



H.E.B. ECOLE SUPERIEUR D'INFORMATIQUE

LABORATOIRE D'ANALYSE : PROJET D'ANALYSE

PLAN DE TESTS FONCTIONNELS ÉLÉMENTAIRES - UC
1104 - DÉFINIR UN RENDEZ-VOUS

Medicagenda

Auteurs :

Florian KNOP
39310@heb.be
Paul KRIWIN
39171@heb.be
Simon PLACENTINO
39631@heb.be

Titulaire du cours :

Mr. Nicolas PETTIAUX
npettiaux@heb.be

2 mai 2015

Table des matières

1	Plan de test fonctionnels élémentaires U.C. 1104	2
1.1	Introduction	2
1.1.1	Objectifs du document	2
1.1.2	Domaine de définition du document	2
1.1.3	Définitions, acronymes et abréviations	2
1.1.4	Références	2
1.2	Types de tests élémentaires fonctionnels	3
1.2.1	Validation des règles de saisie	3
1.2.2	Validation des règles de calcul	3
1.2.3	Validation des mises à jour du SI	3
1.2.4	Validation des outputs utilisateurs	3
1.2.5	Tests de consolidation	3
1.2.6	Tests de saisie	3
1.2.7	Cas extrêmes	3
1.3	Scénarios de tests	4
1.4	Valeurs	6
1.4.1	Valeurs d'initialisation	6
1.4.2	Valeurs spécifiques	6

Chapitre 1

Plan de test fonctionnels élémentaires U.C. 1104

1.1 Introduction

1.1.1 Objectifs du document

du document Ce document présente le plan des tests fonctionnels du U.C. 1104 - définir un rendez-vous.

1.1.2 Domaine de définition du document

Ce document ne présente que les scénarios correspondant à la définition d'un rendez-vous (correspondant au diagramme d'activités de la description fonctionnelle du UC).

1.1.3 Définitions, acronymes et abréviations

Définition INAMI : voir étude de cas et MCD¹.

1.1.4 Références

Étude de cas de l'agenda médical²

M.C.D. de **Medicagenda**³

M.C.T. de **Medicagenda**⁴

Spécifications fonctionnelles du U.C. 1104⁵

-
1. voir document M.C.D.
 2. voir étude de cas
 3. voir document M.C.D.
 4. voir document M.C.T.
 5. Voir spécifications fonctionnelles du U.C. 1104

Remarque dans le plan de test, il est prévu de faire appel aux UC 1101a/b – enregistrer un compte patient/médecin ainsi qu’à l’U.C. 1103 - définir une disponibilité. Les U.C. 1101a/b et 1103 devront donc être réalisés et testés avant le présent U.C. 1104.

1.2 Types de tests élémentaires fonctionnels

1.2.1 Validation des règles de saisie

UC interactif Validation des formats et valeurs des champs : voir description de l’interface et des messages d’erreur liés à son utilisation.

La date et heure du rendez-vous à réserver ne doivent pas être antérieurs au moment où a lieu la réservation.

1.2.2 Validation des règles de calcul

S.O.

1.2.3 Validation des mises à jour du SI

Vérification de la création correcte des nouveaux enregistrements dans la base de données, pour les différents scénarios décrits ci-dessous, ainsi que des liens entre objets de classes différentes. Cela implique que le nouveau rendez-vous créé concerne les bons comptes utilisateurs, est à la date et heure convenue et que la plage horaire occupée par celui-ci soit dorénavant indisponible.

1.2.4 Validation des outputs utilisateurs

Vérification que les messages d’erreurs et de validations apparaissent pertinemment et s’affichent correctement. Les messages de récapitulation de la réservation doivent être pertinents pour les deux types d’utilisateurs (médecin et utilisateur) et exempts de toutes erreurs.

1.2.5 Tests de consolidation

1.2.6 Tests de saisie

1.2.7 Cas extrêmes

Au cours de différents scénarios, tests de l’abandon du U.C. et vérification que la base de données est effectivement restée intacte.

1.3 Scénarios de tests

Scénario | cas | description | Résultat attendu

scénario 0. cas 1.

– description :

Le patient possède déjà un compte, a sélectionné un médecin ou une date ou une plage horaire mais désire annuler son choix pour en sélectionner un/une autre.

– résultat :

La base de donnée n'est jamais modifiée, et la fenêtre de l'étape précédente est réaffichée.

scénario 1. cas 1.1

– description :

Le patient possède déjà un compte, a sélectionné et validé un médecin et une date et heure disponible.

– résultat :

un fenêtre affiche le calendrier et la barre de recherche pour entrer le nom du medecin. Une fois ce dernier et une date valide sélectionnés, est affichée une grille horaire de cette date avec les disponibilités du médecin. Après que la plage horaire eu été sélectionnée, le message récapitulatif de la réservation s'est affiché et a été validé. Le nouveau rendez-vous est enregistré, il s'affiche dans le calendrier du patient et dans celui du médecin, la disponibilité correspondant à la plage horaire du nouveau rendez-vous n'est plus.

scénario 1. cas 1.2

– description :

idem que 1.1 mais le patient n'est pas enregistré

– résultat :

idem que 1.1 mais le patient s' enregistre au préalable (appel au U.C. 1101)

scénario 2. cas 2.1

– description :

Le patient sélectionne un médecin et essaye de sélectionner une date ne contenant pas de disponibilité.

– résultat :

Statu quo, les dates concernées doivent être non cliquable et rien ne doit alors se passer.

scénario 2. cas 2.2

– description :

Le patient sélectionne un médecin et essaye de sélectionner une date

antérieure au moment de la réservation.

– résultat :

Idem que 2.1.

scénario 2. cas 2.3

– description :

Le patient sélectionne un médecin et essaye de sélectionner une date à laquelle le médecin possède des disponibilités, mais une plage horaire à laquelle il possède déjà un rendez-vous.

– résultat :

statu quo, les cases représentant ces disponibilités doivent être rougie avec un message décrivant le fait que le patient a déjà un rendez vous à ce moment là et donc, doivent être non cliquable et rien ne doit alors se passer.

1.4 Valeurs

1.4.1 Valeurs d’initialisation

Pour pouvoir faire les tests des scénarios 1 et 2, il faudra disposer d’une base de données contenant un nombre minimal de comtes médecins avec disponibilités, ainsi que de quelques comptes patients. Il faudra également disposer du U.C. 1101 testé qui peut être appelé par le présent U.C. 1104.

1.4.2 Valeurs spécifiques

Avant les tests, il faut constituer la base de données de tests et des cas de tests avec des disponibilités et rendez-vous déjà présents dans les calendrier des médecins et patients (ne pas oublier que l’environnement de tests (comme l’environnement de développement) doit être séparé de l’environnement de production). Les données spécifiques se retrouveront dans la base de données qui sera mise à la disposition des testeurs. Pour “peupler” cette base de données, il sera fait appel à des outils d’encodage rapide de données.