|  |
| --- |
| Abris de recharge – Consommation électrique |
| Dossier de Tests de Validation |

**Référence :** **ConsoElectriqueAbris/DTV\_ ConsoElectriqueAbris-V3.0**

**Date :** **07/05/2024**

HISTORIQUE DES RÉVISIONS DU DTV

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Commentaires** |
| **1.0** | 09/02/24 | Version initiale |
| **2.0** | 23/02/24 | Version révisée |
| **3.0** | 07/05/24 | Version finale |

SOMMAIRE

1. INTRODUCTION 4

1.1 Objet 4

1.2 Documents de référence 4

2. DESCRIPTION DE L’ENVIRONNEMENT DE TESTS 5

2.1 Configuration matérielle et logicielle 5

2.1.1 Généralités 5

2.1.2 Configuration du radio réveil pour l’exécution du plan de test 5

3. FICHES DE TESTS 6

3.1 Faire une demande 7

3.2 Gestion des demandes 9

3.3 Consulter l’état d’une demande 13

3.4 Gérer les barrières 17

3.5 Supervision du système 24

# INTRODUCTION

## Objet

Le présent document constitue le dossier de test et de validation du système de consommation électrique de l’abris à bicycles rechargeables.

## Documents de référence

|  |  |
| --- | --- |
| **Référence** | **Titre** |
| Diagramme des cas d’utilisation simplifié | Diagramme des cas d’utilisation du projet |
| Diagramme des exigences | Diagramme des exigences du projet |

# DESCRIPTION DE L’ENVIRONNEMENT DE TESTS

## Configuration matérielle et logicielle

### Généralités

Ce cahier de recette sert à valider le fonctionnement du système de consommation électrique de l’abris à bicycles rechargeables avant sa livraison au client pour son utilisation ; Le bon fonctionnement du système de récupération des données de consommation, notamment leur affichage sur le site web. L’ensemble des fonctionnalités du système sont validées par ce document.

A l’issue de l’exécution du plan de tests prévu dans ce document, le document ainsi obtenu est transformé en RTV (Rapport de Tests de Validation), ce RTV permettra de connaître l’état de validation du système au moment où il a été réalisé. En cas d’identification de non-conformités, le RTV servira à la correction de celle-ci ; il faut donc, lorsqu’une non-conformité est constatée, détailler au maximum le problème constaté et dans quel cas celle-ci s’est présentée. La section observation de chaque cas de test sert à détailler les conditions d’apparition des non-conformités.

Comme indiqué précédemment le plan de tests prévu va permettre de valider l’ensemble du fonctionnement du système tel que celui-ci est prévu dans les documents de spécification à savoir :

* Le diagramme des cas d’utilisation validé avec le client, qui recense l’ensemble des fonctionnalités du système à valider ;
* Le diagramme des exigences validé avec le client, qui recense l’ensemble des contraintes que le système doit respecter.

Le plan de test décrit dans ce document vérifie que l’ensemble des demandes du client et des spécifications sont respectées. L’objectif étant de vérifier que le produit est conforme aux attentes du client.

### Configuration du système pour l’exécution du plan de test

Le système de consommation électrique de l’abris à bicycles rechargeables utilisé pour l’exécution du plan de tests doit être dans sa configuration d’usine afin de s’assurer qu’aucune manipulation ne puisse altérer le résultat du plan de tests prévu.

### Configuration du site web pour l’exécution du plan de test

Le site web de l'abri utilisé pour l'exécution du plan de tests doit être accessible et fonctionnel. Il doit être configuré de manière à ce que les données de consommation puissent être récupérées et affichées correctement.

# FICHES DE TESTS

Le présent chapitre contient les fiches de tests suivantes :

Réf. : FE1.1 : Récupérer les informations de chacune des box à vélo / Récupérer les données des capteurs 7

Réf. : FE1.2 : Récupérer les informations de chacune des box à vélo / Connexion au Node.js 8

Réf. : FE1.3 : Récupérer les informations de chacune des box à vélo / Envoi des données au Node.js 9

Réf. : FE2.1 : Récupérer les informations du capteur de luminosité / Récupérer les données du capteur 10

Réf. : FE2.2 : Récupérer les informations du capteur de luminosité / Connexion au Node.js 11

Réf. : FE2.3 : Récupérer les informations du capteur de luminosité / Envoi des données au Node.js 12

Réf. : FE3.1 : Serveur Node.js / Connexion aux programmes C++ 13

Réf. : FE3.2 : Serveur Node.js / Récupérer les données des programmes C++ 14

Réf. : FE3.3 : Serveur Node.js / Calculer les proportions et les moyennes de temps vert 15

Réf. : FE3.4 : Serveur Node.js / Envoyer les données à l'API CRUD Consommation électrique 16

Réf. : FE3.5 : Serveur Node.js / Envoyer les données à l'API CRUD Gestion Accès 17

Réf. : FE4.1: Gérer les barrière/ Se connecter 17

Réf. : FE4.2: Gérer les barrière/ choisir mode 18

Réf. : FE5.1: Réaction présence voiture/ mode cas par cas 19

Réf. : FE4.3: Gérer les barrière/ choisir mode 20

Réf. : FE5.2: Réaction présence voiture/ mode gestion globale 21

Réf. : FE4.4: Gérer les barrière/ choisir mode 22

Réf. : FE5.2: Réaction présence voiture/ mode manuel 23

Réf. : FE6.1: supervision du système/ Se connecter 24

Réf. : FE6.1: Supervision du système/ afficher l’état du système 25

## Programmes C++

### Récupérer les informations des box à vélo

Les tests suivants permettent de vérifier la bonne récupération des données des box à vélo en amont.

|  |  |  |
| --- | --- | --- |
| Réf. : FE1.1 : Récupérer les informations de chacune des box à vélo / Récupérer les données des capteurs | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons dans un premier temps vérifier qu’il est possible de récupérer les informations des différents capteurs de l’abris. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Récupération des informations. | Les informations sont récupérées et sont utilisables (sans pertes ou fausses informations). |  |
| Erreur de récupération des informations | Un message d’erreur apparait pour indiquer une erreur lors de la récupération des informations et indiquer de quel capteur l’erreur vient. |  |
| Reformatage des informations. | Les informations sont correctement reformatées et prêtent à être utilisées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE1.2 : Récupérer les informations de chacune des box à vélo / Connexion au Node.js | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons de vérifier qu’il est possible de se connecter au serveur Node.js. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Connexion au serveur Node.js. | La connexion au serveur Node.js est établie et est opérationnelle. |  |
| Erreur de connexion au serveur Node.js | Un message d’erreur apparait si la connexion au serveur Node.js n’a pas pu être établie. |  |
| Reconnexion au serveur Node.js. | * Le programme C++ tente de se reconnecter à intervalles réguliers. * La connexion au serveur Node.js est rétablie et est opérationnelle. |  |
| Déconnexion du serveur Node.js. | La connexion au serveur Node.js est rompue. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE1.3 : Récupérer les informations de chacune des box à vélo / Envoi des données au Node.js | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons de vérifier qu’il est possible d’envoyer les données au serveur Node.js. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Afficher les données à envoyer. | Les données à envoyer sont affichées. |  |
| Gérer la file d’attente. | Si la connexion au serveur Node.js n’est pas établie :   * Si la file d’attente est pleine, le couple d’information le plus ancien est retiré et un message apparait. * Sinon, le couple d’information qui n’a pas pu être envoyé est ajouté à la file d’attente et un message apparait.   Si la connexion au serveur Node.js est établie :   * Le couple d’informations est ajouté à la file d’attente et un message apparait. |  |
| Envoi des données au serveur Node.js. | Toutes les données de la file d’attente sont correctement envoyées au serveur Node.js. |  |
| Erreur d’envoi des données au serveur Node.js. | Un message d’erreur apparait pour dire que les données n’ont pas pu être envoyées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

### Récupérer les informations du capteur de luminosité

Les tests suivants permettent de vérifier la bonne récupération des données du capteur de luminosité en amont.

|  |  |  |
| --- | --- | --- |
| Réf. : FE2.1 : Récupérer les informations du capteur de luminosité / Récupérer les données du capteur | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons dans un premier temps vérifier qu’il est possible de récupérer les informations du capteur de luminosité. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Récupération des informations. | Les informations sont récupérées et sont utilisables (sans pertes ou fausses informations). |  |
| Erreur de récupération des informations | Un message d’erreur apparait pour indiquer une erreur lors de la récupération des informations et indiquer de quel capteur l’erreur vient. |  |
| Reformatage des informations. | Les informations sont correctement reformatées et prêtent à être utilisées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE2.2 : Récupérer les informations du capteur de luminosité / Connexion au Node.js | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons de vérifier qu’il est possible de se connecter au serveur Node.js. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Connexion au serveur Node.js. | La connexion au serveur Node.js est établie et est opérationnelle. |  |
| Erreur de connexion au serveur Node.js | Un message d’erreur apparait si la connexion au serveur Node.js n’a pas pu être établie. |  |
| Reconnexion au serveur Node.js. | * Le programme C++ tente de se reconnecter à intervalles réguliers. * La connexion au serveur Node.js est rétablie et est opérationnelle. |  |
| Déconnexion du serveur Node.js. | La connexion au serveur Node.js est rompue. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE2.3 : Récupérer les informations du capteur de luminosité / Envoi des données au Node.js | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le système de consommation électrique de l’abris à bicycles rechargeables étant dans sa configuration d’usine. Nous devons de vérifier qu’il est possible d’envoyer les données au serveur Node.js. Les tests seront effectués à l’intérieur d’un programme C++. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Afficher les données à envoyer. | Les données à envoyer sont affichées. |  |
| Gérer la file d’attente. | Si la connexion au serveur Node.js n’est pas établie :   * Si la file d’attente est pleine, le couple d’information le plus ancien est retiré et un message apparait. * Sinon, le couple d’information qui n’a pas pu être envoyé est ajouté à la file d’attente et un message apparait.   Si la connexion au serveur Node.js est établie :   * Le couple d’informations est ajouté à la file d’attente et un message apparait. |  |
| Envoi des données au serveur Node.js. | Toutes les données de la file d’attente sont correctement envoyées au serveur Node.js. |  |
| Erreur d’envoi des données au serveur Node.js. | Un message d’erreur apparait pour dire que les données n’ont pas pu être envoyées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

## Serveur Node.js

### Connexion aux programmes C++

Les tests suivants permettent de vérifier la connexion entre le serveur Node.js et les deux programmes C++.

|  |  |  |
| --- | --- | --- |
| Réf. : FE3.1 : Serveur Node.js / Connexion aux programmes C++ | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le serveur Node.js sera testé sur une machine virtuelle (VM) configurée à l’avance dans un mode de fonctionnement normal. Les tests seront effectués à l’intérieur d’un programme Javascript. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Connexion aux programmes C++. | Le serveur Node.js est connecté aux deux programmes C++. |  |
| Erreur lors de la connexion aux programmes C++. | Un message d’erreur apparait indiquant une erreur de connexion aux programmes C++ et précisant de quel programme l’erreur vient. |  |
| Reconnexion aux programmes C++. | * Le serveur Node.js tente de se reconnecter à intervalles réguliers. * La connexion aux programmes C++ est rétablie et est opérationnelle. |  |
| Déconnexion des programmes C++. | La connexion aux programmes C++ est rompue. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

### Récupérer données des programmes C++

Les tests suivants permettent de vérifier la bonne récupération des données des deux programmes C++ en amont.

|  |  |  |
| --- | --- | --- |
| Réf. : FE3.2 : Serveur Node.js / Récupérer les données des programmes C++ | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le serveur Node.js sera testé sur une machine virtuelle (VM) configurée à l’avance dans un mode de fonctionnement normal. Les tests seront effectués à l’intérieur d’un programme Javascript. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Récupération des informations. | Les informations sont récupérées et sont utilisables (sans pertes ou fausses informations). |  |
| Erreur de récupération des informations | Un message d’erreur apparait pour indiquer une erreur lors de la récupération des informations et indiquer de quel programme C++ l’erreur vient. |  |
| Reformatage des informations. | Les informations sont correctement reformatées et prêtent à être utilisées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

### Calculer les proportions et le moyennes de temps vert

Les tests suivants permettent de vérifier les calculs des proportions et des moyennes de proportions de temps vert.

|  |  |  |
| --- | --- | --- |
| Réf. : FE3.3 : Serveur Node.js / Calculer les proportions et les moyennes de temps vert | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le serveur Node.js sera testé sur une machine virtuelle (VM) configurée à l’avance dans un mode de fonctionnement normal. Les tests seront effectués à l’intérieur d’un programme Javascript. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Incrémenter les variables de temps de 5000 si les variables sont égales à 1. | Les variables de temps sont correctement incrémentées. |  |
| Arrondir les proportions à deux chiffres après la virgule. | Les proportions sont arrondies à deux chiffres après la virgule. |  |
| Stocker les proportions dans un tableau et faire leur moyenne. | Le tableau est bien créé et les moyennes stockées à l’intérieur. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

### Envoi des données aux API CRUD

Les tests suivants permettent de vérifier l’acheminement des données du serveur Node.js jusqu’aux API CRUD des groupes Abris - Gestion Accès / Consommation électrique.

|  |  |  |
| --- | --- | --- |
| Réf. : FE3.4 : Envoyer les données à l'API CRUD Consommation électrique | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le serveur Node.js sera testé sur une machine virtuelle (VM) configurée à l’avance dans un mode de fonctionnement normal. Les tests seront effectués à l’intérieur d’un programme Javascript. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Vérification des données. | Toutes les données sont définies, aucune donnée ne manque. Un message s’affiche si une valeur a changée. |  |
| Encapsuler les données au format JSON. | Les données sont correctement encapsulées au format JSON. |  |
| Réception de la réponse. | La réponse de l’API est positive et les données sont bien envoyées. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE3.5 : Envoyer les données à l'API CRUD Gestion Accès | | |
| **ENVIRONNEMENT DU TEST** | | |
| Le serveur Node.js sera testé sur une machine virtuelle (VM) configurée à l’avance dans un mode de fonctionnement normal. Les tests seront effectués à l’intérieur d’un programme Javascript. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Cliquer sur le bouton « Consulter ses demandes ». | La page pour consulter ses demandes s’affiche. |  |
| Lors de l’accès à la page, choisir une demande dans le tableau. | Affiche la demande avec tous les informations et son statut. |  |
| Demande refuser | Affiche le statut refuser dans le récapitulatif de la demande |  |
| Cliquer sur le bouton « Déconnexion ». | Redirige vers la page de connexion |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

## Gérer les barrières

Les tests suivants permettent de tester les fonctions spécifiques lors de la gestion de la barrière.

|  |  |  |
| --- | --- | --- |
| Réf. : FE4.1: Gérer les barrière/ Se connecter | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Lancer l’application. | La page de connexion s’affiche. |  |
| Accéder à la page de connexion. Remplir les informations suivantes :  Login : gardien  Password : gardien  Cliquer sur le bouton « Se connecter ». | Affichage de la page d’accueil. |  |
| Login ou password incorrect. | Un message d’erreur s’affiche. |  |
| Cliquer sur « Mot de passe oublié ». | Affiche une page de récupération de mot de passe. |  |
| Saisir un e-mail. | Saisir l’e-mail associée au compte pour recevoir un mail de récupération de compte. |  |
| Cliquer sur le lien | Affiche une page de changement de mot de passe |  |
| Saisir nouveau mot de passe :  Nouveau : password | Mot de passe changer |  |
| Accéder à la page de connexion. Remplir les informations suivantes :  Login : lucas  Password : password  Cliquer sur le bouton « Se Connecter ». | Affiche la page d’accueil. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE4.2: Gérer les barrière/ choisir mode | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Cliquer sur le bouton « Gestion mode de la barrière ». | Affichage de la page de gestion de la barrière. |  |
| Choisir un mode. | Cliquer sur un des bouton du mode en question. |  |
| Mettre en Mode cas-par-cas. | La barrière s’ouvre si le véhicule est autorisé, sinon le gardien intervient |  |
| Cliquer sur le bouton « Déconnexion ». | Redirige vers la page de connexion. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE5.1: Réaction présence voiture/ mode cas par cas | | |
| **ENVIRONNEMENT DU TEST** | | |
| Les réaction de la barrière au véhicule se fait sur maquette | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Mettre plaque | Scan la plaque. |  |
| Plaque autorisé | La barrière s’ouvre |  |
| Plaque non autorisé | Le gardien intervient pour laisser ou non passer la voiture |  |
| Stocke les plaques | Envoie en base de donner |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE4.3: Gérer les barrière/ choisir mode | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Cliquer sur le bouton « Gestion mode de la barrière ». | Affichage de la page de gestion de la barrière. |  |
| Mettre en Mode gestion globale . | La barrière reste ouverte en permanence et enregistre les plaques. |  |
| Cliquer sur le bouton « Déconnexion ». | Redirige vers la page de connexion. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE5.2: Réaction présence voiture/ mode gestion globale | | |
| **ENVIRONNEMENT DU TEST** | | |
| Les réaction de la barrière au véhicule se fait sur maquette | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Mettre plaque | Scan la plaque. |  |
| Stocke les plaques | Envoie en base de donner |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE4.4: Gérer les barrière/ choisir mode | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Cliquer sur le bouton « Gestion mode de la barrière ». | Affichage de la page de gestion de la barrière. |  |
| Mettre en mode manuelle. | Le gardien gère la barrière. |  |
| Cliquer sur le bouton « Déconnexion ». | Redirige vers la page de connexion. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE5.2: Réaction présence voiture/ mode manuel | | |
| **ENVIRONNEMENT DU TEST** | | |
| Les réaction de la barrière au véhicule se fait sur maquette | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Mettre plaque | Scan la plaque. |  |
| Stocke les plaques | Envoie en base de donner |  |
| Clique sur bouton ouvrir | Ouvre la barrière |  |
| Clique sur le bouton fermer | Ferme la barrière |  |
| Pas de clique sur le bouton | La barrière reste fermer |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

.

## Supervision du système

Les tests suivants permettent de tester les fonctions spécifiques lors de la supervision du système

|  |  |  |
| --- | --- | --- |
| Réf. : FE6.1: supervision du système/ Se connecter | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Lancer l’application. | La page de connexion s’affiche. |  |
| Accéder à la page de connexion. Remplir les informations suivantes :  Login : gardien  Password : gardien  Cliquer sur le bouton « Se connecter ». | Affichage de la page d’accueil. |  |
| Login ou password incorrect. | Un message d’erreur s’affiche. |  |
| Cliquer sur « Mot de passe oublié ». | Affiche une page de récupération de mot de passe. |  |
| Saisir un e-mail. | Saisir l’e-mail associée au compte pour recevoir un mail de récupération de compte. |  |
| Cliquer sur le lien | Affiche une page de changement de mot de passe |  |
| Saisir nouveau mot de passe :  Nouveau : password | Mot de passe changer |  |
| Accéder à la page de connexion. Remplir les informations suivantes :  Login : lucas  Password : password  Cliquer sur le bouton « Se Connecter ». | Affiche la page d’accueil. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |

|  |  |  |
| --- | --- | --- |
| Réf. : FE6.1: Supervision du système/ afficher l’état du système | | |
| **ENVIRONNEMENT DU TEST** | | |
| L’application C++ sera testée sur un ordinateur. | | |
| **DESCRIPTION** | **CRITERES D'ACCEPTATION** | **RESULTAT** |
| Cliquer sur bouton « Supervision du système ». | Affichage de la page de supervision et envoie un Ping au différent composant. |  |
| Afficher la page. | Affiche les différents états de chaque composant du système en fonction de la réponse des Ping. |  |
| Cliquer sur le bouton « Déconnexion ». | Redirige vers la page de connexion. |  |
| **ETAT DU TEST** | | |
| Etat du test : Accepté  Refusé  Accepté sous Réserve | | |
| Observations : | | |