

Projet IDDY - Plan de test

Ce document décrit le plan de test complet du projet I Double Dare You développé grâce à l'utilisation de JEE.

Périmètre de test

L'application sera testée selon deux méthodes : les tests fonctionnels avec Selenium et les tests de charge avec JMeter selon les différents scénarios de test. Les fonctionnalités à tester pour mettre en valeur le bon fonctionnement de l'application sont listées ci-après. Ces fonctionnalités sont les plus importantes et les plus critiques dans la garantie du bon fonctionnement.

Objectifs et portée des tests

L'application a été testée selon deux méthodes : les tests fonctionnels avec Selenium et les tests de charge avec JMeter selon les différents scénarios de test. L'objectif étant avant tout de documenter la méthode plutôt que les tests eux-mêmes, étant donné que l'application possède un nombre limité de fonctionnalités.

Méthodologie pour les tests fonctionnels

Les tests fonctionnels ont été effectués avec Selenium sur une exécution locale de l'application. Les fonctionnalités principales ont été testées pour garantir une utilisation correcte de la plateforme web développée dans le cadre du projet. Les tests sont réalisés en enregistrant des séquences de cliques et d'entrées utilisateur dans un contexte précis. Elles peuvent ensuite être relancées avec Selenium de façon automatisée. Les tests sont ensuite exportés au format HTML compatible Selenium et sous forme de code Java Junit 4.0 avec Webdriver.

Méthodologie pour les tests de charge

Les tests de performance sont réalisés avec le programme JMeter. Le programme installé sur Windows dans notre cas précis permet d'automatiser l'exécution et l'envoi des requêtes HTTP sur notre application locale. Le but étant d'effectuer un grand nombre de requêtes pour déterminer le comportement de la plateforme face à une charge élevée. Cette suite de test est exportée sous la forme d'un fichier compatible avec JMeter d'extension .jmx

Tests fonctionnels

| Nom | Type de test | Niveau de test | Test |
|---|----------------|----------------|-------------|
| Authentification | Fonctionnalité | Test système | Must test |
| L'authentification comprend l'inscription et la connexion des utilisateurs. Ces derniers doivent pouvoir se créer un nouveau compte en spécifiant toutes les informations nécessaires et ensuite se connecter à la plateforme en réutilisant la même combinaison login et mot de passe. | | | |
| Abonnement | Fonctionnalité | Test système | Should test |
| Le système d'abonnement permet à un utilisateur d'avoir des "amis" auxquels il pourra proposer des défis. En effet, on ne peut proposer des défis qu'aux utilisateurs qui nous suivent. | | | |
| Recherche | Fonctionnalité | Test système | Must test |
| La recherche doit permettre de retrouver un utilisateur spécifique en filtrant les résultats à afficher sur la page concernée. La recherche doit se faire rapidement et efficacement. | | | |
| CRUD de défis | Fonctionnalité | Test système | Could test |
| Les défis sont les points centraux de l'application. Il est donc nécessaire de pouvoir les créer, mettre à jour et supprimer. Les informations spécifiées lors de la création d'un défi doivent être correctes. | | | |
| Navigation agréable | Ergonomie | Validation | Could test |
| La navigation sur l'application doit être instinctive et facile d'utilisation. C'est un élément important pour garantir un confort utilisateur. | | | |

Tests de performance

Les tests de performances sont effectués selon trois scénarios différents pour chacune des fonctionnalités testées.

| Nom | Type de test | Niveau de test | Test |
|-------------------------|---|---------------------------|------------------|
| Charge de pages | Performances | Test d'acceptation | Must test |
| <i>Scénario 1</i> | 10 utilisateurs sur la page d'accueil Temps de réponse inférieur à une seconde | | |
| <i>Scénario 2</i> | 100 utilisateurs sur la page d'accueil Temps de réponse inférieur à une seconde | | |
| <i>Scénario 3</i> | 1000 utilisateurs sur la page d'accueil Temps de réponse inférieur ou égal à une seconde | | |
| Authentification | Performances | Test d'acceptation | Must test |
| <i>Scénario 1</i> | 10 connexions sur la plateforme Temps de réponse inférieur à une seconde | | |
| <i>Scénario 2</i> | 100 connexions sur la plateforme Temps de réponse inférieur à une seconde | | |
| <i>Scénario 3</i> | 1000 connexions sur la plateforme Temps de réponse inférieur ou égal à une seconde | | |
| Recherche | Performances | Test d'acceptation | Must test |
| <i>Scénario 1</i> | 10 recherches sur la plateforme Temps de réponse inférieur à une seconde | | |
| <i>Scénario 2</i> | 100 recherches sur la plateforme Temps de réponse inférieur à une seconde | | |
| <i>Scénario 3</i> | 1000 recherches sur la plateforme Temps de réponse inférieur ou égal à une seconde | | |

Stratégie et approche détaillée de test

| Caractéristique | Risk class | Test système | Acceptance utilisateur | Acceptance production |
|-------------------------------|------------|--------------|------------------------|-----------------------|
| Fonctionnalités | | | | |
| <i>Inscription</i> | A | ●●● | ● | |
| <i>Connexion</i> | A | ● | ● | |
| <i>Recherche</i> | C | ● | ●●● | |
| <i>Création de défi</i> | A | ● | ●●● | |
| Performance | B | | ● | |
| Facilité d'utilisation | B | | ●● | |
| Sécurité | C | | | |

Légendes

A → Risque élevé

B → Risque moyen

C → Risque faible

●●● → Ampleur élevée

●● → Ampleur moyenne

● → Ampleur faible

Livrables

Les livrables pour ce projet sont les suivants :

- Spécification de tests
- Rapport de tests
- Démonstration du projet

Infrastructure et logistique de test

Environnement de tests

Les tests sont tous effectués en local, le serveur et le navigateur se trouvant sur la même machine (localhost). Le temps de connexion réseau n'est donc pas pris en compte dans la configuration de tests puisqu'elle est dans tous les cas quasi-instantanée.

Tests système

Tests fonctionnels automatisés

Tests d'acceptance

Tests de performances automatisés

Logiciels & Matériel de test

- Selenium
- JMeter
- Firefox

Gestion des risques

Déterminer les risques de test

| N° | Description | Source | Impact | Probabilité | Gravité |
|----|--|----------------------|--------|-------------|---------|
| 1 | Accès non-autorisé | Utilisateur / hacker | 80 | 0.3 | 80 |
| 2 | Injection de données | Utilisateur / hacker | 80 | 0.1 | 80 |
| 3 | Retard du développement | Développeurs | 80 | 0.5 | 80 |
| 4 | Manque d'expertise pour les techniques de test | Développeurs | 60 | 0.5 | 60 |
| 5 | Manque de compétences fonctionnelles | Développeurs | 80 | 0.5 | 60 |