

Exercice 1.

- Quelle est la différence entre les tests statiques et les tests dynamiques ?
- Que sont la vérification et la validation ?
- Quels sont les différents niveaux de test ?
- Quelle est la différence entre re-test et test de régression ?
- Expliquez quels cas de test sont écrits en premier : les boîtes noires ou les boîtes blanches ?

Exercice 2.

- Faire correspondre les notions et les définitions.
 - 1) Tests d'intégration
 - 2) Scénarios de test
 - 3) Cas de test
 - 4) Script de test
 - 5) Test ascendant (bottom-up)
 - 6) Les tests de régression
 - a) Est une fonctionnalité qui peut être testée. Il est également appelé condition de test ou possibilité de test.
 - b) Est une approche des tests d'intégration, où les composants de niveau le plus bas sont testés en premier, puis utilisés pour faciliter le test des composants de niveau supérieur.
 - c) Il est écrit dans un langage de programmation et c'est un programme court utilisé pour tester une partie de la fonctionnalité du système logiciel.
 - d) Est un niveau de processus de test logiciel, où les unités individuelles d'une application sont combinées et testées.
 - e) Des tests effectués après que le logiciel a changé ou lorsque l'environnement a changé.
 - f) C'est un document qui contient les étapes qui doivent être exécutées ; c'est prévu plus tôt.

Exercice 3.

- Faire correspondre les notions et les définitions.
 - 1) phantom
 - 2) Scrum
 - 3) Modèle en V
 - 4) Selenium
 - 5) MuJava
 - a) Cycle de vie d'un logiciel
 - b) Outil de test de mutation
 - c) Outil de test logiciel pour GUI Windows
 - d) Modèle de développement agile
 - e) Outil de test Web

Exercice 4.

- Traduire les mots suivants en Français:

Anglais	Français	Anglais	Français
<i>black-box testing</i>		<i>defect</i>	
<i>boundary value</i>		<i>expected result</i>	
<i>control flow</i>		<i>fail</i>	
<i>coverage criteria</i>		<i>failure</i>	
<i>data flow</i>		<i>GUI</i>	
<i>software lifecycle</i>		<i>pass</i>	
<i>Load testing</i>		<i>regression testing</i>	
<i>functional requirement</i>		<i>reliability</i>	
<i>risk analysis</i>		<i>safety</i>	

<i>statement</i>		<i>stub</i>	
<i>system under test (SUT)</i>		<i>test automation</i>	
<i>test data</i>		<i>usability</i>	
<i>white-box testing</i>		<i>tester</i>	

Exercice 5.

Vous êtes un testeur travaillant sur une application e-commerce. Lors de vos tests, vous découvrez qu'un problème survient lorsque l'utilisateur essaie d'ajouter un produit au panier. Après avoir cliqué sur le bouton "Ajouter au panier", aucun message de confirmation n'apparaît et le produit ne s'ajoute pas au panier. Votre mission est de suivre le cycle de vie de ce défaut en répondant aux questions ci-dessous.

Questions :

1. **Détection et création du défaut :**
 - Quel titre donneriez-vous à ce défaut ?
 - Quelles informations incluriez-vous dans la description du défaut ?
2. **Correction :**
 - Quel type de tests le développeur doit-il effectuer avant de livrer la correction ?
3. **Re-test et validation :**
 - Quels tests effectuez-vous pour vérifier la correction ?
 - Si le bug est corrigé, quel statut lui attribuez-vous ?
 - Si le bug persiste, que faites-vous ?
4. **Cas particuliers :**
 - Que faites-vous si le chef de projet décide de reporter la correction à une future version ?
 - Que faire si un autre testeur a déjà signalé ce même bug ?

Exercice 6.

Vous êtes un testeur logiciel dans une entreprise qui développe une application de commerce en ligne. Lors de vos tests, vous identifiez les défauts suivants :

1. **Le bouton "Ajouter au panier" ne fonctionne pas.**
2. **Le logo de l'entreprise est mal aligné sur la page d'accueil.**
3. **L'application plante lorsque l'utilisateur valide son paiement.**
4. **Une faute d'orthographe dans la description d'un produit.**
5. **Un lien hypertexte vers les conditions générales redirige vers une page d'erreur 404.**

Question: Attribuez un niveau de **sévérité** (Critique, Majeure, Moyenne, Mineure) et de **priorité** (Élevée, Moyenne, Faible) à chaque défaut. Justifiez vos choix.

Exercice 7.

Dans le cadre d'un projet informatique de gestion d'un cabinet médical, l'analyste définit, dans le document SRS (Software Requirements Specification), le cas d'utilisation « S'authentifier » à l'aide d'une fiche descriptive et d'une interface utilisateur, comme suit :

Cas d'utilisation	S'authentifier
Acteur principale	Médecin
Acteur secondaire	//
Objectif	Ce cas permet à l'utilisateur de s'authentifier et d'ouvrir son espace
Pré-condition	L'acteur a déjà un compte.

Post-condition	L'utilisateur a l'accès à son espace
Scénario nominal	1. L'utilisateur demande d'accéder à son espace. 2. Le système répond par l'interface d'authentification. 3. L'utilisateur remplit le formulaire et valide. 4. Le système vérifie les informations saisies par l'utilisateur. 5. Le système affiche l'espace de l'utilisateur
Scénario alternative	5.1. Un message d'erreur est affiché « Les informations saisies sont non valides ». Reprend le au scénario nominal du point 2



Connexion

E-mail

Mot de passe

S'authentifier **Annuler**

Questions :

1) En vous basant sur les documents précédents qui décrivent le cas d'utilisation « S'authentifier », générez un ensemble de cas de test pour valider ce cas d'utilisation.

2) Supposons que, lors de la phase d'exécution des tests, un défaut ait été détecté : lorsque l'utilisateur saisit une adresse e-mail et un mot de passe valides, le système affiche un message d'erreur d'authentification.

Rédigez un rapport de défaut détaillé pour documenter ce problème.