



Université
de Paris

Projets tutorés
2020-2021

Projet L3AX1 - Facturation

Plan de test

Atia Salah Eddine Bouldja Lounis Namolaru Leonard Saghroun Amos
18 avril 2021

Cahier de recette - projet L3AX1

Les informations d'identification du document

Référence du document : D7

Version du document : 1.01

Date du document : 18/04/21

Auteurs : Atia Salah Eddine Bouldja Lounis
Namolaru Leonard Saghroun Amos

Les éléments de vérification du document

Soumis le : 18/04/21

Type de diffusion : Document électronique (.pdf)

Confidentialité : Réservé aux étudiants
UFR Maths-Info de l'université de Paris

Les éléments d'authentification

Maître d'ouvrage : M. Alaa Dandan

Chef de projet : Atia Salah Eddine

Mots clés

Plan de test, Application Web, Facture

Sommaire

Sommaire	2
1. Introduction	3
1.1. Objectifs et méthodes	3
1.2. Documents de référence	3
2. Guide de lecture	4
2.1. Maîtrise d'œuvre	4
2.2. Maîtrise d'ouvrage	4
3. Concepts de base	5
4. Tests fonctionnels	6
4.1. Introduction	6
4.2. Connexion au panneau d'administration	7
4.3. Gestion des paramètres du système	8
4.4. Panneau de gestion des clients	9
4.5. Panneau de gestion des détails de facture	12
4.6. Panneau de gestion des contrats d'un client	13
4.7. Panneau de gestion des factures d'un client	15
4.8. Suivi des paiements	16
4.9. Gestion des utilisateurs	17
5. Conformité aux spécifications d'interfaces	19
6. Tests unitaires	22
7. Tests d'intégrations	27
8. Glossaire	36
9. Sources	38
10. Index	39

1. Introduction

1.1. Objectifs et méthodes

L'objectif principal du plan de tests est de démontrer l'ordre dans lequel chaque composant de notre système sera complété, testé individuellement, et intégré avec les autres composants du système. Il est nécessaire de détecter le plus rapidement possible les problèmes logiciels de manière à pouvoir les corriger à moindre coût. De plus, les tests sont également réalisés pour développer la confiance et la connaissance de notre produit livré. C'est ainsi tester un logiciel est le processus utilisé pour vérifier et valider un programme, une application ou un produit : - Répond aux exigences techniques et professionnelles telles que décrites dans le cahier des charges, les exigences, les documents d'analyse et de conception. - Fonctionne comme prévu et est utilisable dans son environnement opérationnel. - Est implémenté avec les caractéristiques logicielles non-fonctionnelles présumées. Chaque membre de l'équipe devra être impliqué dans le processus des tests. La planification des tests, ainsi que le suivi et le contrôle des tests font partie de notre projet. Nous allons établir les processus et mécanismes de livraison des objectifs de tests, qui seront intégrés dans le plan de tests. Pour terminer, nous nous intéresserons dans un premier temps aux tests fonctionnels de notre application web, puis au test d'intégration pour vérifier les différents modules, paquetages, etc... Et enfin, nous expliquerons ensuite les tests unitaires afin de vérifier le comportement de chaque fonction.

1.2. Documents de référence

- Cahier de charge.
- Cahier de recettes.

2. Guide de lecture

Ce document s'appuiera grandement sur le cahier des charges et le cahier de recette du projet.

2.1. Maîtrise d'œuvre

Le maître d'œuvre (ou maîtrise d'œuvre) intervient au niveau de la réalisation du projet. Il s'agit souvent de la direction technique, du chef de projet ou encore du développeur directement dans le cas de projets plus petits et elle est représentée par (développeurs) :

• Lounis BOULDJA • Atia Salah Eddine • Namolaru Leonard • Saghroun amos

2.2. Maîtrise d'ouvrage

Les délais, la documentation, le code source ainsi que le site web doivent être fournis le 18 avril 2020. Afin de s'assurer du bon fonctionnement, le site créé devra pouvoir être responsive c'est-à-dire visible et ergonomique aussi bien depuis un téléphone que depuis un ordinateur.

3. Concepts de base

Le plan de tests se compose de 3 éléments principaux :

- ❖ Tests unitaires par composant : les tests permettant de vérifier le comportement de chaque fonction, méthodes, etc... de l'application correspond de manière satisfaisante à ce qui a été défini lors de la phase de conception détaillée.
- ❖ Tests d'intégration des divers composants : les tests permettant de vérifier que les différents modules, paquetages, etc... de l'application s'interface correctement. On distingue les interfaces internes des interfaces externes de notre application

- ❖ Tests fonctionnels : les scénarios permettant de vérifier que l'application recouvre bien le périmètre fonctionnel qui a été défini lors de la phase de spécification. Le périmètre fonctionnel est défini dans le cahier des charges

A ce niveau, pour tester tous ces éléments il faut aussi mentionner l'approche globale du test. Il faut spécifier les niveaux de test, les types de test et les méthodes, en fonction des objectifs. Il faut aussi définir les critères utilisés afin d'établir si chaque élément du produit a réussi ou échoué lors des tests

4. Tests fonctionnels

4.1. Introduction

Cette partie du document décrit les scénarios permettant de vérifier que l'application fournie recouvre bien le périmètre fonctionnel qui a été défini lors de la phase de spécification. Le périmètre fonctionnel est défini dans le cahier des charges.

Chacun des scénarios comprend les éléments suivants :

- ❖ **Identification** - Chaque scénario a un identifiant unique.
- ❖ **Description** - Description du but et le principe de réalisation du test ainsi que l'environnement de test.
- ❖ **Contraintes** - Description des contraintes liées au scénario : environnement de test particulier, installation particulière, intervention humaine spécifique, etc.
- ❖ **Dépendances** - Description des tests à mener préalablement à la réalisation du scénario.
- ❖ **Procédure de test** - Description des données en entrée, les résultats attendus, et les critères de validation.

4.2. Connexion au panneau d'administration

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
A1	Se connecter à son compte Indispensable	Tester si un utilisateur existant dans la base de données est capable de se connecter avec son email et son mot de passe.	Saisie d'un e-mail et d'un mot de passe dans les champs appropriés de la page de connexion du panneau d'administration	Un utilisateur avec un tel email et mot de passe doit être présent dans la base de données (le premier utilisateur sera ajouté à l'aide d'une requête SQL)	L'utilisateur utilise les champs appropriés pour saisir son email et son mot de passe. Après avoir saisi les données, l'utilisateur clique sur le bouton "Continuer". Si la connexion est établie avec succès, l'utilisateur est transféré vers le panneau d'administration.
A2	"J'ai oublié mon mot de passe" Indispensable	Tester si un utilisateur existant dans la base de données peut créer un nouveau mot de passe s'il oublie son mot de passe actuel.	Cliquer sur le lien "Mot de passe oublié?", Entrer un email dans le champ approprié sur la page "Réinitialisation du mot de passe" et cliquer sur le bouton "Continuer"	Un utilisateur avec un tel email doit être présent dans la base de données (le premier utilisateur sera ajouté à l'aide d'une requête SQL)	Cliquer sur le lien de réinitialisation du mot de passe envoyé à la boîte e-mail et remplir les champs «Nouveau mot de passe» et «Confirmer le mot de passe» sur la page Web qui s'ouvre, cliquer sur le bouton «Continuer». Si la réinitialisation du mot de passe a réussi, le test A1 peut maintenant être effectué avec succès à l'aide du nouveau mot de passe.

A3	"Se souvenir de moi" Optionnelle	Vérifier si la session de l'utilisateur reste ouverte si la case "Se souvenir de moi" est cochée.	Saisie d'un e-mail et d'un mot de passe dans les champs appropriés de la page de connexion du panneau d'administration, cocher la case "Se souvenir de moi"	Le test A1	L'utilisateur utilise les champs appropriés pour saisir son email et son mot de passe. Après avoir saisi les données, l'utilisateur clique sur le bouton "Continuer". Si la connexion est établie avec succès, l'utilisateur est transféré vers le panneau d'administration. <p>Même après avoir fermé le navigateur et redémarré l'ordinateur, la session doit toujours être ouverte dans ce navigateur.</p>
----	--	---	---	------------	---

4.3. Gestion des paramètres du système

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
B1	Accès à la page de gestion des paramètres Indispensable	Vérifier si un utilisateur qui s'est connecté avec succès au panneau d'administration peut accéder à la page de gestion des paramètres.	Cliquer sur l'onglet "Paramètres" dans le panneau d'administration	Le test A1	Si tout va bien, l'utilisateur est maintenant dirigé vers la page des paramètres système, à travers laquelle il peut saisir le nom de l'entreprise, le logo de l'entreprise, le numéro SIRET de l'entreprise, les coordonnées bancaires de l'entreprise, etc.

B2	Définir les paramètres Indispensable	Vérifier si les différents paramètres peuvent être définis à l'aide du panneau d'administration.	Cliquer sur l'onglet "Paramètres" dans le panneau d'administration	Le test A1 Le test B1	Saisir des valeurs dans les différents champs et cliquer sur le bouton "Enregistrer". Les paramètres doivent être stockés dans un fichier dédié (et non dans la base de données).
----	--	--	--	------------------------------	--

4.4. Panneau de gestion des clients

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
C1	Ajouter un nouveau client Indispensable	Vérifier si un nouveau client peut être ajouté avec succès à la base de données à l'aide du panneau d'administration	Cliquer sur l'onglet "Clients" dans le panneau d'administration puis sur le bouton "Ajouter un nouveau client"	Le test A1	Après avoir cliqué sur le bouton «ajouter un nouveau client», une fenêtre «pop-up» doit s'ouvrir dans laquelle les informations client doivent être saisies. Si le client a été ajouté avec succès, on peut voir que le client a été ajouté à la liste des clients sous l'onglet "Clients"

C2	<p>page de données sur un client</p> <p>Indispensable</p>	<p>Après avoir ajouté un nouveau client, est-il possible d'accéder à la page de données concernant ce client</p>	<p>Cliquer sur l'onglet "Clients" dans le panneau d'administration puis sur le nom du client</p>	<p>Le test A1</p> <p>Le test C1</p>	<p>Cliquer sur le nom de l'un des clients de la liste des clients. Si tout va bien, nous devrions être redirigés vers une page qui montre toutes les données concernant le même client: contrats, factures, données personnelles, etc.</p>
C3	<p>Modifier les informations sur le compte client</p> <p>Indispensable</p>	<p>Lorsque nous arrivons à la page qui comprend toutes les données concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation, informations sur la livraison et informations sur la TVA. Chacune de ces catégories peut être modifiée séparément.</p> <p>Ce test testera l'édition de la section Informations sur le compte.</p>	<p>cliquer sur le bouton "modifier" en haut à droite du bloc "Informations sur le compte"</p>	<p>Le test A1</p> <p>Le test C1</p> <p>Le test C2</p>	<p>Le bouton "Modifier" doit ouvrir une fenêtre "pop-up" dans laquelle il est possible de modifier le nom, l'adresse e-mail ou la description du client. Après avoir effectué les modifications, cliquer sur le bouton "modifier" en bas à droite de la fenêtre "pop-up". Si tout se passe bien, les données de la page de ce client ont changé en fonction des modifications apportées.</p>
C4	<p>Modifier les informations sur la facturation</p> <p>Indispensable</p>	<p>Lorsque nous arrivons à la page qui comprend toutes les données concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation, informations sur la livraison et informations sur la TVA. Chacune de ces catégories peut être modifiée séparément.</p> <p>Ce test testera l'édition de la section informations sur la facturation.</p>	<p>cliquer sur le bouton "modifier" en haut à droite du bloc "informations sur la facturation"</p>	<p>Le test A1</p> <p>Le test C1</p> <p>Le test C2</p>	<p>Le bouton "Modifier" doit ouvrir une fenêtre "pop-up" dans laquelle il est possible de modifier le pays, l'adresse, le code postal ou la ville. Après avoir effectué les modifications, cliquer sur le bouton "modifier" en bas à droite de la fenêtre "pop-up". Si tout se passe bien, les données de la page de ce client ont changé en fonction des modifications apportées.</p>

C5	Modifier les informations sur la livraison Indispensable	Lorsque nous arrivons à la page qui comprend toutes les données concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation, informations sur la livraison et informations sur la TVA. Chacune de ces catégories peut être modifiée séparément. Ce test testera l'édition de la section informations sur la livraison.	cliquer sur le bouton "modifier" en haut à droite du bloc "informations sur la livraison"	Le test A1 Le test C1 Le test C2	Le bouton "Modifier" doit ouvrir une fenêtre "pop-up" dans laquelle il est possible de modifier le pays, l'adresse, le code postal ou la ville. Après avoir effectué les modifications, cliquer sur le bouton "modifier" en bas à droite de la fenêtre "pop-up". Si tout se passe bien, les données de la page de ce client ont changé en fonction des modifications apportées.
C6	Modifier les informations sur la TVA Indispensable	Lorsque nous arrivons à la page qui comprend toutes les données concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation, informations sur la livraison et informations sur la TVA. Chacune de ces catégories peut être modifiée séparément. Ce test testera l'édition de la section informations sur la TVA.	cliquer sur le bouton "modifier" en haut à droite du bloc "informations sur la TVA"	Le test A1 Le test C1 Le test C2	Le bouton "Modifier" doit ouvrir une fenêtre "pop-up" dans laquelle il est possible de modifier la localisation et le numéro TVA. Après avoir effectué les modifications, cliquer sur le bouton "modifier" en bas à droite de la fenêtre "pop-up". Si tout se passe bien, les données de la page de ce client ont changé en fonction des modifications apportées.
C7	Recherche de clients Optionnelle	Sous l'onglet "Clients", il y a un moteur de recherche en haut de la page pour rechercher un client qui existe dans la base de données du système. Le but de ce test est de vérifier que grâce à ce moteur de recherche, on peut bien	Cliquer sur l'onglet "Clients" dans le panneau d'administration ,entrer le nom d'un client qui existe dans la base de	Le test A1 Le test C1	Après avoir cliqué sur la loupe, les résultats de la recherche seront affichés. Si l'utilisateur concerné apparaît dans les résultats de la recherche, le test a réussi.

		trouver un client figurant sur la liste des clients.	données dans le champ de recherche et cliquer sur la loupe à gauche		
--	--	--	---	--	--

4.5. Panneau de gestion des détails de facture

Détails de facture : produits, services, etc.

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
D1	Ajouter un nouveau détail de facture Indispensable	Vérifier si un nouveau détail de facture peut être ajouté avec succès à la base de données à l'aide du panneau d'administration	Cliquer sur l'onglet "Détails de facture" dans le panneau d'administration puis sur le bouton "Ajouter un nouveau détail de facture"	Le test A1	Après avoir cliqué sur le bouton «Ajouter un nouveau détail de facture», une fenêtre «pop-up» doit s'ouvrir dans laquelle des informations doivent être saisies : Type (produit / service), Nom, etc. Si la détail de facture a été ajouté avec succès, on peut voir que la détail de facture a été ajouté à la liste des détails de facture sous l'onglet "Détails de facture"

D2	Supprimer Un nouveau détail de facture Indispensable	vérifier si un détail de facture peut être supprimé de la base de données à l'aide du panneau d'administration	Cliquer sur l'onglet «Détails de la facture», parcourir la liste des détails de facture disponible dans la base de données. Pour supprimer un détail de facture spécifique, cliquer sur les trois points à la fin de la ligne, Cela ouvrira un petit menu avec l'option: "Supprimer".	Le test D1	Une boîte de dialogue s'ouvre dans laquelle l'utilisateur doit répondre s'il est sûr de vouloir supprimer. Si la réponse est oui, la suppression sera effectuée à condition que ce détail de facture n'ait encore été utilisé dans aucune facture.
----	---	--	---	------------	--

4.6. Panneau de gestion des contrats d'un client

Nous avons décidé de ne pas utiliser d'onglet séparé pour les contrats (cependant, il existe des onglets pour «Clients», «Détails de la facture», etc.). Les factures seront gérées séparément pour chaque client, c'est-à-dire que sur la page spécifique qui existe pour chaque client dans le système, il y a un tableau qui résume tous les contrats relatifs à ce client. À partir de ce tableau, vous pouvez effectuer des actions telles que l'ajout d'un nouveau contrat.

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
E1	Ajouter un nouveau contrat Indispensable	Vérifier si un nouveau contrat peut être ajouté avec succès à la base de données à l'aide du panneau d'administration	<p>Cliquer sur l'onglet "Clients" dans le panneau d'administration puis sur le nom du client.</p> <p>Lorsque nous arrivons à la page qui comprend toutes les données concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation... Il existe également un tableau répertoriant tous les contrats signés avec ce client.</p>	<p>Le test A1</p> <p>Le test B1</p> <p>Le test B2</p> <p>Le test C1</p> <p>Le test D1</p>	Après avoir cliqué sur le bouton "Ajouter un nouveau contrat", une fenêtre «pop-up» doit s'ouvrir dans laquelle des informations doivent être saisies : date de signature du contrat, durée du contrat, produits et services inclus dans le contrat, etc. Si le contrat a été ajouté avec succès, on peut voir que la contrat a été ajouté à la liste des contrats sur la page qui comprend toutes les données concernant le client.
E2	Supprimer Un contrat Indispensable	vérifier si un contrat peut être supprimé de la base de données à l'aide du panneau d'administration	<p>Cliquer sur l'onglet "Clients" dans le panneau d'administration puis sur le nom du client.</p> <p>Pour supprimer un contrat spécifique,</p>	<p>Le test A1</p> <p>Le test B1</p> <p>Le test B2</p> <p>Le test C1</p> <p>Le test D1</p>	Une boîte de dialogue s'ouvre dans laquelle l'utilisateur doit répondre s'il est sûr de vouloir supprimer. Si la réponse est oui, la suppression sera effectuée à condition que ce

			cliquer sur les trois points à la fin de la ligne, Cela ouvrira un petit menu avec l'option: "Supprimer".	Le test E1	contrat n'ait encore été utilisé.
--	--	--	---	------------	-----------------------------------

4.7. Panneau de gestion des factures d'un client

Nous avons décidé de ne pas utiliser d'onglet séparé pour les factures (cependant, il existe des onglets pour «Clients», «Détails de la facture», etc.). Les factures seront gérées séparément pour chaque client, c'est-à-dire que sur la page spécifique qui existe pour chaque client dans le système, il y a un tableau qui résume tous les facteurs relatifs à ce client. À partir de ce tableau, vous pouvez effectuer des actions telles que l'ajout d'une facture.

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
F1	Création d'une nouvelle facture Indispensable	Vérifier si un utilisateur qui s'est connecté avec succès au panneau d'administration peut accéder à l'interface de création d'une nouvelle facture.	<p>Cliquer sur l'onglet "Clients" dans le panneau d'administration puis sur le nom du client.</p> <p>Lorsque nous arrivons à la page qui comprend toutes les données</p>	<p>Le test A1</p> <p>Le test B1</p> <p>Le test B2</p> <p>Le test C1</p> <p>Le test D1</p> <p>Le test E1</p>	Après avoir cliqué sur le bouton "Créer", une fenêtre «pop-up» doit s'ouvrir dans laquelle des informations doivent être saisies.. Si la facture a été ajoutée avec succès, on peut voir que la facture a été ajoutée à la liste des factures sur la page qui comprend toutes

			concernant un client particulier, les données sont divisées en plusieurs catégories: informations sur le compte, informations sur la facturation.. Il existe également un tableau répertoriant toutes les factures.		les données concernant le client.
--	--	--	---	--	-----------------------------------

4.8. Suivi des paiements

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
G1	Accès à la page de suivi des paiements Indispensable	Vérifier si un utilisateur qui s'est connecté avec succès au panneau d'administration peut accéder à la page de suivi des paiements.	Cliquer sur l'onglet "Paiements" dans le panneau d'administration	Le test A1	Si tout va bien, l'utilisateur est maintenant dirigé vers la page de suivi des paiements.
G2	Recherche de clients Optionnelle	Sous l'onglet "Paiements", il y a un moteur de recherche en haut de la page pour rechercher un client qui existe dans la base de données du système. Le but de ce test est de vérifier que	Cliquer sur l'onglet "Paiements" dans le panneau d'administration ,entrer le nom	Le test A1 Le test G1	Après avoir cliqué sur la loupe, les résultats de la recherche seront affichés. Si l'utilisateur concerné apparaît dans les résultats de la recherche, le test a réussi.

		grâce à ce moteur de recherche, on peut bien trouver un client figurant sur la liste des clients.	d'un client qui existe dans la base de données dans le champ de recherche et cliquer sur la loupe à gauche		
--	--	---	--	--	--

4.9. Gestion des utilisateurs

Gérer les utilisateurs qui ont accès au panneau d'administration.

L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.

ID	Nom de la fonction	Description	Contraintes	Dépendances	Procédure de test
H1	Accès à la page de Gestion des utilisateurs Indispensable	Vérifier si un utilisateur qui s'est connecté avec succès au panneau d'administration peut accéder à la page de gestion des utilisateurs	Cliquer sur l'onglet "Utilisateurs" dans le panneau d'administration	Le test A1	Si tout va bien, l'utilisateur est maintenant dirigé vers la page de gestion des utilisateurs.
H2	Ajouter un nouvel utilisateur Indispensable	Vérifier si un nouvel utilisateur peut être ajouté avec succès à la base de données à l'aide du panneau d'administration	Cliquer sur l'onglet "Utilisateurs" dans le panneau d'administration puis sur le	Le test A1 Le test H1	Après avoir cliqué sur le bouton « Ajouter un nouvel utilisateur », une fenêtre «pop-up» doit s'ouvrir dans laquelle les informations sur l'utilisateur doivent être

			bouton "Ajouter un nouvel utilisateur"		saisies. Si l'utilisateur a été ajouté avec succès, on peut voir que l'utilisateur a été ajouté à la liste des utilisateurs sous l'onglet "Utilisateurs"
H3	Page de données sur un utilisateur Indispensable	Après avoir ajouté un nouvel utilisateur, est-il possible d'accéder à la page de données concernant l'utilisateur ?	Cliquer sur l'onglet "Utilisateurs" dans le panneau d'administration puis sur le nom de l'utilisateur	Le test A1 Le test H1 Le test H2	Cliquer sur le nom de l'un des utilisateurs de la liste des utilisateurs. Si tout va bien, nous devrions être redirigés vers une page qui montre toutes les données concernant le même utilisateur.

5. Conformité aux spécifications d’interfaces

Les tests permettant la vérification des interfaces de l’application

Le but de ces tests est de s'assurer que notre application Web fonctionne correctement sous différents systèmes d'exploitation et différents navigateurs. De plus, des tests seront effectués sur des écrans de différents types (ordinateur portable, tablette, smartphone Android, iPhone).

ID	Description	Contraintes	Dépendances	Procédure de test
IN1	L'objectif est de s'assurer que l'application Web fonctionne	L'environnement de test est un navigateur Web acheminé vers	Posséder un ordinateur sur lequel le système d'exploitation	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.



	correctement sous le système d'exploitation Windows (Microsoft)	un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Windows de Microsoft est installé	
IN2	L'objectif est de s'assurer que l'application Web fonctionne correctement sous un système d'exploitation Linux	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Posséder un ordinateur sur lequel le système d'exploitation Linux est installé	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.
IN3	L'objectif est de s'assurer que l'application Web fonctionne correctement sous le système d'exploitation MacOS	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Posséder un ordinateur sur lequel le système d'exploitation MacOS est installé	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.
IN4	L'objectif est de s'assurer que l'application Web fonctionne correctement sous le	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel	Posséder un ordinateur sur lequel le navigateur Google Chrome est installé	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.

	navigateur Google Chrome	l'application Web est stockée.		
IN5	L'objectif est de s'assurer que l'application Web fonctionne correctement sous le navigateur Mozilla Firefox	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Posséder un ordinateur sur lequel le navigateur Mozilla Firefox est installé	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.
IN6	L'objectif est de s'assurer que l'application Web fonctionne correctement sous une tablette.	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Posséder une tablette.	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.
IN7	L'objectif est de s'assurer que l'application Web fonctionne correctement sous un smartphone Android.	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application	Posséder un smartphone Android.	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.

		Web est stockée.		
IN8	L'objectif est de s'assurer que l'application Web fonctionne correctement sous un iPhone.	L'environnement de test est un navigateur Web acheminé vers un serveur local ou un serveur Web sur lequel l'application Web est stockée.	Posséder un iPhone.	Vérifier que l'affichage graphique est correct et que toutes les fonctions fonctionnent correctement.

6. Tests unitaires

L'ensemble des tests unitaires sont testés par un FRAMEWORK : JUnit. JUnit est une bibliothèque JAVA facilitant l'écriture de tests unitaires. Elles fournissent les méthodes pour tester que certaines conditions sont satisfaites, et signaler une erreur dans le cas contraire. Tous les tests se retrouvent dans le fichier : tests de notre projet. Les tests sont séparés par des packages en fonction du type testé.

Rappelons que les tests unitaires ont pour vocation à tester uniquement le contenu d'une méthode, alors que les tests d'intégration impliquent de tester plus largement une fonctionnalité.

Voici un exemple de test unitaire

```

@WebMvcTest(controllers = Controller.class)
public class ControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @MockBean
    private Service service;

    // Exemple d'un test unitaire.
    @Test
    public void testUnit () throws Exception {
        mockMvc.perform(get("/path"))
                .andExpect(status().isOk());
    }

    ...

}

```

Explications :

- **@WebMvcTest**(controllers = Controller.class) déclenche le mécanisme permettant de tester les controllers. On indique également le ou les contrôleurs concernés.
- L'attribut mockMvc est un autre élément important. Il permet d'appeler la méthode "perform" qui déclenche la requête.
- L'attribut Service est annoté **@MockBean**. Il est obligatoire, car la méthode du controller exécutée par l'appel de "/path" utilise cette classe.
- La méthode perform prend en paramètre l'instruction get("/path"). On exécute donc une requête GET sur l'URL /path.
- Ensuite, l'instruction .andExpect(status().isOk()) indique que nous attendons une réponse HTTP 200.
- **@Test** : annotation de JUnit, indique que la méthode est d'un test.

Test - Client

ID	Nom de la fonction	Comportement attendu	Statut
U01	<code>public void contextLoads</code>	<i>Tester si le contrôleur n'est pas null</i>	réussite
U02	<code>public void testUnitGetClients() throws Exception</code>	<i>charger les clients de la base de données. nous attendons une réponse HTTP 200.</i>	réussite
U03	<code>public void testUnitGetClientById() throws Exception</code>	<i>charger un client par de son id de la base de données. nous attendons une réponse HTTP 200.</i>	réussite
U04	<code>public void testUnitGetContratById()</code>	<i>charger les contrats d'un client de la base de données. nous attendons une réponse HTTP 200.</i>	réussite
U05	<code>public void testUnitCreateClient() throws Exception</code>	<i>Créer un client (physique ou morale). nous attendons une réponse HTTP 200.</i>	réussite
U06	<code>public void testUnitUpdateClient() throws Exception</code>	<i>Modifier un client (physique ou morale). nous attendons une réponse HTTP 200.</i>	réussite

Test - Contrat

U07	<code>contextLoads()</code>	<i>Tester si le contrôleur n'est pas null</i>	réussite
U08	<code>public void testUnitGetContrats() throws Exception</code>	<i>Retourner la liste de contrats nous attendons une réponse HTTP 200.</i>	réussite
U09	<code>public void testUnitPostCreateContrats() throws Exception</code>	<i>créer un nouveau contrat nous attendons une réponse HTTP 200.</i>	réussite

Test - Produit

U11	<code>public void contextLoads()</code>	<i>Tester si le contrôleur n'est pas null</i>	réussite
U12	<code>public void testUnitGetProduits() throws Exception</code>	<i>Récupérer les contrat. nous attendons une réponse HTTP 200.</i>	réussite

U13	<pre>public void testUnitPostCreatePro duit() throws Exception</pre>	<i>créer un nouveau produit nous attendons une réponse HTTP 200.</i>	réussite
-----	--	---	-----------------

Test - Facture

U14	<pre>public void contextLoads()</pre>	<i>Tester si le contrôleur n'est pas null</i>	réussite
U15	<pre>public void testUnitGetContrats() throws Exception</pre>	<i>Récupérer les contrats de la base de données. nous attendons une réponse HTTP 200.</i>	réussite

7. Tests d'intégrations

les tests d'intégrations est une phase de tests, précédée par les tests unitaires, vérifiant le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée «unité» ou « module ») ; dans le test d'intégration, chacun des modules indépendants du logiciel est assemblé et testé dans l'ensemble.

Pour écrire des tests d'intégration, on peut utiliser l'annotation `@SpringBootTest`
Voici un exemple de test d'intégration.

```

@SpringBootTest
@AutoConfigureMockMvc
public class ControllerTest {

    @Autowired
    private MockMvc mockMvc;

    // Exemple d'un test d'intégration.
    @Test
    public void test() throws Exception {
        mockMvc.perform(get("/path"))
            .andExpect(status().isOk())
            .andExpect(jsonPath("$.attribut", is(result)));
    }

    . . .

}

```

Explications :

- Les annotations **@SpringBootTest** et **@AutoConfigureMockMvc** permettent de charger le contexte Spring et de réaliser des requêtes sur le controller.
- L'attribut annoté **@MockBean** a disparu, plus besoin d'un mock pour Service, car ce dernier a été injecté grâce à **@SpringBootTest**.
- En plus de vérifier si le statut vaut 200, on vérifie le contenu retourné grâce à `jsonPath("$.attribut", is(result))`.
 - \$ pointe sur la racine de la structure JSON.
 - [0] indique qu'on veut vérifier le premier élément de la liste.
 - attribut désigne l'attribut qu'on veut consulter.
- `is(result)` est ce que l'on attend comme résultat.

Client - Test

ID	Nom de la fonction	Comportement attendu	Statut
i01	<code>getClients() throws Exception</code>	<i>requête HTTP <GET></i> <i>pour récupérer un client de la base de données.</i> <i>nous attendons une réponse HTTP 200</i> <i>+</i> <i>récupérer les client de la base de données</i>	réussite
i02	<code>getClientById() throws Exception</code>	<i>Test la requête HTTP GET pour récupérer un client de la base de données.</i> <i>nous attendons une réponse HTTP 200 + client retourné grâce à jsonPath = client de la base de données.</i>	réussite
i03	<code>createMoraleClient(@RequestBody Morale morale)</code> <code>&</code> <code>creatPhysiqueClient(@RequestBody Physique physique)</code>	<i>requête HTTP <POST></i> <i>nous attendons une réponse HTTP 200</i> <i>+</i> <i>client est ajouté dans la base de données.</i>	réussite

i04	<pre>updatePhysiqueClient(@ PathVariable Long id, @RequestBody Physique clientDetails) & updateMoraleClient(@Pa thVariable Long id, @RequestBody Morale clientDetails)</pre>	<p><i>HTTP < PUT></i></p> <p><i>nous attendons une réponse HTTP 200 .</i></p> <p><i>+</i></p> <p><i>client modifié dans la base de données.</i></p>	réussite
i05	<pre>getClientContratsById(@PathVariable Long id)</pre>	<p><i>requête HTTP < GET></i></p> <p><i>nous attendons une réponse HTTP 200 .</i></p> <p><i>+</i></p> <p><i>les contrats d'un client</i></p>	réussite
i06	<pre>creatPhysiqueClientWit hContrat(@PathVariable Long id, @RequestBody Physique physique) & createMoraleClientWith Contrat(@PathVariable Long id, @RequestBody Morale morale)</pre>	<p><i>requête HTTP < POST></i></p> <p><i>nous attendons une réponse HTTP 200 .</i></p> <p><i>+</i></p> <p><i>création d'un client avec un contrat.</i></p>	réussite

Contrat - Test

i07	<code>getContrats ()</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> <i>+</i> <i>retourner tous les contrats de la base de données.</i>	réussite
i08	<code>Contrat</code> <code>createContrat(@PathVariable Long id,</code> <code>@RequestBody Contrat</code> <code>contrat)</code>	<i>Requête HTTP < POST></i> <i>nous attendons une réponse HTTP 200 .</i> <i>+</i> <i>création d'un contrat.</i>	réussite
i09	<code>deleteContrat(@PathVariable Long id)</code>	<i>Requête HTTP < DELETE></i> <i>nous attendons une réponse HTTP 200 .</i> <i>+</i> <i>suppression d'un contrat.</i>	réussite

Produit - Test

i10	<code>getProduit ()</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i>	réussite
-----	----------------------------	--	-----------------

		<div>+</div> <div><i>recupérer tous les produits de la base de données.</i></div>	
i11	<pre>createProduit (@RequestBody Produit produit)</pre>	<div><i>Requête HTTP < POST></i></div> <div><i>nous attendons une réponse HTTP 200 .</i></div> <div>+</div> <div><i>création d'un produit.</i></div>	réussite

Authentication - Test

i12	<pre>authenticateUser (@Valid @RequestBody LoginRequest loginRequest)</pre>	<div><i>Requête HTTP < POST></i></div> <div><i>nous attendons une réponse HTTP 200 .</i></div> <div>+</div> <div><i>Connexion à l'API</i></div>	réussite
i13	<pre>registerUser (@Valid @RequestBody SignUpRequest signUpRequest)</pre>	<div><i>Requête HTTP < POST></i></div> <div><i>nous attendons une réponse HTTP 200 .</i></div> <div>+</div> <div><i>ajout d'un nouvel utilisateur.</i></div>	réussite

i14	<code>getCurrentUser(@CurrentUser UserPrincipal currentUser)</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>récupérer les utilisateurs connectés</i>	réussite
i15	<code>UserIdentityAvailability checkEmailAvailability (@RequestParam(value = "email") String email)</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>vérifier si un e-mail est disponible pour l'enregistrement</i>	réussite
i16	<code>UserProfile getUserProfile(@PathVariable(value = "username") String username)</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>obtenir le profil public d'un utilisateur.</i>	réussite
i17	<code>List<User> getUsers(@RequestParam(value = "page", defaultValue = PageConstants.DEFAULT_PAGE_NUMBER) int page, @RequestParam(value = "size", defaultValue =</code>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>récupérer utilisateurs.</i>	réussite

	<pre>PageConstants.DEFAULT_PAGE_SIZE) int size)</pre>		
i18	<pre>UserIdentityAvailability activateUser(@RequestParam value = "id") int userId)</pre>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>Activer un utilisateurs.</i>	réussite

Facture - Test

i19	<pre>void getContrats() throws Exception</pre>	<i>Requête HTTP < GET></i> <i>nous attendons une réponse HTTP 200 .</i> + <i>Récupérer la liste des factures.</i>	réussite
-----	--	--	-----------------

8. Glossaire

❖ Test unitaires :

C'est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme.

❖ Test D'intégrations:

les test d'intégration est une phase de tests, précédée par les tests unitaires, vérifiant le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme

(appelée «unité» ou « module ») ; dans le test d'intégration, chacun des modules indépendants du logiciel est assemblé et testé dans l'ensemble.

❖ **JUNIT**

JUnit est un Framework de test unitaire pour le langage de programmation Java.

❖ **Framework**

Un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel

❖ **API Rest**

Representational State Transfer Application Program Interface est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur.

❖ **java EE**

Une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise.

❖ **Spring**

Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. En 2004, Rod Johnson a écrit le livre Expert One-on-One J2EE Design and Development qui explique les raisons de la création de Spring.

❖ **PostgreSQL**

Un système de gestion de base de données relationnelle et objet. C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres, ou propriétaires.

❖ **HTML**

Un langage de balisage conçu pour faire des pages web.

❖ **CSS**

Un langage informatique qui permet de faire la présentation de l'HTML. Ce sont des feuilles de style en cascade.

❖ **React**

Une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état.

9. Sources

- https://fr.wikipedia.org/wiki/Test_unitaire
- <https://www.baeldung.com/spring-boot-testing>
- <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org.springframework.test.web.servlet.MockMvc.html>
- <https://openclassrooms.com/fr/courses/6900101-creez-une-application-java-avec-spring-boot/7078047-testez-et-deployez-votre-projet-web>
- <https://howtodoinjava.com/spring-boot2/testing/spring-boot-mockmvc-example/>

10. Index

Les mots-clés du document et où les trouver dans celui-ci

Tests fonctionnels 4, 6, 7, 8

.....

JUnit 22, 23, 24, 25, 26

.....

Test d'intégration 28, 27

.....

—