Dijonons

# Rapport de Tests

[Sous-titre du document]

Groupe Dijonons
30/06/2023

# Table des matières

# Introduction :

Identification de la nécessité de tester l'application Lors du développement de notre application, il est essentiel d'effectuer des tests pour plusieurs raisons. Tout d'abord, les tests nous permettent de vérifier que toutes les fonctionnalités de l'application fonctionnent correctement et répondent aux besoins des utilisateurs. Les tests nous aident également à identifier les bugs et les erreurs qui pourraient compromettre la qualité de l'application. De plus, les tests nous donnent l'assurance que notre application est fiable, sécurisée et qu'elle offre une expérience utilisateur optimale.

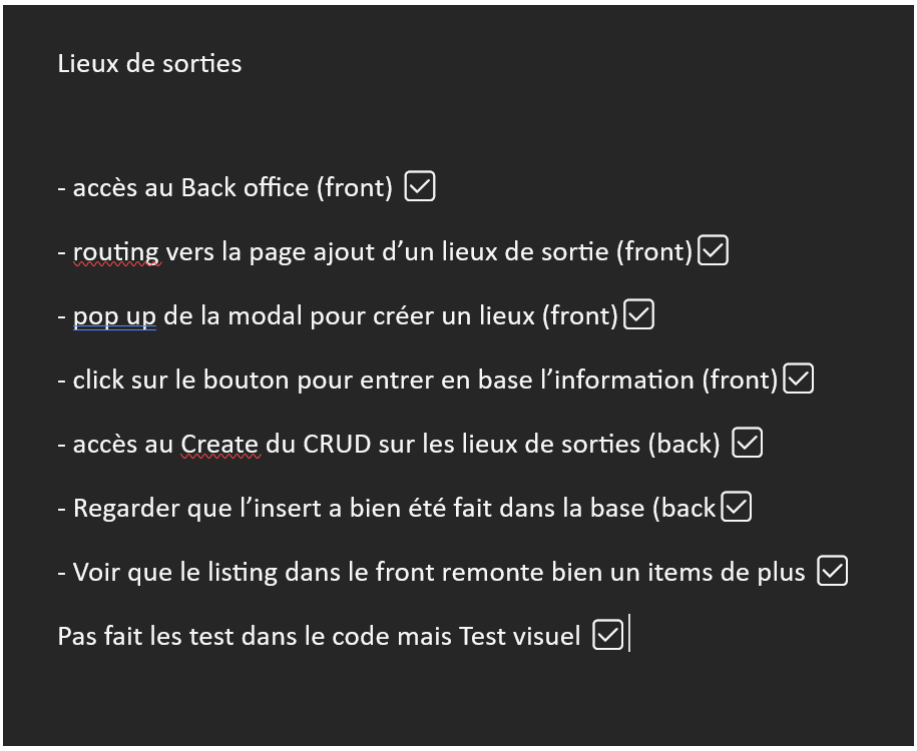
Focalisation sur les bugs récurrents dans les fonctionnalités largement utilisées Au cours des tests, nous avons remarqué certains bugs récurrents dans les fonctionnalités les plus utilisées de l'application. Ces bugs peuvent entraîner des dysfonctionnements, des erreurs d'affichage ou des résultats inattendus. Nous avons donc accordé une attention particulière à ces fonctionnalités lors de nos tests, en nous assurant de les examiner de manière approfondie et de les corriger adéquatement.

Imaginer les comportements insolites des utilisateurs En plus des tests traditionnels, nous avons également cherché à simuler des comportements insolites de la part des utilisateurs. Nous avons imaginé différentes situations dans lesquelles les utilisateurs pourraient interagir de manière inattendue avec l'application. Cela nous a permis de vérifier si l'application était capable de gérer ces situations de manière appropriée, en évitant les plantages, les erreurs de traitement ou les résultats imprévus.

Pour mener à bien nos tests, nous avons séquencé et identifié toutes les actions nécessaires pour tester chaque fonctionnalité de l'application. Nous avons établi une liste complète des scénarios de test en nous basant sur les spécifications fonctionnelles de l'application. Chaque scénario de test comprenait une série d'étapes détaillées à suivre, mettant en évidence les actions à effectuer, les données à saisir et les résultats attendus.

## Exemple :

Pour gérer nos tests on prenais une fonctionnalités et on la testais de bout en bout comme le montre l'exemple ci-dessous :



```
using System;
using System.Threading;
using System.Threading.Tasks;
using Hobbify.Dijonons.Interest.Api.Controllers;
using Hobbify.Dijonons.Interest.Application.Common.Models.PlaceController;
using Hobbify.Dijonons.Interest.Application.Picture.Commands;
using Hobbify.Dijonons.Interest.Application.Place.Commands;
using Hobbify.Dijonons.Interest.Application.Place.Queries;
using MediatR;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Moq;
using Xunit;

namespace Hobbify.Dijonons.Interest.Api.Tests.Controllers
{
    public class PlaceControllerTests
    {
        private readonly Mock<IMediator> _mediatorMock;
        private readonly PlaceController _placeController;

        public PlaceControllerTests()
        {
            _mediatorMock = new Mock<IMediator>();
            _placeController = new PlaceController(_mediatorMock.Object);
        }

        [Fact]
        public async Task GetPlaces_ReturnsOkResult()
        {
            // Arrange
            var cancellationToken = CancellationToken.None;
            var query = new GetPlacesQuery();
            var expectedResult = new PlacesDto(); // Replace with the expected result

            _mediatorMock.Setup(m => m.Send(query, cancellationToken))
                .ReturnsAsync(expectedResult);

            // Act
            var result = await _placeController.GetPlaces(cancellationToken);

            // Assert
            Assert.IsType<OkObjectResult>(result);
        }
    }
}
```

Test pour la liste de Places

Lors de l'exécution de chaque scénario de test, nous avons enregistré avec précision les résultats obtenus. Nous avons consigné les erreurs détectées, les bugs identifiés et les comportements inattendus observés. De plus, nous avons noté les performances de l'application, en mesurant par exemple les temps de réponse, les délais de chargement et la consommation des ressources. Ces informations nous ont permis d'évaluer la stabilité, la fiabilité et les performances de l'application après chaque série de tests.

## Conclusion :

En conclusion, les tests effectués tout au long du développement de notre application ont joué un rôle crucial dans l'assurance de sa qualité. Ils ont permis de détecter et de corriger les bugs récurrents, de simuler des comportements insolites des utilisateurs et de s'assurer du bon fonctionnement de toutes les fonctionnalités. Les résultats des tests ont été documentés de manière précise, fournissant ainsi une base solide pour évaluer la performance globale de l'application.