

Plan de tests logiciels

Version 1.0

**HISTORIQUE DES REVISIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VERSION** | **DATE** | **DESCRIPTION DES EVOLUTIONS** | **REDACTION** | **VERIFICATION** | **APPROBATION** |
| 1.0 | 04/02/2019 | Création du document |  |  |  |
| Signature | | |  |  |  |

**SOMMAIRE**

[1. INTRODUCTION 4](#_Toc532117933)

[1.1 OBJECTIF 4](#_Toc532117934)

[1.2 PORTEE 4](#_Toc532117935)

[1.3 REFERENCES 4](#_Toc532117936)

[1.4 CONVENTIONS 5](#_Toc532117937)

[1.4.1 TEST 5](#_Toc532117938)

[2. VERIFICATION DU LOGICIEL 6](#_Toc532117939)

[2.1 PLAN DE TEST 6](#_Toc532117940)

[2.1.1 ENVIRONNEMENT DE TEST 6](#_Toc532117941)

[2.1.2 INSTALLATION, MISE EN PLACE ET MAINTENANCE 6](#_Toc532117942)

[2.1.3 SUIVI DES BUGS 6](#_Toc532117943)

[2.1.4 PERSONNEL 6](#_Toc532117944)

[2.1.5 METHODES DE VERIFICATION 6](#_Toc532117945)

[2.2 DESCRIPTION DES TESTS 7](#_Toc532117946)

[2.2.1 CATEGORIES DE TEST 7](#_Toc532117947)

[2.2.2 PROGRESSION DES TESTS 8](#_Toc532117948)

[2.2.3 COUVERTURE DES TESTS 8](#_Toc532117949)

[2.2.4 IDENTIFICATION DU TEST 9](#_Toc532117950)

[2.2.5 CONTENU DES TESTS 10](#_Toc532117951)

[2.3 PLANIFICATION DES TESTS 10](#_Toc532117952)

[2.3.1 PHASE DE TEST 1 10](#_Toc532117953)

[2.3.1.1 Couverture du test 10](#_Toc532117954)

[2.3.1.2 Planification des tests 10](#_Toc532117955)

[2.3.2 PHASE DE TEST 2 11](#_Toc532117956)

[2.3.2.1 Couverture du test 11](#_Toc532117957)

[2.3.2.2 Planification des tests 11](#_Toc532117958)

# INTRODUCTION

## OBJECTIF

Ce document a pour but de présenter l’ensemble des informations nécessaires aux tests du sous-système logiciel **GP 1.0**

## PORTEE

Ce document sera utilisé par l’équipe du projet et sera disponible pour le maître d’ouvrage si celui-ci désire l’étudier et faire des remarques.

## REFERENCES

* + 1. NORMES ET REGLEMENTATIONS

|  |  |
| --- | --- |
| **TITRE** | **NOM DU DOCUMENT** |
| IEC 62304:2006/Amd 1:2015 | Processus du cycle de vie du logiciel |
| ISO 13485:2016 | Systèmes de management de la qualité -- Exigences à des fins réglementaires |
| ISO 14971:2007 | Application de la gestion des risques aux dispositifs médicaux |
| IEC 62366-1:2015 | Partie 1 : Application de l'ingénierie de l'aptitude à l'utilisation aux dispositifs médicaux |

* + 1. PROJET

|  |  |
| --- | --- |
| **TITRE** | **NOM DU DOCUMENT** |
| Plan de développement du logiciel-1.0 – Mialon, Long, Demolliens | Plan de développement du logiciel |
| Rapport de tests logiciels-1.0 – Mialon, Long, Demolliens | Rapport de tests logiciels |

## CONVENTIONS

## TEST

|  |  |  |
| --- | --- | --- |
| **Identifiant du test** | **T-XXX-000** | |
| Description | Description du test T-XXX-000 |  |
| Exigence vérifiée | XXX-000 | Méthode de vérification (I, A, D, T) |
| Conditions initiales | Etat du système avant le test | Procédure ou résultat d’un test précédent |
| Entrées | Données d’entrée | Procédure |
| Actions sur les données collectées | Enregistrement et post-traitement des données de sortie | Procédure |
| Sorties | Journaux, nom de fichier des données de sortie | Nom du fichier ou de la fiche de test |
| Hypothèses et contraintes | Si elles existent, accès limité à un outil, licence, … |  |
| Résultats attendus et critères | Liste des résultats du test | Critère d’évaluation du résultat du test |
| **Procédure de test** |  | |
| **Numéro d’étape** | **Action de l’opérateur** | **Résultat attendu et critère d’évaluation** |
| 1 | Début de foo | Foo a débuté |

# VERIFICATION DU LOGICIEL

## PLAN DE TEST

## ENVIRONNEMENT DE TEST

Les tests du projet se ferons au sein de l’université.

Le matériel de test est le suivant :

* Ordinateur avec Windows 8 min.

Les logiciels de test sont les suivants :

* Serveur Easy PHP

## INSTALLATION, MISE EN PLACE ET MAINTENANCE

La mise en place et l’installation sera l’affaire du client. Il sera fournis par la société, une explication écrite précise sur la procédure à suivre pour mener à bien l’installation ainsi que l’utilisation du logiciel.

## PERSONNEL

Le personnel habilité à effectuer les tests est le suivant :

* Responsable du service de recherche & développement : Jordan Long
* Chef de projet : Pierre Demolliens
* Ingénieur du service de recherche & développement : Jérôme Mialon

## METHODES DE VERIFICATION

Les méthodes de vérification des exigences sont définies ci-dessous :

* Inspection (I) : contrôle ou vérification visuelle
  + Contrôle de l’implémentation physique ou de l’installation du composant. Ce contrôle vérifie que l’implémentation ou l’installation du composant est en accord avec les exigences.
  + Contrôle de la documentation décrivant les composants. Ce contrôle vérifie que la documentation est en accord avec les exigences.
* Analyse (A) : vérification basée sur des preuves analytiques
  + Vérification d’une fonctionnalité, des performances ou de la solution technique d’un composant par l’analyse des données collectées lors des tests en conditions réelles, par la simulation de conditions réelles ou par l’analyse d’un rapport.
  + L’analyse des données de test est utilisée pour vérifier les exigences.
  + La vérification est basée sur les preuves analytiques obtenues par calculs tel que la modélisation, la simulation et la prévision.
  + Les analyses sont utilisées quand un niveau de confiance acceptable ne peut pas être établi par d’autres méthodes ou si l’analyse est la solution la moins coûteuse.
* Démonstration (D) : vérification des caractéristiques opérationnelles sans mesure quantitative.
  + Vérifier une exigence par démonstration implique que, la fonctionnalité requise spécifiée par une exigence, est complète.
  + La vérification d’une exigence par démonstration est utilisée quand la mesure quantitative n’est pas requise dans la vérification de cette exigence.
  + La vérification d’une exigence par démonstration inclue le contrôle des solutions techniques spécifiées par les exigences non fonctionnelles.
* Test (T) : vérification des caractéristiques quantitatives par des mesures quantitatives
  + Vérification d’une fonctionnalité, des performances ou de la solution technique d’un composant par l’exécution de scénarios de test dans des conditions prédéfinies, contrôlées et tracées.
  + Les tests nécessitent l’utilisation d’équipements spéciaux, d’instruments, de techniques de simulation ou l’application de procédures et principes établies.
  + Les données produites lors des tests sont utilisées pour évaluer les résultats quantitatifs et les comparer avec les exigences.

Pour chaque exigence logicielle, une méthode de vérification est définie.

## DESCRIPTION DES TESTS

Ce plan de test décrit tous les tests permettant de vérifier les exigences du sous-système logiciel **GP 1.0** grâce aux phases successives suivantes :

* Tests logiciels

Les exigences logicielles sont définies dans le document *Exigences logicielles*

## CATEGORIES DE TEST

* Fonctions principales

## PROGRESSION DES TESTS

* Tests logiciels : les tests sont effectués selon la justification suivante :
  + Les tests appartenant à la catégorie « fonctions principales » sont effectués en premier.
  + Les tests appartenant à la catégorie « ingénierie d’aptitude à l’utilisation » sont effectués ensuite.
  + Les tests appartenant à la catégorie « analyse de la maîtrise des risques » sont effectués ensuite.

## COUVERTURE DES TESTS

La couverture des tests dépend des phases de test suivantes :

* Les tests logiciels qui couvent le reste des exigences logicielles.

La traçabilité entre les tests définis ci-dessous et les spécifications logicielles définies dans le document *Exigences logicielles* est listée dans le document *Matrice de traçabilité des exigences*

Une spécification peut nécessiter plusieurs tests pour être vérifiée, dans ce cas, elle apparaîtra dans tous les tests qui permettent de la vérifier.

## IDENTIFICATION DU TEST

Chaque test est unique et contient :

* Un identifiant unique
* La catégorie du test
* Une description textuelle des objectifs du test
* La traçabilité de la spécification logicielle
* La méthode de vérification
* L’enregistrement des données et la procédure d’analyse
* Hypothèses et contraintes si elles existent
* Préoccupations sur la sécurité et la confidentialité si elles existent

Chaque identifiant de test possède un identifiant unique qui est l’identifiant de la spécification logicielle moins SOFTREQ puis précédée de **T-**

*Exemple pour la spécification logicielle SOFTREQ-MAIN-001, le test correspondant sera T-MAIN-001,*

Si plusieurs tests sont nécessaires pour vérifier la spécification logicielle, l’identifiant sera suivi de **-1, -2, -n**.

*Exemple T-MAIN-001-1, T-MAIN-001-2*

## CONTENU DES TESTS



|  |  |  |
| --- | --- | --- |
| **Identifiant du test** | **T-MAIN-001** | |
| Description | Vérification que l’ajout d’un patient fonctionne. | |
| Exigence vérifiée | SOFTREQ-MAIN-001 | Analyse |
| Conditions initiales | Le sous-système logiciel est dans l’état SUSPENDU |  |
| Entrées | Les informations demandées |  |
| Actions sur les données collectées | Mise en place dans une requête POST avec le protocole Fhir |  |
| Sorties |  | **R-MAIN-001** |
| Hypothèses et contraintes | Il faut que le serveur EasyPHP soit allumé.  L’utilisateur soit connecté.  Interface ajout patient affichée |  |
| Résultats attendus et critères | Le serveur est lancé la page index est affichée  Code 201 | Cf. ci-dessous |
| **Procédure de test** | | |
| **Numéro d’étape** | **Action de l’opérateur** |  |
| 1 | Saisir dans le champ nom le texte « Test » | Dans le champ nom il y a « Test » |
| 2 | Appuyer sur le bouton Ajout | Le bouton à bien été cliqué et la page change. |
| 3 | Vérification que le code est 201 et que l’ID soit affiché |  |
| 4 | Faire une récupération de donnée avec cet id. | Vérifier que le patient a pour nom « Test » |

|  |  |  |
| --- | --- | --- |
| **Identifiant du test** | **T-MAIN-002** | |
| Description | Vérification que la suppression d’un patient fonctionne. | |
| Exigence vérifiée | SOFTREQ-MAIN-002 | Analyse |
| Conditions initiales | Le sous-système logiciel est dans l’état SUSPENDU |  |
| Entrées | Les informations demandées |  |
| Actions sur les données collectées | Mise en place dans une requête DELETE avec le protocole Fhir |  |
| Sorties |  | **R-MAIN-001** |
| Hypothèses et contraintes | Il faut que le serveur EasyPHP soit allumé.  L’utilisateur soit connecté.  Interface suppression patient affichée  Le patient avec cet id existe déjà. |  |
| Résultats attendus et critères | Le patient est bien supprimé | Cf. ci-dessous |
| **Procédure de test** | | |
| **Numéro d’étape** | **Action de l’opérateur** |  |
| 1 | Saisir dans le champ id, l’id du patient à supprimer | Dans le champ nom il y a l’id rentré |
| 2 | Appuyer sur le bouton Supprimer | Le bouton à bien été cliqué et la page change. |
| 3 | Faire une récupération de donnée avec cet id. | Vérifier que la requête renvoie une erreur. |

|  |  |  |
| --- | --- | --- |
| **Identifiant du test** | **T-MAIN-003** | |
| Description | Vérification que la modification d’un patient fonctionne. | |
| Exigence vérifiée | SOFTREQ-MAIN-003 | Analyse |
| Conditions initiales | Le sous-système logiciel est dans l’état SUSPENDU |  |
| Entrées | Les informations demandées |  |
| Actions sur les données collectées | Mise en place dans une requête PUT avec le protocole Fhir |  |
| Sorties |  | **R-MAIN-001** |
| Hypothèses et contraintes | Il faut que le serveur EasyPHP soit allumé.  L’utilisateur soit connecté.  Interface modification d’un patient soit affichée  Le patient existe et on a son id |  |
| Résultats attendus et critères | Le serveur est lancé la page index est affichée | Cf. ci-dessous |
| **Procédure de test** | | |
| **Numéro d’étape** | **Action de l’opérateur** |  |
| 1 | Saisir dans le champ id l’id du patient à modifier | On vérifie qu’on à bien l’ID rentré |
| 2 | Saisir dans le champ nom le texte « Test2 » | Dans le champ nom il y a « Test2 » |
| 3 | Appuyer sur le bouton Modification | Le bouton à bien été cliqué et la page change. |
| 4 | Faire une récupération de donnée avec cet id. | Vérifier que le change du nom est bien « Test2 » |

|  |  |  |
| --- | --- | --- |
| **Identifiant du test** | **T-MAIN-004** | |
| Description | Vérification que la récupération d’un patient fonctionne. | |
| Exigence vérifiée | SOFTREQ-MAIN-004 | Inspection. |
| Conditions initiales | Le sous-système logiciel est dans l’état SUSPENDU |  |
| Entrées | Les informations demandées |  |
| Actions sur les données collectées | Mise en place dans une requête POST avec le protocole Fhir |  |
| Sorties |  | **R-MAIN-001** |
| Hypothèses et contraintes | Il faut que le serveur EasyPHP soit allumé.  L’utilisateur soit connecté.  Interface Modification patient affichée  Le patient doit exister |  |
| Résultats attendus et critères | Le serveur est lancé la page index est affichée | Cf. ci-dessous |
| **Procédure de test** | | |
| **Numéro d’étape** | **Action de l’opérateur** | **Le patient est bien géré dans la base de donnée (vérification en analysant les logs du serveur Fhir)** |
| 1 | Saisir dans le champ id l’id du patient | Dans le champ nom il y a l’id du patient |
| 2 | Appuyer sur le bouton Rechercher | Le bouton à bien été cliqué et la page change. |
| 3 | Vérifier que l’on a bien toutes les informations de ce patient | Vérifier que l’id est le même |

## PLANIFICATION DES TESTS

## PHASE DE TEST 1

## Couverture du test

Les tests de la phase 1 couvrent les tests logiciels.

## Planification des tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifiant du test** | **Description** | **Méthode** | **Catégorie** |
|  |  |  | Principale |

## PHASE DE TEST 2

## Couverture du test

Les tests de la phase 2 couvrent les tests logiciels.

## Planification des tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifiant du test** | **Description** | **Méthode** | **Catégorie** |
| T-MAIN-001 |  |  |  |