

## Introduction

Ce plan de test vise à vérifier la fonctionnalité de notre système. Nous allons examiner la création, la gestion, l'exécution et le suivi des résultats des tournois. Nous nous concentrons sur des fonctionnalités clés telles que la création d'équipes à partir de joueurs et la mise en place de tournois. En revanche, nous excluons la fonctionnalité permettant de charger des fichiers pour générer des équipes ou de charger des équipes prêtes à concourir. Pour garantir que notre logiciel remplit ses fonctions attendues, nous utiliserons différents types de tests :

Tests Unitaires : Pour évaluer les parties cruciales de la logique de notre code.

Tests d'Intégration : Pour vérifier que l'introduction de nouvelles fonctionnalités n'altère pas le fonctionnement du code existant.

Tests Fonctionnels : Pour vérifier chaque fonctionnalité par rapport à ses spécifications, entre autres.

## Environnement de Test

Étant donné que le projet est encore à l'étape de prototype, les tests seront effectués sur nos propres machines. L'automatisation des tests n'est pas nécessaire à ce stade, et aucun environnement de déploiement n'est prévu.

## Analyse de Risques

En se basant sur le périmètre défini, une première analyse de risques permet d'identifier les menaces potentielles pesant sur la qualité et la fiabilité du programme :

Risque de Défaillance des Tests Unitaires : Il est possible que certains aspects cruciaux du code ne soient pas couverts par les tests unitaires, ce qui pourrait laisser des défauts non détectés dans la logique du programme.

Risque de Conflits d'Intégration : L'ajout de nouvelles fonctionnalités pourrait interférer avec le code existant, engendrant des conflits d'intégration et des bogues imprévus.

Risque de Non-Conformité aux Spécifications : Les tests fonctionnels doivent garantir que chaque fonctionnalité respecte les spécifications établies, mais il existe un risque de

comportement inattendu pouvant conduire à des résultats imprévus.

Risque de Couverture Insuffisante des Tests : Bien que différents types de tests soient prévus, il est possible que certaines parties du programme ne soient pas testées, laissant des vulnérabilités non découvertes.

Risque de Dépendances Externes : Si le programme dépend de bibliothèques tierces ou de services externes, des dysfonctionnements ou des changements de comportement dans ces dépendances pourraient entraîner des problèmes dans notre programme.

Risque de Performance : En fonction de la taille des données et de la complexité des algorithmes, il existe un risque de performances insatisfaisantes, pouvant affecter l'expérience utilisateur ou la capacité du programme à gérer de grandes quantités de données.

Risque de Sécurité : Les fonctionnalités de création, gestion et suivi des tournois pourraient présenter des failles de sécurité telles que l'accès non autorisé aux données des utilisateurs ou la manipulation des résultats des tournois, nécessitant une attention particulière lors des tests.

En conclusion, ces risques doivent être pris en compte lors de la planification et de l'exécution des tests afin de réduire les chances de défauts et d'assurer la qualité du programme. Des mesures appropriées, telles qu'une couverture de test exhaustive, une analyse statique du code et des tests de sécurité, peuvent être mises en place pour atténuer ces risques.