

Les tests d'intégration du système

Table des matières

I. Différents types de tests	3
II. Exercice : Quiz	7
III. Exemple de test de vérification	8
IV. Exercice : Quiz	11
V. Essentiel	11
VI. Auto-évaluation	12
A. Exercice	12
B. Test	12
Solutions des exercices	12

I. Différents types de tests

Durée : 1 h 30

Prérequis :

- Les tests unitaires doivent avoir été réalisés pour chaque système intégré inférieur.
- Avoir installé Wamp, Xampp ou Lamp pour pouvoir créer une base de données MySQL.

Contexte

Une fois tous les composants d'un système regroupés il faut faire le test d'intégration système afin de vérifier que tout fonctionne ensemble. Pour cela, on doit vérifier la récupération, la modification et l'ajout de données.

Définition

Le **test logiciel** est le processus qui permet de vérifier qu'il n'y ait pas d'erreur et que le logiciel correspond aux exigences du client.

Le **test unitaire** est effectué par le développeur afin de vérifier son code au niveau des composants qu'il réalise.

Le **test d'intégration de système (SIT)** a pour objectif de s'assurer que toutes les dépendances des modules fonctionnent correctement et que toutes les données sont préservées entre les modules de l'ensemble du système.

Le **test système** a pour but de vérifier que le système dans son ensemble répond aux exigences définies dans les spécifications. On teste plusieurs exemples d'utilisation.

Le **test d'acceptation** a pour but de valider la conformité au cahier des charges du client ou de l'utilisateur final.

Fondamental **Les différents types de tests**

Il existe plusieurs types d'approches de test logiciel.

- **Test de la boîte blanche** : ce test est effectué sur le code lui-même.
- **Test de la boîte noire** : le test est fait sur les entrées et sorties du système le code interne n'est pas vérifié.
- **Test de la boîte grise** : c'est un mélange des deux tests précédents.

Attention

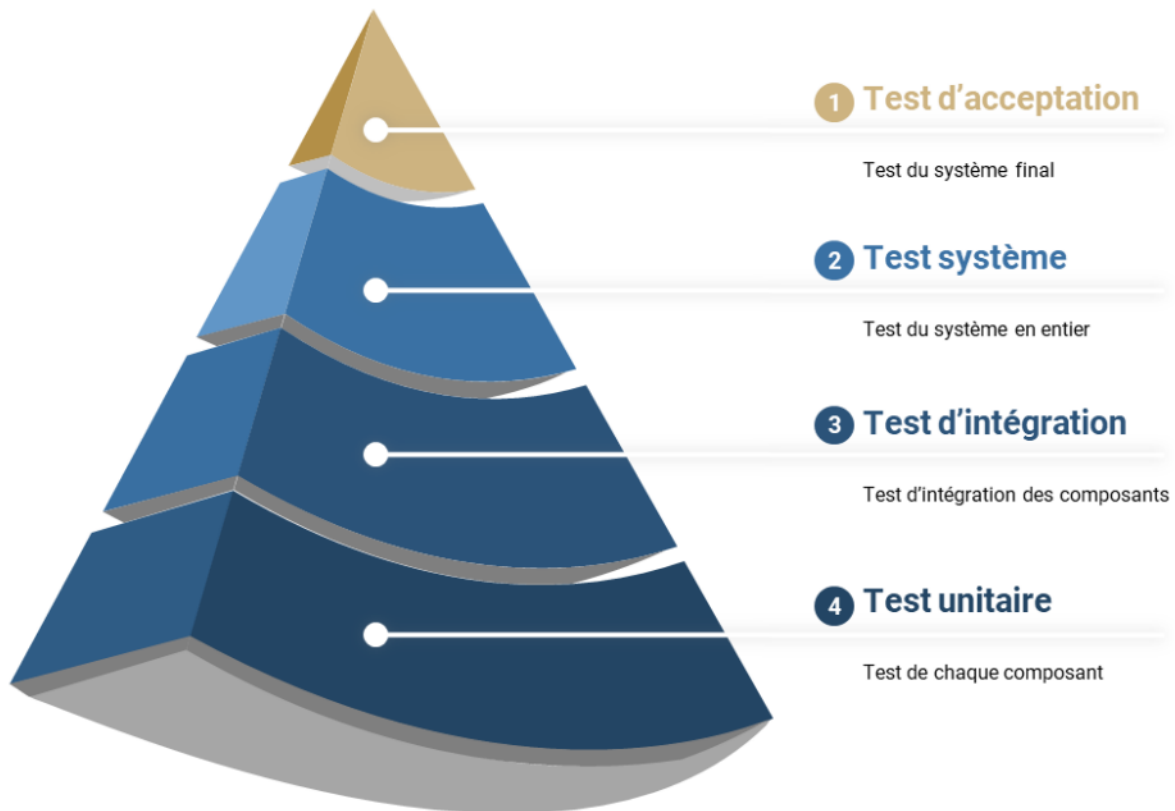
Attention à ne pas confondre test système et test d'intégration du système.

Les tests sont importants, car ils permettent de vérifier les éventuelles erreurs avant la livraison du logiciel au client, cela permet de montrer que le système est fiable.

Les avantages des tests logiciel sont les suivants :

- Garantir la sécurité du logiciel.
- Prévenir des bugs dès le départ et éviter d'avoir à corriger les erreurs par la suite ce qui assure un gain de temps.
- Garantir le bon fonctionnement et donc la satisfaction du client.

Les tests se font à différents niveaux cela peut être représenté sous forme de pyramide à 4 niveaux :



Ces tests peuvent être manuels ou automatisés. Lorsque les tests sont automatisés il faut commencer du bas vers le haut.

Quand est-il plus intéressant d'automatiser des tests ?

- Si l'équipe comprend plusieurs développeurs, il est intéressant que les tests soient automatisés afin que le développeur qui modifie son code vérifie que cela ne fasse pas dysfonctionner le reste du code.
- Si un projet est long, il est plus intéressant d'automatiser les tests afin de perdre moins de temps dessus et de privilégier la qualité du reste du code.
- Lorsqu'il y a des corrections régulières sur certains endroits du code afin de vérifier rapidement la bonne marche du code.
- Lorsqu'il y a beaucoup de données à traiter afin de ne pas perdre de temps à tout faire manuellement.

Il existe de nombreux logiciels d'automatisation de tests. Ces logiciels permettent de faciliter la gestion des tests, ils sont simples d'utilisation et permettent l'obtention de résultats rapides.

Définition

stub : module logiciel simulant la présence d'un autre composant, soit parce que cet autre composant est indisponible, soit parce qu'il n'est pas encore connu.

pilote : programme informatique qui permet d'interagir avec un périphérique comme par exemple pilote d'imprimante qui permet la communication entre l'ordinateur et l'imprimante.

cluster : bloc de système.

Les tests d'intégration du système

La plupart des projets étant réalisés en méthode agile, il est nécessaire de faire les tests d'intégration une fois toutes les parties mises en commun afin de vérifier le bon fonctionnement de l'ensemble et de s'assurer également que la transmission des données se fasse correctement.

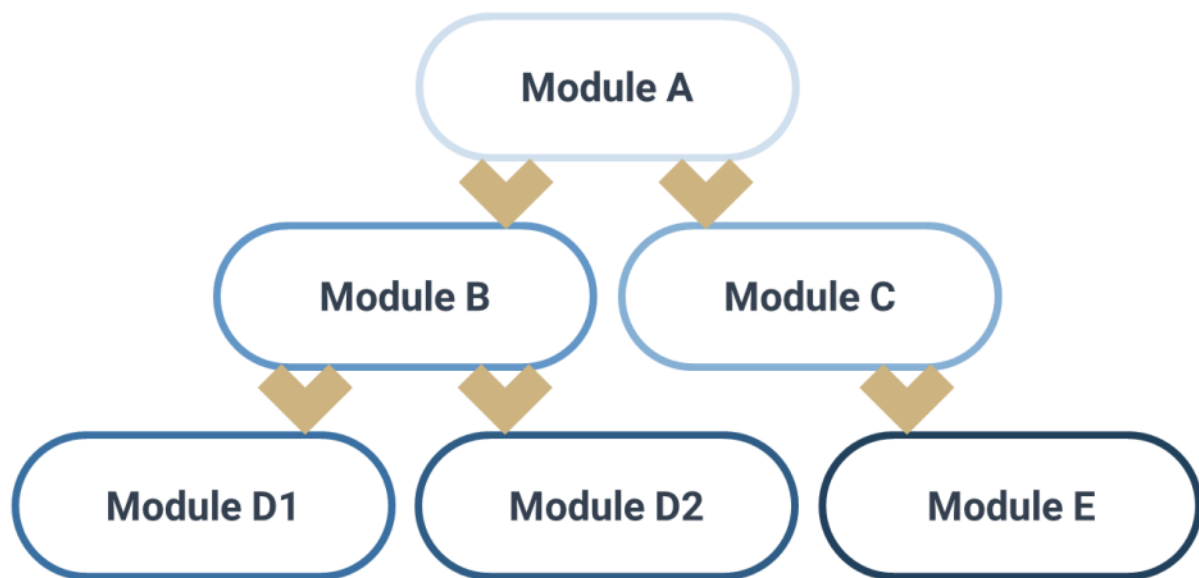
L'intégration a pour but de valider le fait que toutes les parties développées indépendamment fonctionnent bien ensemble. Ces tests sont indispensables afin de corriger les éventuelles pannes avant de déployer le système du côté de l'utilisateur.

Il existe 4 approches pour faire des tests d'intégrations du système :

- Approche descendante
- Approche ascendante (appelée aussi approche en profondeur)
- Approche sandwich
- Approche Big Bang

Approche descendante

Le test commence par le module le plus haut et les fonctionnalités des modules inférieurs sont simulées par des stubs. Une fois le module testé on le remplace par le module réel.



Avantages : il n'y a pas besoin de pilote et la localisation des erreurs est plus facile.

Inconvénients : la dépendance aux fonctionnalités des modules de niveau inférieur qui peut allonger le délai des tests le temps de remplacer les modules par des stubs.

Approche ascendante

Les modules du niveau le plus bas sont assemblés et forment des clusters qui servent de sous fonction de l'application. Un pilote est créé pour gérer l'entrée et la sortie lors du scénario de test du cluster. Une fois le test effectué, le pilote est supprimé et on combine le cluster avec le niveau supérieur, on procède de la même manière que précédemment et on continue jusqu'à avoir testé toute l'application.

Cette approche est recommandée pour les systèmes orientés objet et les systèmes en temps réel.

Avantages : il n'y a pas besoin de stubs, localisation facile des erreurs et les tests peuvent être fait en parallèle du développement.

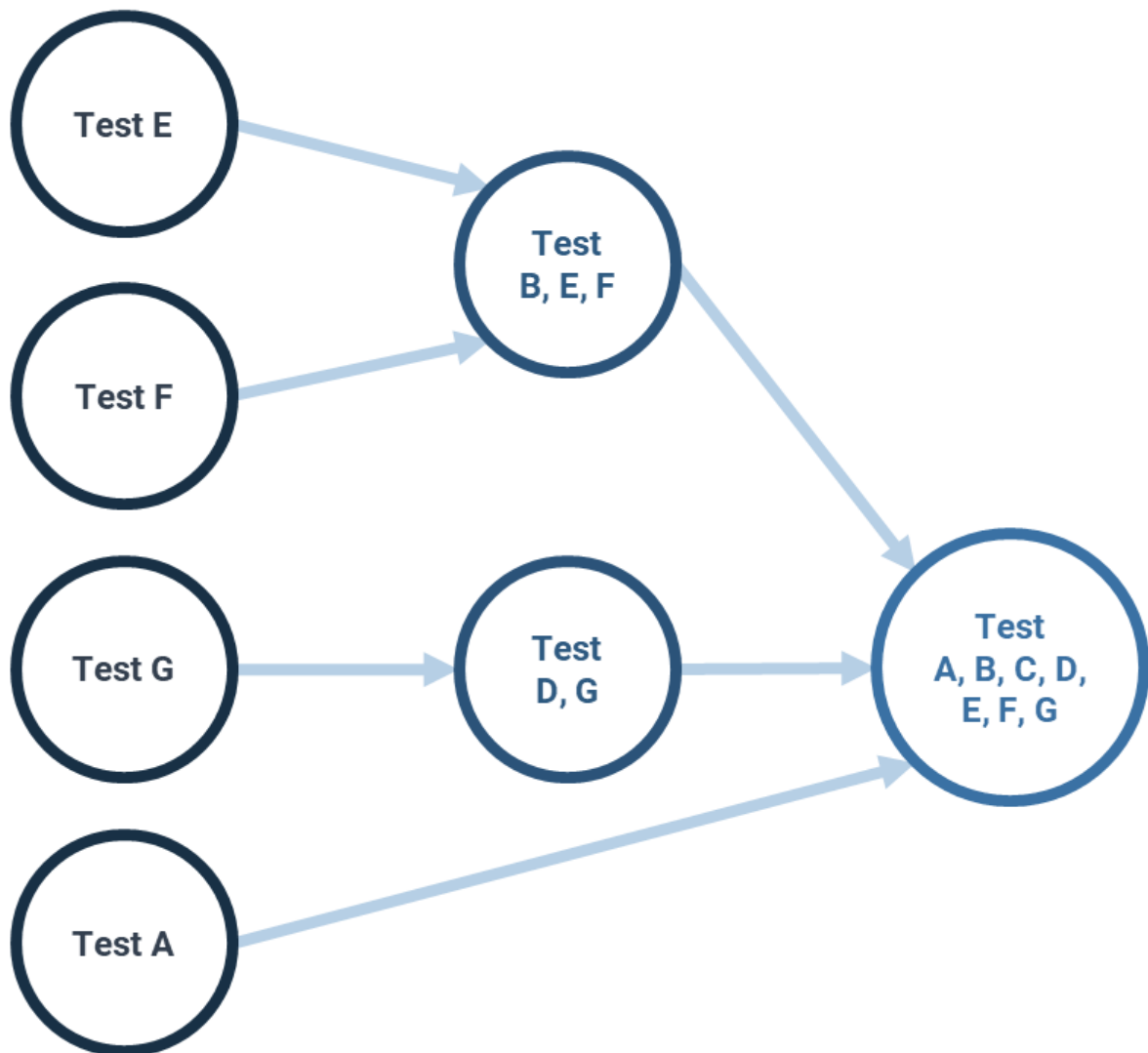
Inconvénients : l'interface utilisateur est testée en dernier.

Approche sandwich

Combinaison des approches descendantes et ascendantes. Le système est considéré comme s'il avait 3 couches : la couche que l'on veut tester, la couche supérieure et la couche inférieure. Les tests se font dans les 2 directions et se rassemblent au niveau de la couche intermédiaire.

Avantages : les couches supérieures et inférieures peuvent être testées en même temps.

Inconvénients : les sous-systèmes ne sont pas testés de manière individuelle et les tests sont plus complexes à mettre en place.

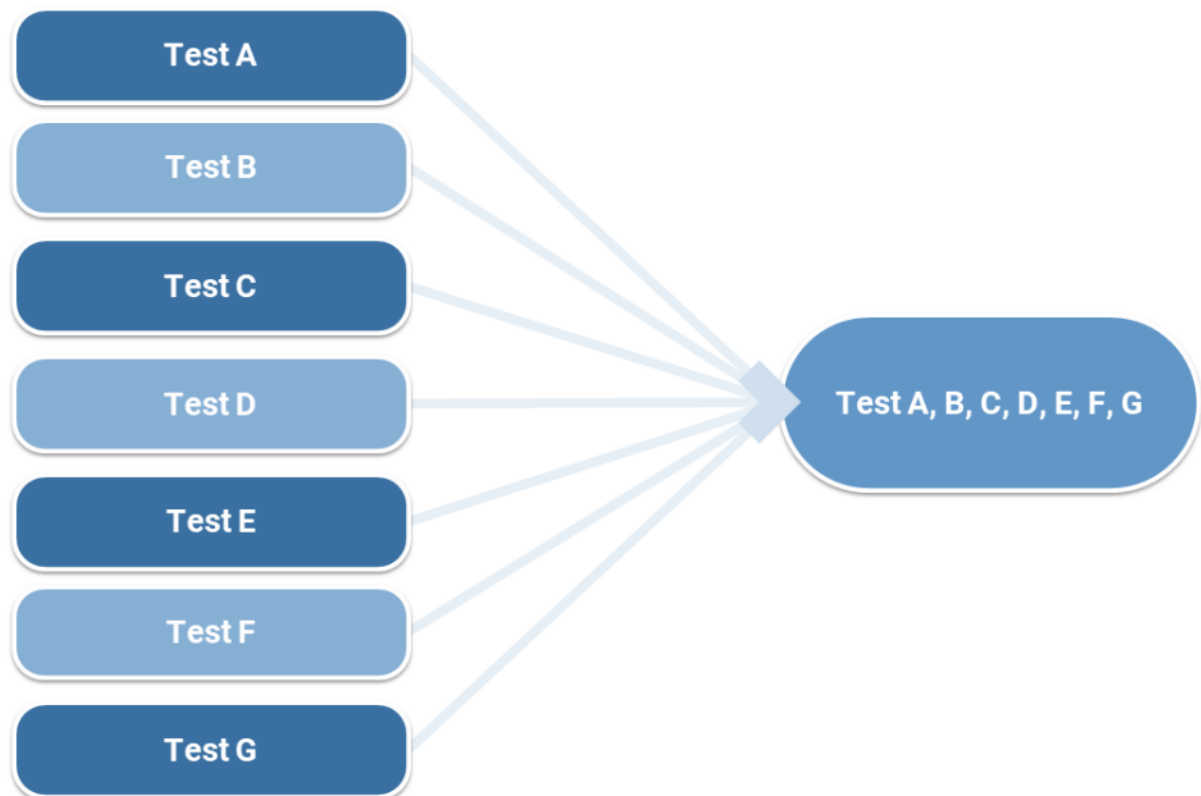


Approche Big Bang

Cette approche se fait lorsque les modules sont tous finis. On contrôle alors si le système fonctionne correctement.

Avantages : cela convient aux petits systèmes et permet de gagner du temps.

Inconvénients : difficulté à trouver la cause du problème, car tout est intégré en même temps et ne permet pas le développement en parallèle.



Une nouvelle forme d'intégration est en train d'apparaître avec l'utilisation de génie logiciel (comme Eclipse, Visual Studio, Jbuilder, etc.) : c'est l'intégration continue, qui est comme une fusion des tests unitaires et des tests d'intégrations.

Exercice : Quiz

[solution n°1 p.13]

Question 1

Le test d'intégration du système vérifie que le système, dans son ensemble, répond aux exigences définies dans les spécifications ?

- ☐ Vrai
- ☐ Faux

Question 2

Combien y a-t-il de niveaux sur le schéma de la pyramide des tests ?

- ☐ 3
- ☐ 4
- ☐ 5

Question 3

Combien y a-t-il de niveaux d'approches des tests d'intégration du système :

- ☐ 3
- ☐ 4
- ☐ 5

Question 4

L'approche descendante est une approche qui utilise des stubs.

- ☐ Vrai
- ☐ Faux

Question 5

Quel est l'inconvénient de l'approche ascendante ?

- ☐ L'interface utilisateur est testée en dernier
- ☐ L'interface utilisateur est testée en premier

III. Exemple de test de vérification

Contexte

Nous avons un site relié à une base de données MySQL et pour aller sur la page d'accueil il faut se connecter.

On arrive sur la page de connexion, une fois les identifiants rentrés et le clic sur le bouton « *connecter* » on doit être redirigé sur la page d'accueil.

Objectif du test : vérifier le lien entre la page de login et la page d'accueil.

Description : entrer les identifiants de connexion et cliquer pour se connecter.

Résultat attendu : être redirigé sur la page d'accueil à la suite de la connexion.

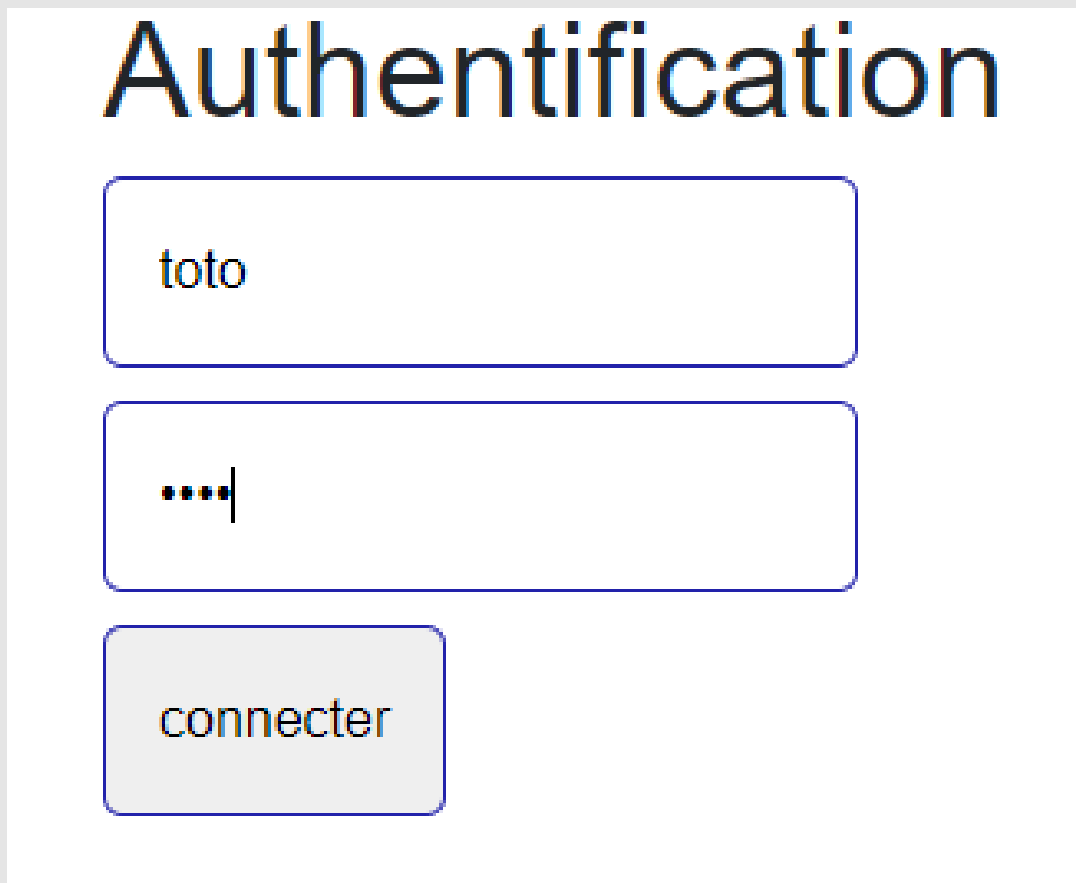
Méthode

Étape 1 : page d'authentification

Authentification

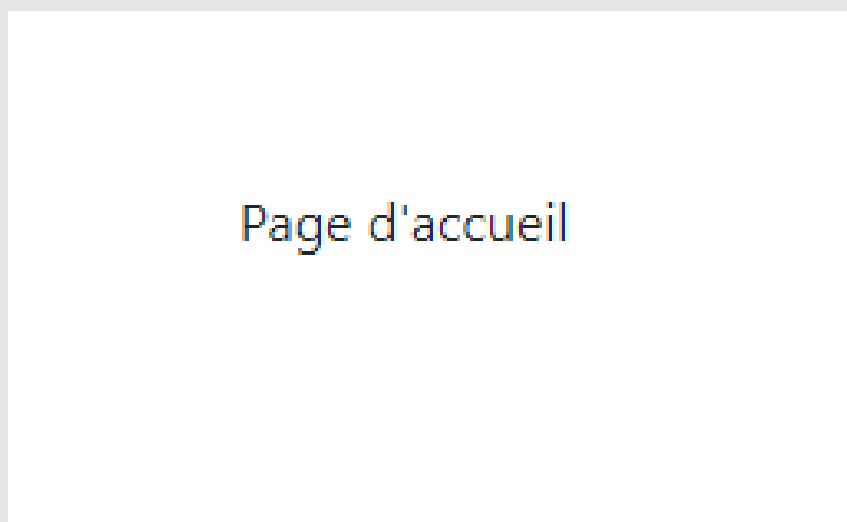
Nous avons un module d'authentification fonctionnel qui nous permet de nous connecter.

Étape 2 : pour le test on tape le login et le mot de passe valide puis on clique sur le bouton connecter et on regarde le résultat. Après avoir fait cela, on est bien redirigé sur la page d'accueil.

A screenshot of a web application's authentication page. At the top, the word "Authentication" is displayed in a large, bold, black font. Below it, there are two input fields. The first field contains the text "toto". The second field contains four dots, indicating a password. Below these fields is a button labeled "connecter". The entire form is enclosed in a light gray border.

Étape 3 : résultat

Nous avons bien été redirigés vers la page d'accueil.



Étape 4 :

Maintenant il faut vérifier également que lorsqu'on rentre un login ou un mot de passe faux, qu'il n'y ait pas de connexion.

Authentification

Étape 5 :

Résultat : pas de connexion

Authentification

Mauvais login ou mot de passe!

En conclusion, on peut voir que le test est validé, lorsque le login et mot de passe sont corrects la connexion s'effectue et lorsqu'un des deux est incorrect la connexion ne se fait pas.

Exercice : Quiz

[solution n°2 p.14]

Question 1

Le sigle SIT signifie :

- ☐ Test d'intégration du système
- ☐ Test du système
- ☐ Système d'intégration du test

Question 2

Le test unitaire vérifie le code au niveau des composants ?

- ☐ Vrai
- ☐ Faux

Question 3

Dans quel ordre doit-on effectuer les tests ?

- ☐ Test unitaire, test système, test d'intégration du système, test d'acceptation
- ☐ Test unitaire, test d'intégration du système, test système, test d'acceptation
- ☐ Test unitaire, test d'acceptation, test d'intégration du système, test système

Question 4

La meilleure approche de test d'intégration du système est la méthode sandwich.

- ☐ Vrai
- ☐ Faux

Question 5

Le test d'intégration du système ne vérifie pas la transmission des données.

- ☐ Vrai
- ☐ Faux

V. Essentiel

Il existe 4 types de tests qui doivent être faits dans l'ordre suivant :

- Test unitaire qui vérifie le code au niveau des composants,
- Test d'intégration du système qui vérifie essentiellement le transfert des données d'un module à l'autre,
- Test système qui vérifie le système dans son ensemble,
- Test d'acceptation qui vérifie la conformité aux attentes du client.

Pour le test d'intégration du système il existe 4 approches différentes :

- Approche descendante
- Approche ascendante
- Approche sandwich
- Approche Big Bang

Ces tests sont donc importants pour vérifier le bon fonctionnement, éviter les bugs, et cela permet de satisfaire le client.

VI. Auto-évaluation

A. Exercice

Vous comptez déployer une application de chat en ligne, cette application nécessite une connexion, l'espace de chat est limité à 5 personnes maximum. Un carnet de contact permet d'ajouter ou supprimer des contacts.

Question

[solution n°3 p.15]

Écrivez 5 tests d'intégration en respectant la forme objectif - description - résultat.

B. Test

Exercice 1 : Quiz

[solution n°4 p.15]

Question 1

Quel est l'avantage de l'approche sandwich ?

- ☐ Les couches supérieures et inférieures peuvent être testées en même temps
- ☐ La couche supérieure est testée avant la couche inférieure
- ☐ La couche inférieure est testée avant la couche supérieure

Question 2

Quand peut-on utiliser l'approche Big Bang ?

- ☐ À chaque module créé
- ☐ Une fois tous les modules finis

Question 3

Il y a transmission de données lors d'une connexion avec un login mot de passe.

- ☐ Vrai
- ☐ Faux

Question 4

Peut-on faire les tests d'intégration du système sans avoir fait les tests unitaires ?

- ☐ Oui
- ☐ Non

Question 5

Peut-on faire les tests logiciels dans n'importe quel ordre ?

- ☐ Oui
- ☐ Non


Solutions des exercices

Exercice p. 7 Solution n°1**Question 1**

Le test d'intégration du système vérifie que le système, dans son ensemble, répond aux exigences définies dans les spécifications ?

☐ Vrai

☒ Faux

 C'est le test système qui s'assure de cela. Le test d'intégration du système s'assure du bon fonctionnement et des préservations des données entre chaque module.

Question 2

Combien y a-t-il de niveaux sur le schéma de la pyramide des tests ?

☐ 3

☒ 4

☐ 5

 Il y a 4 niveaux de tests : tests unitaires, tests d'intégration système, tests système et tests d'acceptation.


Question 3

Combien y a-t-il de niveaux d'approches des tests d'intégration du système :

☐ 3

☒ 4

☐ 5


 Il y a 4 niveaux d'approches (descendante, ascendante, sandwich et Big Bang).

Question 4

L'approche descendante est une approche qui utilise des stubs.

☒ Vrai

☐ Faux


 Le test commence par le module le plus haut et les fonctionnalités des modules inférieurs sont simulées par des stubs.

Question 5

Quel est l'inconvénient de l'approche ascendante ?

☒ L'interface utilisateur est testée en dernier


☐ L'interface utilisateur est testée en premier

 Les modules du bas sont testés en premier et en montant, donc l'interface utilisateur est testée en dernier.

Exercice p. 11 Solution n°2


Question 1

Le sigle SIT signifie :

- ☒ Test d'intégration du système
- ☐ Test du système
- ☐ Système d'intégration du test
-  Cela signifie test d'intégration du système (System Integration Testing).


Question 2

Le test unitaire vérifie le code au niveau des composants ?

- ☒ Vrai
- ☐ Faux
-  Le développeur fait un test unitaire de son code à chaque composant qu'il crée.


Question 3

Dans quel ordre doit-on effectuer les tests ?

- ☐ Test unitaire, test système, test d'intégration du système, test d'acceptation
- ☒ Test unitaire, test d'intégration du système, test système, test d'acceptation
- ☐ Test unitaire, test d'acceptation, test d'intégration du système, test système
-  Il faut effectuer dans l'ordre test unitaire, test d'intégration du système, test système et test d'acceptation comme montré dans le 1^{er} schéma. Cela permet, au départ, de tester des petites parties de code jusqu'à tester le système entier et la validation des attentes du client au final.


Question 4

La meilleure approche de test d'intégration du système est la méthode sandwich.

- ☐ Vrai
- ☒ Faux
-  Chaque méthode a ses avantages et ses inconvénients, il n'y en a pas de meilleure que d'autres, tout dépend de ce que l'on veut faire.

Question 5

Le test d'intégration du système ne vérifie pas la transmission des données.

- ☐ Vrai
- ☒ Faux
-  Le test d'intégration du système vérifie la transmission des données d'un module à un autre, c'est le principe même de ce test.


p. 12 Solution n°3

Objectif	Description	Résultat
Vérifier le lien entre espace login et espace user	Entrez les identifiants de connexion et cliquez sur le bouton Connexion	Être redirigé vers son espace personnel
Créer une conversation	Cliquez sur un contact	Ouverture d'une nouvelle fenêtre de chat à destination du contact
Vérifier la limite de 5 personnes par conversation	Ajoutez 2, 3, 4, 5, 6 personnes dans une conversation	Une fois 5 personnes ajoutées dans la même conversation, l'application doit retourner un message d'information sur la limite de 5 personnes maximum par conversation
Vérifier l'ajout de contact	Cliquez sur créer un nouveau contact, ajouter des données et enregistrer	Le contact doit apparaître dans le carnet de contact
Vérifier la suppression de contact	Cliquez sur l'icône de suppression d'un contact	Le contact doit disparaître du carnet de contact et apparaître dans la corbeille.

Exercice p. 12 Solution n°4

Question 1


Quel est l'avantage de l'approche sandwich ?

- ☒ Les couches supérieures et inférieures peuvent être testées en même temps
- ☐ La couche supérieure est testée avant la couche inférieure
- ☐ La couche inférieure est testée avant la couche supérieure
-  Les tests se font dans les 2 directions : les couches supérieures et inférieures peuvent donc être testées en même temps.

Question 2

Quand peut-on utiliser l'approche Big Bang ?

- ☐ À chaque module créé
- ☒ Une fois tous les modules finis


-  Cette approche se fait lorsque les modules sont tous finis. On contrôle alors si le système fonctionne correctement.

Question 3

Il y a transmission de données lors d'une connexion avec un login mot de passe.

☒ Vrai

☐ Faux


-  Il y a transmission de données, car il faut aller vérifier dans la base de données que le login et mot de passe soit bien enregistré afin de donner l'autorisation de connexion ou non.

Question 4

Peut-on faire les tests d'intégration du système sans avoir fait les tests unitaires ?

☐ Oui

☒ Non


-  Il faut d'abord avoir fait les tests unitaires afin de vérifier que chaque partie du code fonctionne avant de faire les tests d'intégration du système.

Question 5

Peut-on faire les tests logiciels dans n'importe quel ordre ?

☐ Oui

☒ Non

-  Il y a un ordre précis : tests unitaires, tests d'intégration du système, tests système et tests d'acceptation.