Bilan de campagne de test pour 724Events

Sommaire

Introduction

Ce document présente le bilan des tests effectués pour l'application 724Events.

Nous y aborderons les tests fonctionnels décrits dans le cahier de recette ainsi que les tests exploratoires.

Pour ces derniers, nous avons testé avec plusieurs outils différents bien connus du monde du testing. Nous les aborderons au moment opportun.

Puis, nous comparerons les résultats obtenus avec ceux attendus avant de finir sur le mot de la fin, le Go ou No Go.

Présentation des objectifs

L'objectif de cette campagne de test est d'évaluer la qualité de la V1 de l'application 724Events.

Pour rentrer dans les détails, cela nous donne :

- Valider le cahier de recette
- Exécuter des tests exploratoires divers
- Valider le fonctionnement global de l'application dans sa globalité (frontend et backend à la fois)
- Valider les fonctionnalités principales qui sont la présentation des événements et l'inscription à un événement
- Valider les fonctionnalités secondaires comme par exemple, la prise de contact par un utilisateur via un formulaire, la pagination des événements etc...
- Valider le fonctionnement de l'application sur un mobile (responsive).
- Valider le fonctionnement de l'API en isolée
- Vérifier l'absence de failles de sécurité critiques
- Déterminer les anomalies et les répertorier avec leurs origines afin de faciliter leurs résolutions si possibles
- Rapport des observations

Outils utilisés

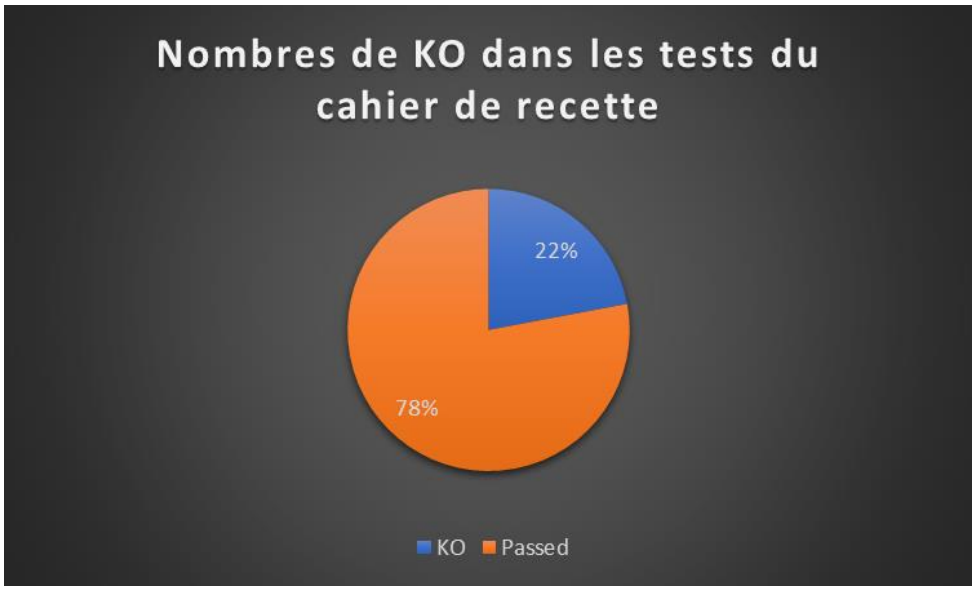
Le tableau ci-dessous récapitule les outils utilisés.

Outil	Utilisation
Navigateur internet	Utilisé pour la navigation, la visualisation des résultats mais aussi pour la recherche d'informations non visibles par l'utilisateur lambda.
Postman	Utilisé pour tester l'API de manière plus poussée
Zaproxy	Utilisé pour tester la sécurité globale de l'api
Cypress	Utilisé pour automatiser les tests
Swagger	Utilisé en premier lieu pour tester simplement l'API
SonarQube	Utilisé pour tester la qualité globale du code
Jira	Utilisé pour décrire et répertorier les anomalies rencontrées.

Présentations des résultats

Cahier de recettes

Le cahier de recette présente un total de 59 tests réparties sur plusieurs éléments du site. Nous avons décelé 13 anomalies ce qui représente un total de 18%.



Ces anomalies sont réparties de la façon suivante :

Nous pouvons voir clairement que le formulaire ainsi que le slider posent problèmes.

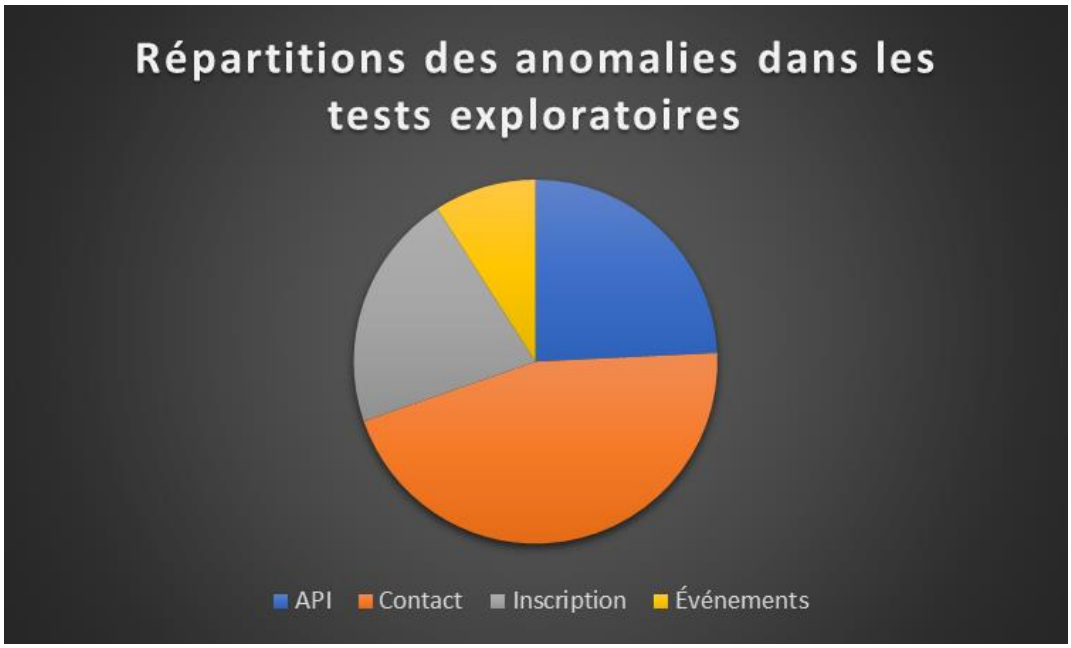
Enfin, concernant la répartition de la criticité, nous voyons clairement une majorité de criticité majeure et mineure :

Pour l’anecdote, quasiment tous les défauts critiques concernent le formulaire de contact.

Tests exploratoires

Les tests exploratoires ont porté leurs fruits et ont révélé un total de 30 anomalies. En effet, nous avons testé l'API, navigué sur l'application, testé la sécurité, testé la qualité générale du code. Nous en reparlerons dans les observations ci-dessous.

Ces anomalies sont sans surprises centrées sur le formulaire de contact et d'inscription comme on peut le voir :



Certaines anomalies ont même fait planter l'application et cela a nécessité un redémarrage à la main.

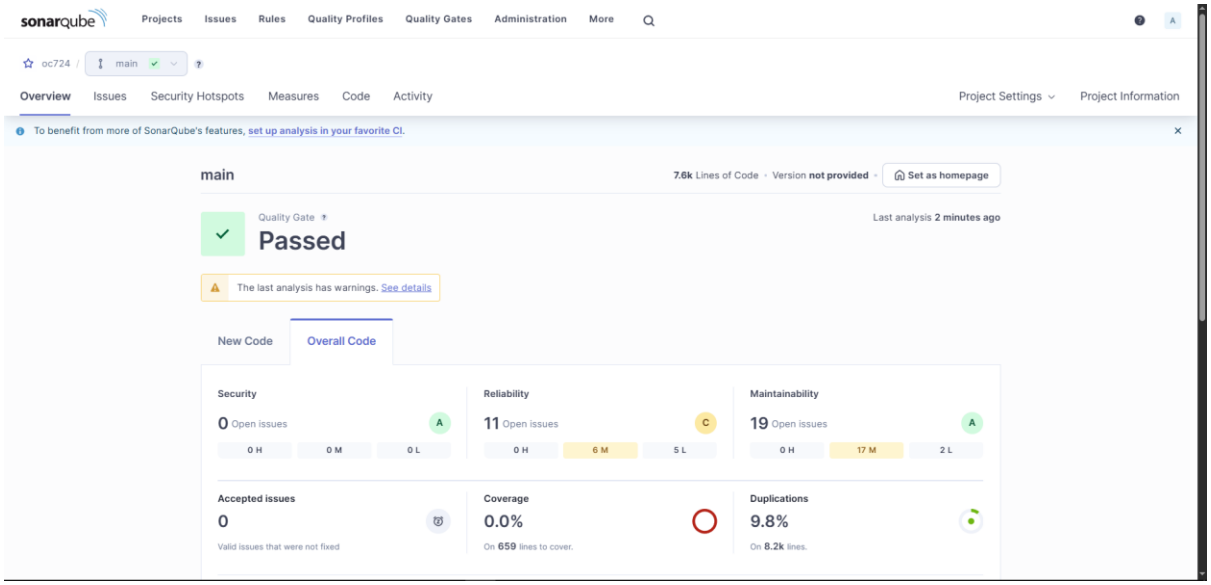
Nous avons été le plus précis possible ce qui explique ce grand nombre d'anomalies.

Finalement, nous avons donc 43 anomalies déterminées dans cette application.

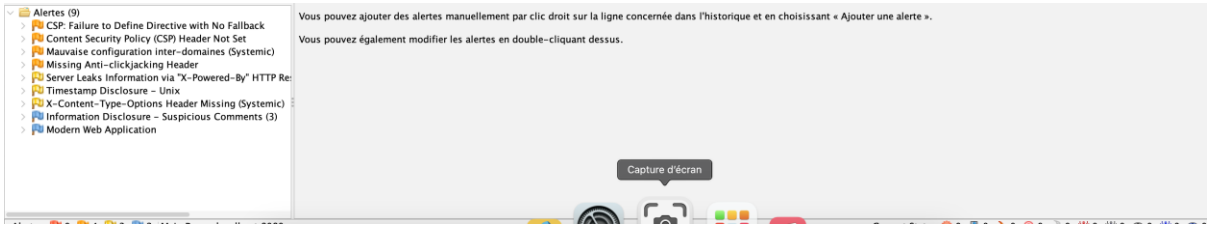
Observations

Lors des tests exploratoires et du cahier de recette, nous avons pu faire quelques observations :

- “Oeuvre” est mal orthographiée dans la page de la description du WEF.
    - La correction est “œuvre”
- Favicon
    - La favicon est celle du framework React et non celle de l’application
- Responsive
    - Le site “commence à tirer la tête” dès que la largeur descend en dessous de 436px.
- Image section
    - L’image de la section “Slider” est trop large, elle déborde ce qui “casse” le layout.
- Qualité générale du code
    - La qualité générale du code est “Passed” comme le montre la capture ci-dessous



- Sécurité
    - L’application ne présente pas de failles de sécurité critique comme le montre la capture ci-dessous.



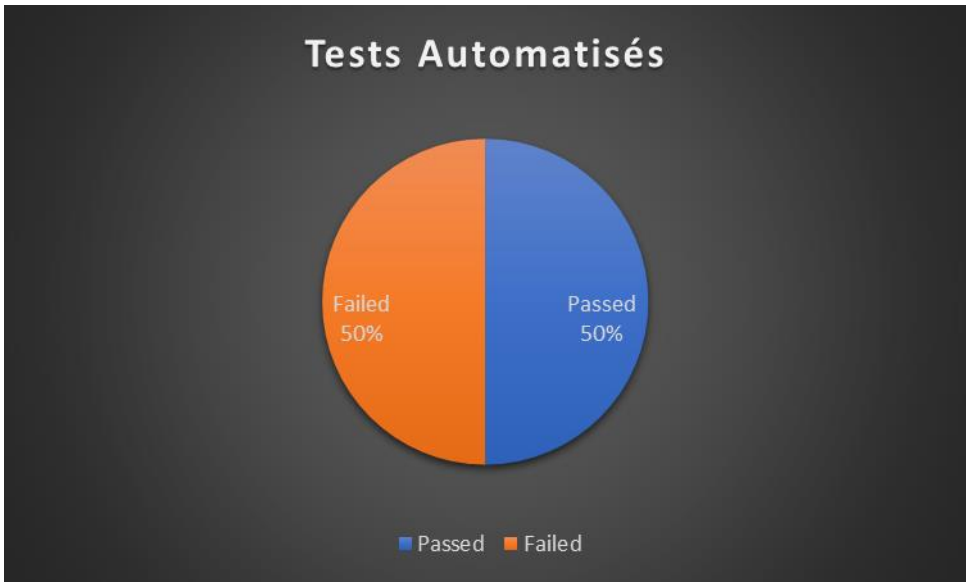
Tests automatisés

Nous avons choisi d'automatiser le formulaire de contact et la récupération du dernier événement car ces derniers sont la pierre angulaire de l'application.

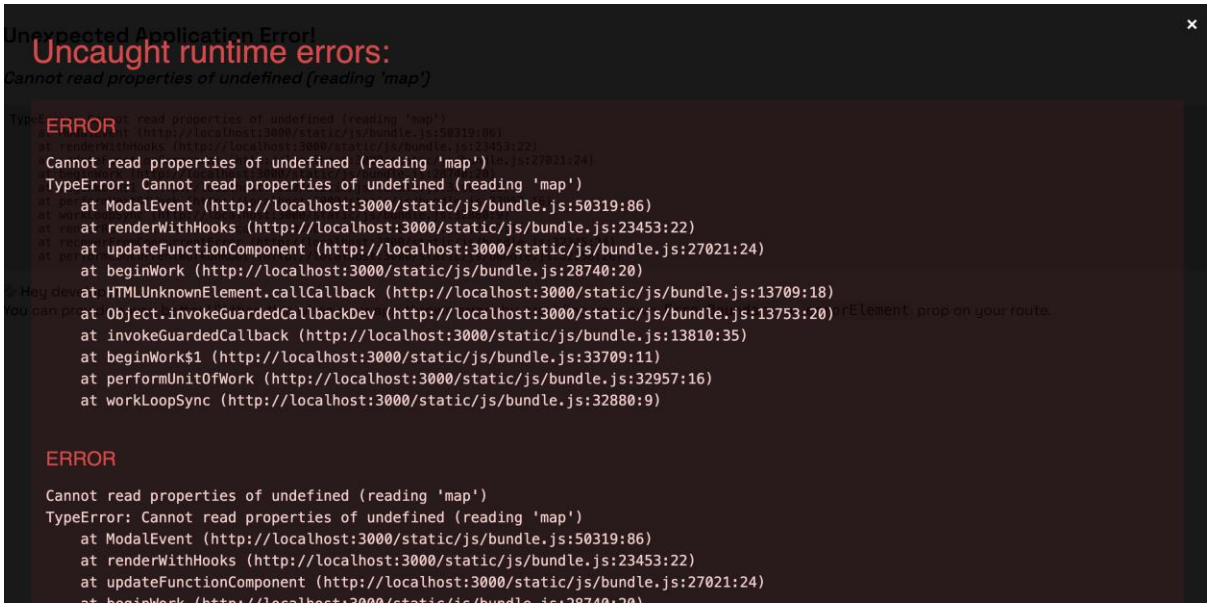
Le premier est une source potentielle de bug mais aussi la porte d'entrée des personnes malveillantes, raisons pour laquelle il a été choisi.

Le deuxième étant dynamique et intéressant pour le visiteur doit fonctionner.

Sur les 2 tests, 1 a été "Passed". Celui du formulaire.



La cause de ce test "Failed" est une erreur dans le code :



Recommandations

Afin de faciliter les tests futurs, ne pas hésiter à privilégier les id uniques pour les éléments frontend. En effet, la sélection de l'élément est plus simple, plus rapide mais surtout plus robuste car il faudra changer chaque test ou l'élément est compris afin d'être à jour si un développeur change une partie du DOM.

Concernant les inputs des formulaires, privilégier les outils déjà en place des frameworks frontend afin de déterminer la validité ou au minimum utiliser une expression régulière.

Conclusion

L'application en l'état n'est pas déployable à cause des anomalies majeures et critiques rencontrées.

C'est un "No Go" pour la V1.

Le “Go” est atteignable si les corrections se focalisent sur les parties critiques notamment concernant la sécurité mais aussi sur le formulaire de contact...