutions mises en place pour permettre la génération et la lecture de DataMatrix respectant le protocole 2D-Doc.

Sous-composant : Il est un constituant des composants décrits dans le document d'architecture technique. (*voir section 5.4*)

Responsive : Affichage dynamique selon la taille de l'écran pour l'affichage dans un navigateur.

ANSSI : Agence Nationale de la Sécurité des Systèmes d'Information

4. Environnement de test

Les tests unitaires seront réalisés par le développeur en charge de la fonctionnalité concernée. Il devra effectuer les tests présents dans ce document concernant sa ou ses fonctionnalités pour vérifier si ces derniers sont conformes aux attentes. Ces tests seront majoritairement effectués en local sur l'espace de développement de la personne en charge. Ces tests permettront un déploiement serein sur les serveurs Apache hébergés par nos machines virtuelles. Ce mécanisme pourra être intégré à notre déploiement automatique lors d'un changement du code sur le dépôt Git.

L'ensemble des tests fonctionnels devront valider la conformité des affichages, des comportements de nos outils. Ces derniers seront exécutés sur les différents navigateurs web actuels (Firefox, Chrome, Safari, Opera) et pareillement avec différentes tailles d'écrans (écran

smartphone, écran d'ordinateur portable, écran fixe). En effet, les tests seront accomplis sur différentes tailles d'écran pour vérifier que les sites s'afficheront de manière *responsive*.

Pour l'application mobile, les tests fonctionnels seront réalisés directement sur l'application et en amont via les outils de prévisualisation d'Android Studio lors du développement.

Pour tester correctement les différentes fonctionnalités, plusieurs jeux de données seront utilisables. Par exemple, pour valider le fonctionnement du système d'authentification, avec des comptes préenregistrés dans la base de données, ou encore l'utilisation de DataMatrix 2D-Doc existant pour tester la lecture de DataMatrix.

5. Responsabilités

C'est pendant le développement que les tests unitaires seront créés. En effet, après la réalisation d'une fonctionnalité, un deuxième développeur devra produire les tests unitaires et sera responsable de ceux-ci. Il devra de plus produire des jeux de données s'ils sont requis dans la phase de test de cette fonctionnalité. Les tests fonctionnels attachés à la fonctionnalité seront également conçus par ce même développeur. Avant chaque livraison, les différents tests fonctionnels seront effectués par l'équipe afin de valider le bon fonctionnement du livrable.

6. Stratégie de tests

Le déploiement de 2DMatrix requiert préalablement une phase de test dans le but de s'assurer qu'il est conforme aux besoins de la cliente. C'est une étape importante du processus de déploiement et garantira la qualité et la fiabilité de 2DMatrix. Nous soumettrons ainsi 2DMatrix à trois types de test qui seront exécutés dans l'ordre suivant :

- Tests unitaires : Ils interviennent lors de la phase de développement et s'appliquent sur toutes les unités de code dans le but de vérifier leur bon fonctionnement. En effet, c'est le bon fonctionnement de chacune des fonctions du code qui implique un système global sans bug. Ainsi, nous donnerons en entrée aux différentes fonctions un ensemble de données pour le test et contrôlerons si elles sont capables de bien réagir quel que soit le cas de figure.
- Tests fonctionnels : Nous établirons une liste de scénario pour tester toutes les différentes fonctionnalités citées dans le cahier de charges pour constater leurs comportements. Nous suivrons les étapes de test suivantes :
 - Création de client HTTP
 - Effectuer une requête HTTP sur la page (fonctionnalité) que nous devons tester sur les différents navigateurs web actuels (Firefox, Chrome, Safari, Opera...).
 - S'assurer que les fonctionnalités sur la page testée sont bien présentes et respectent leurs exigences.Tous les scénarios seront détaillés dans la suite du document, plus précisément dans les procédures de test.
- Tests sécurité : ces tests doivent répondre aux différentes exigences de sécurité définies dans le Référentiel Général de Sécurité de l'ANSSI. Ainsi, l'utilisation de scanners de vulnérabilités appuyée également par des tests manuels seront nécessaires pour cette étape.

7. Gestion des anomalies

La gestion des anomalies permet d'éviter de nombreux risques lors de la phase de développement afin d'obtenir un produit conforme aux spécifications.

Les anomalies seront déclarées directement avec la section *Issues* du GitLab. Cet ajout informe l'ensemble de l'équipe et permet un suivi de l'erreur pour communiquer sur l'avancement de la résolution de l'anomalie. Chacun aura donc la possibilité de créer une Issue sur GitLab pour avoir une vision sur l'anomalie. De plus, les Issues notifient directement l'équipe de développement via un WebHook sur notre plateforme de communication (Discord) pour avoir une visibilité de l'ensemble des *issues* assez rapidement. Si la cliente détecte une anomalie lors du test de son livrable, l'équipe se réunira pour créer des tâches de résolutions de bug et les tests de la fonctionnalité seront à nouveau effectués.

Les grandes étapes dans la gestion des anomalies seront les suivantes :

- Production de tests unitaires pour chaque fonctionnalité
- Refaire les tests unitaires lors d'un ajout d'une nouvelle version d'une fonctionnalité.
- Lister les anomalies connues et les répartir entre les membres du groupe pour les résoudre avant une nouvelle livraison.
- Effectuer des tests fonctionnels régulièrement (à chaque développement de fonctionnalité) pour repérer de nouvelles anomalies.
- Discuter avec le reste de l'équipe en cas de problème sur la résolution d'une anomalie pouvant conduire à un blocage général du projet.
- Lister également les failles de sécurité qui pourraient nuire au projet, les répartir entre les membres du groupe qui se chargeront de trouver des mesures de sécurité et de les appliquer.

8. Procédures de test

Légende :

- **OK** : Test passé avec succès
 - **NOK** : Test non réussi
 - - : Test pas encore effectué, car la fonctionnalité n'est pas déployée
- Scénario de tests fonctionnels

Objet testé : 2DMatrix-scan		Version : 0.1		
Objectif de test : Vérifier que l’outil de scan est conforme à sa spécification.				
Procédure n° 1				
N°	Actions	Résultats attendus	Exig.	OK/N OK/-
1	Envoi ¹ d’un DataMatrix 2D-Doc	Le DataMatrix est reconnu comme	EF-STB-1.1	OK

¹ Par envoi, on entend envoi d'un DataMatrix 2D-Doc ou d'un QR-Code à l'outil de scan 2DMatrix-Scan

	interne valide.	valide et les données brutes ne sont pas altérées.		
2	Envoi d'un DataMatrix 2D-Doc interne invalide.	Le DataMatrix est reconnu comme invalide.	EF-STB-1.1	OK
3	Envoi d'un DataMatrix 2D-Doc révoqué.	Le DataMatrix est reconnu comme invalide.	EF-STB-1.1	OK
4	Envoi d'un QR-Code de pass sanitaire valide.	Le Pass sanitaire est reconnu comme valide.	EF-STB-1.2	OK
5	Envoi d'un DataMatrix 2D-Doc interne dont le certificat est expiré.	Le DataMatrix est reconnu comme invalide.	EF-STB-1.1 EF-STB-1.2	OK
6	Modification des données brutes d'un DataMatrix 2D-Doc interne	Le DataMatrix régénéré contient bien les nouvelles données, et la même signature si on le repasse à 2DMatrix-Scan.	EF-STB-9	OK

Objet testé : 2DMatrix-Create		Version : 0.1		
Objectif de test : Vérifier que l'outil de création est conforme à sa spécification.				
Procédure n° 2				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	Connexion à l'outil de création avec un compte d'utilisateur d'organisme/	La connexion est réussie, l'utilisateur organisme est redirigé vers la page de création.	EF-STB-6	OK
2	Connexion à l'outil de création sans compte utilisateur d'organisme	L'utilisateur est redirigé vers la page de connexion.	EF-STB-6	OK
3	Connexion à l'outil de création avec des informations d'identifications incorrectes.	L'utilisateur est redirigé vers la page de connexion.	EF-STB-6	OK
4	Création d'un DataMatrix 2D-Doc interne avec cet outil	Le code est reconnu comme valide par 2DMatrix-Scan.	EF-STB-2	OK
5	Se connecter à l'outil de création, révoquer un DataMatrix 2D-Doc interne et l'envoyer dans 2DMatrix-Scan.	Le DataMatrix est reconnu comme invalide.	EF-STB-4	OK

Objet testé : 2DMatrix-Manage			Version : 0.1	
Objectif de test : Vérifier que l'outil de gestion est conforme à sa spécification.				
Procédure n° 3				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	Connexion à l'outil de gestion en tant que Super Administrateur par TLS/SSL.	La connexion est réussie, le Super Administrateur est redirigé vers sa page d'administration.	EF-STB-3	OK
2	Connexion à l'outil de gestion en tant que Super Administrateur en envoyant des identifiants invalides (certificat).	La connexion échoue. Un message d'erreur est affiché.	EF-STB-3	OK
3	Connexion à l'outil de gestion en tant que Super Administrateur via l'URL. (sans certificat)	La page n'est pas accessible. Un message d'erreur est affiché.	EF-STB-3	OK
4	Connexion à l'outil de gestion en tant qu'administrateur Organisme via un compte existant.	La connexion est réussie, l'administrateur organisme est redirigé vers l'interface de gestion.	EF-STB-3	OK
5	Se connecter à l'outil de gestion, révoquer un de nos organismes existant et envoyer un de ses DataMatrix dans 2DMatrix-Scan.	Le DataMatrix est reconnu comme invalide.	EF-STB-5.1	OK
6	Se connecter à l'outil de gestion, supprimer un de nos organismes existant et envoyer un de ses DataMatrix dans 2DMatrix-Scan.	Le DataMatrix est reconnu comme valide.	EF-STB-5.2	OK
7	Se connecter à l'outil de gestion, ajouter un utilisateur organisme non existant et se connecter à 2DMatrix-Manage avec ce compte.	La connexion est réussie, l'utilisateur organisme est redirigé vers la page de l'outil create.	EF-STB-7.1	OK
8	Se connecter à l'outil de gestion, supprimer un utilisateur organisme existant et tenter de se connecter à 2DMatrix-Manage avec ce compte.	La connexion échoue et l'utilisateur en est informé.	EF-STB-7.2	OK
9	Se connecter à l'outil de gestion, créer un nouvel organisme (non existant), générer un DataMatrix interne et envoyer à 2DMatrix-Scan	Le DataMatrix est reconnu comme valide.	EF-STB-8	OK

Objet testé : La base de données		Version : 0.1		
Objectif de test : Vérifier que la base de données est sécurisée.				
Procédure n° 4				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	Envoi de requêtes SQL dans les champs de saisi des différents formulaires	Aucune réponse envoyée par la base de données	EQ-STB-06	OK

Objet testé : La page web		Version : 0.1		
Objectif de test : Vérifier que la page web est sécurisée.				
Procédure n° 5				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	Tenter d'effectuer des injections (XSS, SQL) via les formulaires	Les injections ne fonctionnent pas	EQ-STB-05	OK
2	Insérer dans l'outil 2DMatrix-Scan un Datamatrix avec une mauvaise extension (faille upload)	L'insertion est impossible	EQ-STB-05	OK

Objet testé : La connexion à un service tiers		Version : 0.1		
Objectif de test : Vérifier que 2DMatrix n'utilise pas de service tiers dans son fonctionnement				
Procédure n° 6				
N°	Actions	Résultats attendus	Exig.	OK/ NOK
1	Pour 2DMatrix-Scan : Utilisation de l'outil en hors ligne	Aucune anomalie ou restriction dans le fonctionnement	EQ-STB-07	OK
2	Pour tous les outils : Utilisation d'un logiciel permettant d'analyser les différentes requêtes émises/recues	Aucune requête (entrante ou sortante), hormis celles en lien avec nos serveurs	EQ-STB-07	OK

9. Jeux de données de test :

Les jeux de données sont à insérer dans les différents outils 2DMatrix pour effectuer les différents tests. Il s'agira de construire des données spécialement pour chaque test en entrée et en sortie. Les résultats obtenus avec les données en entrée devront être comparés avec ceux de la sortie. Alors, nous aurons :

- Pour le scan :
Une base de données d'image en entrée contenant d'une part des images de datamatrix (resp. Pass Sanitaires) valides et d'autre part des images invalides (révoquées, certificat invalide, autre code barres).
Les données attendues en sortie sont les messages stipulant la validité ou l'invalidité du datamatrix/Pass Sanitaire.
- Pour la création d'un datamatrix 2D-Doc :
On fournira en entrée un ensemble de données dont chaque élément sera énuméré selon la disposition suivante :
 - Date
 - Nom prénom
 - Numéro du document
 - MoyenneLes données attendues en sortie sont des datamatrix contenant un 2d-doc valide..
- Pour l'authentification aux outils de gestion :
Les données d'entrées sont des identifiants des comptes administrateurs organismes et super administrateur valides d'une part et de l'autre invalides.
Ces données sont insérées dans l'outil selon l'ordre suivant :
 - Identifiant
 - Mot de passeLes données attendues en sortie informent de la réussite ou de l'échec de la connexion.

10. Couverture de test

<i>Id Exigence STB</i>	<i>Méthode de vérification</i>	<i>Procédures utilisées</i>	<i>Commentaire</i>
EF-STB-1.1	Test unitaire, test fonctionnel	1	
EF-STB-1.2	Test unitaire, test fonctionnel	1	
EF-STB-2	Test unitaire, test fonctionnel	2	
EF-STB-3	Test unitaire, test fonctionnel	3	
EF-STB-4	Test unitaire, test fonctionnel	3	
EF-STB-5.1	Test unitaire, test fonctionnel	3, 5	
EF-STB-5.2	Test unitaire, test	3, 5	

	fonctionnel		
EF-STB-6	Test unitaire, test fonctionnel, test de sécurité	2, 4	
EF-STB-7.1	Test unitaire, test fonctionnel, test de sécurité	3, 6	
EF-STB-7.2	Test unitaire, test fonctionnel, test de sécurité	3, 6	
EF-STB-8	Test unitaire, test fonctionnel	3	
EF-STB-9	Test unitaire, test fonctionnel	1	