

CAHIER DE RECETTE

Conduite du projet Agora

Version :	0.2
Date :	24/12/2023
Rédigé par :	Michel NASSALANG
Relu par :	Robin SAPIN, Yacine BEN-AHMED, Sid Ahmed BRAHIMI, Anass ELGHARBAOUI, Corentin PILLE, Michel NASSALANG, Mohamed CHERFI

Objectif: Ce document est consacré à la définition des moyens et des procédés mis en œuvre pour assurer la recette du produit logiciel développé.

L'objectif de la recette ou validation est de vérifier que le logiciel est conforme à la spécification référencée au chapitre 2 et qu'il répond aux attentes exprimées dans le cahier des charges initial.

Le cahier de recette recense les objectifs de tests de validation et les moyens nécessaires pour les atteindre. Il précise

- les conditions à satisfaire préalablement à l'exécution des tests,
- les moyens matériels requis (plate-forme de tests),
- la logique de leur déroulement (étapes successives),
- les conditions d'arrêt.

Le document de validation fait partie des documents contractuels du projet, au même titre que les spécifications techniques.

MISES A JOUR

Version	Date	Modifications réalisées
0.1	12/12/2023	Création
0.2	22/01/2024	Modifications après retours sur les documents d'étude du projet de M Abdellah Godard

Table des matières

Table des matières

CAHIER DE RECETTE	1
MISES A JOUR	2
	3
Table des matières	3
I. Introduction	4
II. Documents applicables et de référence	4
III. Terminologie et sigles utilisés	5
IV. Environnement de test	5
V. Responsabilités	5
VI. Stratégie de tests	5
VII. Gestion des anomalies	9
VIII. Procédures de test	10
1. Joueur	10
2. Jeux	11
3. Tables	11
4. Jouer une partie	13
5. Gestion compte	14
6. Communiquer dans un forum	15
7. Consulter le profil d'un joueur	16
8. Revoir une partie	16
9. Modérer les actions des joueurs	17
10. Gestion de l'équipe de modération	18
11. Gestion des données du site	19
IX. Jeux de données de test	20
X. Couverture de test	20

I. Introduction

Dans ce document, nous définissons un cahier de recettes pour le projet Agora afin d'assurer la validation du logiciel auprès du client. Il nous permet d'assurer la conformité entre le produit logiciel et les attentes du client. C'est à ces fins que nous définissons l'ensemble des tests de vérification et de validation des besoins exprimés qui doivent être satisfaites par le logiciel.

La plateforme AGORA comporte différentes fonctionnalités comme la gestion des tables, la gestion des utilisateurs, l'historisation des parties qui doivent être satisfaites.

Nous aurons principalement à tester l'interface jeu plateforme qui permet de bien gérer les jeux de la plateforme. Les fonctionnalités listés dans les spécifications techniques de besoin doivent être testées et validées. La plateforme doit être maintenue stable et améliorée en utilisant plusieurs techniques de validation et en évitant les régressions avec les bogues.

Les tests seront réalisés sur un serveur local avec l'utilisation de plusieurs techniques d'optimisation comme SonarQube pour la détection des vulnérabilités et l'optimisation du code.

Pour mener à bien le projet, il est spécifié que nous devons utiliser le langage de programmation PHP avec le framework Symfony, tout en respectant les principes de gestion de projets abordés dans nos cours. Nous utiliserons Tailwind principalement pour mieux gérer la réalisation du CSS de la plateforme. Nous aurons aussi utilisé Symfony UX pour l'interactivité dans la plateforme. La compréhension des documents rédigées est également nécessaire pour tout membre actif du projet afin d'assurer la conformité du logiciel.

II. Documents applicables et de référence

Les références des documents qui nous permettent de définir ce cahier de recette (CDR) sont la fiche de présentation du projet, les comptes rendus de réunion ainsi que les anciennes spécifications techniques de besoin et anciens cahiers de recette.

Le document présent fait foi pour le cahier de recette de la plateforme AGORA.

- **Fiche de présentation de présentation** : nous fournit la liste des besoins initiaux du client.
- **Compte rendu** : nous informe des prises de décisions, des validations, des modifications apportées ainsi que les attentes du client.
- **Spécification technique de besoin (STB)** : résume le besoin et les fonctionnalités du client pour le projet.

III. Terminologie et sigles utilisés

Glossaire :

- **Partie** : instance d'un jeu sur la plateforme
- **Table** : Salon composé d'un ou plusieurs joueur, lié à une partie d'un jeu en cours ou en attente de joueurs
- **Visiteur** : utilisateur non authentifié sur la plateforme
- **Joueur** : utilisateur authentifié pouvant créer, rejoindre une table
- **Modérateur** : Joueur avec des privilèges, des droits de modération, de gestion du site et des joueurs dans la plateforme.
- **Administrateur** : utilisateur avec la totalité des droits d'administration et de gestion du site
- **Fixtures** : données de test générées par Symfony
- **STLC** : Software Testing Life Cycle

Abréviations :

- **CDR** : cahier de recette
- **STB** : spécification technique de besoins

IV. Environnement de test

Les plateformes utilisées pour les tests sont les terminaux mobiles, tablettes et ordinateurs avec les navigateurs les plus utilisés (Firefox, Chrome, Safari, Edge ...). Nous devons assurer l'accès de la plateforme sur différentes configurations matérielles pouvant être utilisées. Pour les tests fonctionnels, nous aurons à utiliser des fixtures sur Symfony.

- Nous aurons à réaliser des tests du produit logiciel tout d'abord sur un serveur local fourni par Symfony.
- Nous testerons ensuite le produit logiciel sur un serveur de développement de l'université avec une configuration Linux. Il sera préconfiguré avec un serveur Apache, le langage de programmation PHP et une base de données MariaDB.

V. Responsabilités

Tous les membres du projet seront en charge de réaliser les tests afin de prendre en compte toutes les fonctionnalités qui seront développées durant l'avancée du projet. Chaque membre s'occupera d'un typologie de test.

Le bon fonctionnement sera basé sur la validation après les tests sur les fonctionnalités joueurs, administrateur et modérateur et aussi sur l'interface jeu-plateforme. Nous devons dans les premiers temps assurer la validation des tests sur l'interface jeu-plateforme.

Les données de test seront générées avec les fixtures sur Symfony.

Chaque version de la plateforme fournie doit faire l'objet de tests et donc d'une validation avant d'être livrée.

VI. Stratégie de tests

Nous avons élaboré une stratégie de tests basée sur le cycle de vie des tests logiciels STLC. Elle sera globalement centré sur les fonctionnalités joueurs et gestionnaires de la plateforme.

Nous donnons aussi une grande importance à l'interface jeu – plateforme dans les tests.
Tous nos tests suivront ces étapes données dans l'image illustrative suivante.



Et Chaque résultat de test sera noté grâce à une matrice de traçabilité.
Nous utiliserons la matrice de traçabilité qui nous permettra de sécuriser le périmètre des tests grâce au recensement, au suivi des besoins et aux statuts des tests. Elle nous permettra d'assurer le respect des spécifications fonctionnelles.
Cette matrice sera utilisé sous la forme suivante pour respecter l'établissement, l'exécution et la traçabilité des tests de la plateforme.

Total de tous tests inclus Nombre de tests réussis : X tests Nombre de tests échoués : Y tests				Dernier cycle de tests Nombre de tests réussis : X tests Nombre de tests échoués : Y tests				
Numéro Cycle	●	●	●	●	●	●	●	●
Test Exigence	01	02	03	4	05	06
STB-X-Y-Z	■	■	■			■		
STB-X-Y-Z		■	■		■	■	■	
STB-X-Y-Z	■				■	■	■	■
STB-X-Y-Z				■	■	■	■	

Le test Z de l'exigence STB-X-Y est répertorié dans la matrice de traçabilité et toutes les exécutions en rapport avec lui sont notés par le résultat qu'il génère.

Les phases de tests assurant une qualité de conformité des besoins avec le client seront axées sur la performance de nos méthodes et rectifications apportées. Nous déroulerons ces phases de tests avec les 6 types de tests suivants :

- **Les tests unitaires** : ils nous permettent de vérifier le bon comportement de nos méthodes et fonctions. Nous allons utiliser **PHPUnit** pour valider le comportement des méthodes et des fonctions
- **Les tests d'intégration (interface)**: ils nous assurent l'intégration et le fait de bien gérer les jeux dans la plateforme . Ils vérifient la communication et l'interaction entre les composants. L'interface jeu plateforme sera l'objectif de ces tests pour gérer l'interopérabilité entre les données des jeux et la plateforme en plus de l'affichage des jeux dans la plateforme. Nous aurons à utiliser **Symfony Panther** vu que nous coderons la plateforme en symfony.
- **Les test fonctionnels** : ils testent le comportement global de la plateforme. Ils vérifient si la plateforme respecte les spécifications fonctionnelles et répond aux attentes du client. Ils se concentrent que sur les entrées, les sorties et le comportement prévu de la plateforme.Nous pouvons utiliser aussi **Symfony Panther** qui est un framework de tests PHP qui prend en charge les tests fonctionnels.
- **Les tests de non-régression** : ces tests nous permettent de nous assurer que les modifications ou mises à jour sur la plateforme n'ont pas introduit de régressions c'est à dire de nouveaux problèmes par rapport à des fonctionnalités. Nous pourrons utiliser **Behat** pour ce type de test.
- **Les tests techniques** : ces tests nous permettent d'évaluer les aspects techniques tels que la performance, la sécurité et la scalabilité de la plateforme comme le fait d'évaluer la réactivité de la plateforme sous différents charges pendant que plusieurs utilisateurs jouent. Nous pourrons réaliser ces tests avec **Blackfire.io**
- **Les tests de recette client** : également connus sous le nom de tests d'acceptation client nous permette de s'assurer que notre plateforme répond aux besoins et aux attentes du client. Les objectifs de ces tests sont la validation des exigences, la validation des scénarios d'utilisation, d'assurer aussi la qualité de l'expérience utilisateur. Nous pourrons utiliser **Behat** pour réaliser ces tests.

Les critères d'arrêts des tests sont :

- la couverture des exigences données dans le STB
- la stabilité et la fiabilité de la plateforme
- le respect de qualité et validation des tests
- l'approbation du client

Type de Test	Outil	Responsable	Fréquence	Livrable
Tests unitaires	PHPUnit	Anass, Yacine	A chaque développement ou modification de méthodes et fonctions	Rapports de tests unitaires générés par PHPUnit
Tests d'intégration	Symfony Panther	Sid Ahmed	Avant chaque déploiement ou mise à jour	Rapports de tests unitaires générés par Symfony Panther
Tests fonctionnels	Symfony Panther	Robin	A chaque itération ou ajout de nouvelles fonctionnalités	Rapports de tests fonctionnels générés par Symfony Panther
Tests de non régression	Behat	Corentin	Avant chaque déploiement où mise à jour	Rapports de tests de non régression générés par Behat
Tests techniques	Blackfire.io	Mouhamed	A des étapes du développement ou avant mise à jour	Rapports de tests techniques détaillés et générés par Blackfire.io
Tests de recette client	Behat	Michel	A la fin de chaque itération ou avant chaque livraison au client	Rapports de tests de recettes générés par

Les rapports de tests et les documents associés pourront être sauvegardés dans un référentiel dédié.

Les résultats des tests seront pris en compte pour l'ajustement continu de la stratégie qualité et des processus de développement.

VII. Gestion des anomalies

La gestion des anomalies nous permet de s'assurer du bon fonctionnement de la plateforme après chaque modification effectuée. Il est adopté pour éviter les régressions de fonctionnalités. Pour cela, il est nécessaire de:

- Effectuer l'ensemble des stratégies de test nécessaires après une modification.
- Fournir à intervalle régulier un rapport aux membres du groupe décrivant les différents bogues trouvés afin qu'ils puissent corriger le code.
- Faire un test dit de "non régression" sur les nouvelles versions produites pour vérifier que les modifications apportées ont bel et bien corrigé les erreurs et n'en ont pas introduites de nouvelles dans des composants non modifiés.

Lors d'une anomalie, nous devons prendre en considération l'ensemble des erreurs indiquées.

Il faudra essayer de trouver si le problème n'est pas lié à des erreurs de cascade.

Si il devient difficile de corriger une erreur, il faudra demander de l'aide soit au référent technique du projet, soit d'organiser une réunion de toute l'équipe pour procéder ensemble à une éventuelle résolution du problème.

VIII. Procédures de test

1. Joueur

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Vérifier que la création d'un compte répond au fonctionnement demandé (STB-NEW-1)				
Procédure n°2 Créer un compte utilisateur sur la plateforme				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le visiteur clique sur le bouton d'inscription.	Une page contenant un formulaire de création de compte, tel que défini dans la STB, est affichée.	STB-GES-01	
2	Le visiteur valide son compte sans avoir correctement rempli les champs.	La création du compte échoue et il est demandé au visiteur de remplir les champs non ou mal remplis. Le nom d'utilisateur est déjà utilisé, un message l'indique.	Non Respect STB-GES-01	
3	Le visiteur valide son compte en ayant correctement rempli les champs.	Un message indiquant qu'un compte a bien été créé dans la base de données.	Respect STB-GES-01	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : S'authentifier en tant utilisateur (STB-NEW-2)				
Procédure n°3 Authentification d'un utilisateur sur la plateforme				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Un visiteur clique sur le bouton de connexion.	Une page contenant un formulaire de connexion, tel que défini dans la STB est affichée	STB-GES-02	
2	Le visiteur valide ses identifiants sans avoir correctement rempli les champs.	L'authentification échoue et il est demandé au visiteur de remplir les champs non remplis.	Non Respect STB-GES-02	
3	Le visiteur valide ses identifiants en ayant correctement rempli les champs mais ils sont incorrects.	L'authentification échoue et un message d'erreur est affiché.	Non Respect STB-GES-02	
4	Le visiteur valide ses identifiants en ayant correctement rempli les champs et ils sont valides	L'authentification réussie, le visiteur passe au statut de joueur.	Respect STB-GES-02	

2. Jeux

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Voir la liste des jeux (STB-GAME-1) et les règles d'un jeu (STB-GAME-2)				
Procédure n°4 Afficher l'ensemble des jeux et des règles disponibles				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	L'utilisateur clique sur le menu "Jeux"	Le système récupère la liste des jeux disponibles ou prochainement disponibles, et les affiche sur une page	STB-GES-03	
2	L'utilisateur clique sur le cadre d'un jeu	une agrandissement du cadre avec plus d'informations que précédemment	Respect STB-GES-03	
3	L'utilisateur clique sur le bouton d'un jeu	une nouvelle page s'affiche avec un descriptif du jeu.	Respect STB-GES-03	
4	L'utilisateur sélectionne le bouton "Règles" d'un jeu	Un nouvel onglet est ouvert contenant les règles du jeu sélectionné.	Respect STB-GES-03	

3. Tables

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Créer une table (STB-TABLE-1)				
Procédure n°5 Créer une table pour une partie				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur un bouton “Créer une table”	Une page contenant un formulaire s’affiche pour une configuration de table	STB-GES-04	
2	Le joueur valide sa création sans vérifier que ses champs sont bien remplis	La création de la table échoue et il est demandé au joueur de remplir les champs non ou mal remplis.	Non Respect STB-GES-04	
3	Le joueur valide sa table en ayant correctement rempli les champs.	une nouvelle page s’affiche avec un descriptif du jeu.	Respect STB-GES-04	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Rejoindre une table (STB-TABLE-2)				
Procédure n°5 Rejoindre une table créée				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Rejoindre” d’une table	Si la table est encore disponible pour le joueur, le système l’ajoute à la table et le notifie.	Respect STB-GES-05	
2	Le joueur clique sur le bouton “Rejoindre” d’une table indisponible	La table n’est plus disponible, ou la connexion a la table à échoué. On indique donc un message d’erreur selon le cas d’exclusion	Non Respect STB-GES-05	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Inviter un joueur (STB-TABLE-3)				
Procédure n°6 Inviter un joueur dans une table				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton "Inviter des joueurs"	Le système envoie une notification pour chaque joueur invité, et réserve une place à la table pour chacun d'entre eux.	Respect STB-GES-06	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Quitter une table (STB-TABLE-4)				
Procédure n°7 Quitter une table en cours				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Quitter la table”	Le système supprime les informations du joueur de la base de la table, et le notifie qu’il a bien quitté la table.	STB-GES-07	
2	Le dernier joueur clique sur le bouton “Quitter la table”	Le système supprime la table de la base.	Respect STB-GES-07	

4. Jouer une partie

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Lancer une partie (STB-PLAY-1)				
Procédure n°8 Lancer une partie				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Un joueur rejoint et complète la table	Le système charge les informations du jeu, en paramétrant les bases nécessaires. Il envoie ensuite une notification de début de partie aux joueurs.	Respect STB-GES-08	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Parler dans un chat textuel (STB-PLAY-2)				
Procédure n°9 interagir dans un Tchat : écrire et envoyer un message				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur envoie un message dans le chat	Le message est enregistré et affiché dans tous les chats de chaque joueur	Respect STB-GES-09	
2	Le joueur envoie le message avec une erreur	Le message n'est pas enregistré dans la base et un message d'erreur est généré.	Non Respect STB-GES-09	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Exclure un joueur (STB-PLAY-3)				
Procédure n°10 Exclure un joueur inactif d'une table				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Un joueur clique sur le bouton "Exclure" d'un joueur	Exclusion d'un joueur et la table devient disponible pour un nouveau joueur entrant	Respect STB-GES-18	

5. Gestion compte

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Modifier ses informations de compte(STB-ACCOUNT-1)				
Procédure n°11 Modifier les informations de compte utilisateur par son propriétaire				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Modifier mes informations”	La page de modification de compte s’affiche	STB-GES-10	
2	Le joueur remplit les champs d’une manière incorrecte et valide	Le système renvoie sur la page de formulaire avec indication des champs erronés.	Non Respect STB-GES-10	
3	Le joueur remplit les champs d’une manière correcte et valide	Les modifications ont été enregistrées	Respect STB-GES-10	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Ajouter un jeu par un utilisateur dans sa liste de jeux favoris (STB-ACCOUNT-2)				
Procédure n°12 Un joueur ajoute un jeu de la plateforme dans sa liste de jeux favoris				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur un bouton en forme d'étoile	Le système actualise les données du joueur pour ajouter le jeu à sa liste de favoris.	Respect STB-GES-10	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Consulter sa messagerie privée (STB-ACCOUNT-3)				
Procédure n°12 Un joueur consulte sa messagerie privée pour afficher ses messages				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le menu “Ma messagerie”	La messagerie du joueur s’affiche avec une liste des messages sur une page	Respect STB-GES-11	

Objet testé : Plateforme AGORA			Version : 0.1	
Objectif de test : Voir les tables en cours (STB-ACCOUNT-4)				
Procédure n°13 Un joueur affiche les tables dont il est participant				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur entre sur la page d'accueil, ou sa page de profil	La liste des tables en cours du joueur est affiché		

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Voir ses notifications (STB-ACCOUNT-5)				
Procédure n°14 Tester le fait que le joueur consulte bien ses notifications				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Cloche” dans l’entête	Un onglet affichant les notifications du joueurs s’affiche	Respect STB-GES-11	

6. Communiquer dans un forum

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Créer un topic (STB-FORUM-1)				
Procédure n°15 Tenter de créer un topic dans le forum de la plateforme				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Nouveau Topic” en remplissant le formulaire conformément	Le topic est créé	Respect STB-GES-14	
2	Le joueur clique sur le bouton “Nouveau Topic” en ne remplissant pas le formulaire conformément	Le système renvoie sur la page “Forums” et affiche un message d’erreur	Non Respect STB-GES-14	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Envoyer un message dans un topic (STB-FORUM-2)				
Procédure n°16 Produire un message dans un topic bien défini				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton “Envoyer” du formulaire du topic avec une zone de texte remplie	Le message est affiché dans le fil du topic	Respect STB-GES-15	
2	Le joueur clique sur le bouton “Envoyer” du formulaire du topic avec une zone de texte non remplie	Le système renvoie sur la même page avec un message d’erreur	Non Respect STB-GES-15	

7. Consulter le profil d'un joueur

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Envoyer un message privé (STB-PLAYER-2)				
Procédure n°17 Tester l'envoi de message à un joueur en privé				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton "Contacter"	Le message du joueur émetteur est envoyé, et le joueur destinataire est notifié du message	Respect STB-GES-13	

8. Revoir une partie

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Revoir une partie jouée (STB-REPLAY-1)				
Procédure n°18 Visualisation d'une partie jouée (STB-REPLAY-1)				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le joueur clique sur le bouton "Résumé" d'un onglet de partie	Le joueur a accès à un rendu du déroulé de la partie	Respect STB-GES-16	
2	Le joueur remplit l'input du filtre	Une recherche et affichage des parties correspondantes aux entrées du filtre s'opèrent	STB-GES-17	

9. Modérer les actions des joueurs

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Supprimer un topic (STB-MODERATOR-1)				
Procédure n°19 Action supprimer un topic d'un forum par un modérateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le modérateur clique sur le bouton "Supprimer" dans l'en-tête du topic	Le topic est supprimé du forum		

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Supprimer un message d'un topic (STB-MODERATOR-2)				
Procédure n°20 Réaliser une suppression d'un message dans un topic par un modérateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le modérateur clique sur le bouton "Supprimer" du message	Le message est supprimé du forum		

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Bannir un joueur (STB-MODERATOR-4)				
Procédure n°21 Bannissement un joueur (STB-MODERATOR-4)				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le modérateur clique sur le bouton “Bannir” d’un joueur	Le joueur possède le statut “banni” dans la base de données des joueurs du serveur	Respect STB-GES-18	
2	Le modérateur clique sur le bouton “Bannir” d’un joueur	Le joueur est notifié de son bannissement	Respect STB-GES-18	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Supprimer une table (STB-MODERATOR-5)				
Procédure n°22 Suppression de table				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	Le modérateur clique sur le bouton “Supprimer” de la table	La table est supprimé	Respect STB-GES-19	
2	Le modérateur clique sur le bouton “Supprimer” de la table	Les joueurs de la table sont notifiés	Respect STB-GES-19	

10. Gestion de l'équipe de modération

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Promouvoir un joueur au rôle de modérateur (STB-TEAM-1)				
Procédure n°22 Donner le rôle de modérateur à un joueur par l'administrateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	L'administrateur clique sur le bouton "Nommer modérateur" associé à un joueur	Le joueur est devenu un modérateur	Respect STB-GES-20	

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Modifier les droits d'un modérateur (STB-TEAM-2)				
Procédure n°23 Réaliser des modifications des rôles d'un utilisateur par l'administrateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	L'administrateur clique sur le bouton "Modifier les droits"	Les modifications de droits se sont effectuées	Respect STB-GES-21	
2	L'administrateur valide ses modifications	Le modérateur a été notifié	Respect STB-GES-21	

11. Gestion des données du site

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Ajouter un jeu (STB-ADMIN-1)				
Procédure n°24 Rendre disponible un jeu par l'administrateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	L'administrateur clique sur le bouton "Ajouter un jeu"	Le jeu est ajouté et disponible		
2	L'administrateur valide son ajout avec un dossier ne respectant pas le format nécessaire	Le système affiche un message d'erreur		

Objet testé : Plateforme AGORA		Version : 0.1		
Objectif de test : Créer des comptes joueurs temporaires en masse (STB-ADMIN-2)				
Procédure n°25 Tester la création de plusieurs comptes temporaires par l'administrateur				
N°	Actions	Résultats attendus	Exigences	OK/ NOK
1	L'administrateur clique sur le bouton "Créer un compte temporaire"	Les comptes sont créés	Respect STB-GES-22	
2	L'administrateur confirme la création et génère une erreur	Le système affiche un message d'erreur	Non Respect STB-GES-22	

IX. Jeux de données de test

Les données de test seront gérées automatiquement par les fixtures. Nous n'aurons à créer des données pour l'administrateur que lorsqu'on aura terminé la phase de développement où les données ne pourront plus être générées par les fixtures.

X. Couverture de test

Les exigences suivantes sont vérifiées et validées pendant la phase de conception.

ID Exigence STB	Méthode de vérification	Procédures utilisées	Commentaire
STB-OPE-01 STB-OPE-03 STB-OPE-06 STB-INT-01 STB-INT-02 STB-INT-04			
STB-OPE-02 STB-OPE-05 STB-QUA-01 STB-QUA-02			
STB-QUA-03 STB-OPE-04			
STB-QUA-04 STB-INT-03			
STB-REA-01 STB-REA-03 STB-REA-04 STB-REA-06 STB-REA-07 STB-REA-08 STB-REA-09			