Documentation projet

Version 1 du 31 mars 2025

Date de création : 24 mars 2025

Alparslan Ceyhan

|  |  |
| --- | --- |
| Une image contenant texte  Description générée automatiquement | Module du 24.03.2025 au 15.04.2025 |

Table des matières

[1 Introduction et contexte du project 3](#_Toc194300511)

[2 Analyse à faire complètement avec EA -> à rendre uniquement le fichier EA 3](#_Toc194300512)

[2.1 Use case client et use case Rest 3](#_Toc194300513)

[2.1.1 Use case client 1 – Alparslan Ceyhan 3](#_Toc194300514)

[2.1.2 Use Rest 1 – Alparslan Ceyhan 3](#_Toc194300515)

[2.2 Activity Diagram d'un cas complet navigant dans les applications avec les explications 4](#_Toc194300516)

[2.2.1 Activity diagramme client 1 – Alparslan Ceyhan 4](#_Toc194300517)

[2.3 Maquettes du projet avec la navigation intégrée et ses explications 5](#_Toc194300518)

[2.4 Sequence System global entre les applications 6](#_Toc194300519)

[2.4.1 Séquence system client 1 6](#_Toc194300520)

[3 Conception à faire complétement avec EA -> à rendre uniquement le fichier EA 9](#_Toc194300521)

[3.1 Class Diagram complet avec les explications de chaque application 9](#_Toc194300522)

[3.1.1 Class Diagram client 1 – Alparslan Ceyhan 9](#_Toc194300523)

[3.1.2 Class Diagram APIGateway 1 – Alparslan Ceyhan 9](#_Toc194300524)

[3.2 Navigation Diagram complète avec les explications des applications 10](#_Toc194300525)

[3.2.1 Navigation Diagram client 1 10](#_Toc194300526)

[4 Bases de données 11](#_Toc194300527)

[4.1 Modèles ER 11](#_Toc194300528)

[4.2 Modèles WorkBench MySQL 11](#_Toc194300529)

[5 Implémentation des applications <Le client Ap1> et <Le client Ap2> 11](#_Toc194300530)

[5.1 Une descente de code client 11](#_Toc194300531)

[6 Implémentation de l'application <API Gateway> 11](#_Toc194300532)

[6.1 Une descente de code APIGateway 11](#_Toc194300533)

[7 Implémentation des applications <API élève1> et <API élève2> 12](#_Toc194300534)

[7.1 Une descente de code de l'API REST 12](#_Toc194300535)

[8 Hébergement 12](#_Toc194300536)

[9 Installation du projet complet avec les 5 applications 12](#_Toc194300537)

[10 Outils, langages (versions, définitions, installations) 12](#_Toc194300538)

[11 Tests de fonctionnement du projet 12](#_Toc194300539)

[12 Auto-évaluations et conclusions 12](#_Toc194300540)

[12.1 Auto-évaluation et conclusion de Ceyhan Alparslan 12](#_Toc194300541)

[12.2 Auto-évaluation et conclusion de 12](#_Toc194300542)

# Introduction et contexte du project

Ce projet consiste en la création d'une plateforme de gestion de tournois de jeux vidéo en ligne. Son objectif est d'offrir une solution complète pour organiser des compétitions, suivre les performances des joueurs et des équipes, et gérer les classements en temps réel.

# Analyse à faire complètement avec EA -> à rendre uniquement le fichier EA

## Use case client et use case Rest

### Use case client 1 – Alparslan Ceyhan

Une image contenant texte, capture d’écran, diagramme, cercle

Le contenu généré par l’IA peut être incorrect.

### Use case client 2 – Vallélian Thibaud

Une image contenant texte, diagramme, capture d’écran, cercle

Le contenu généré par l’IA peut être incorrect.

### Use Rest 1 – Alparslan Ceyhan

Une image contenant texte, capture d’écran, diagramme, nombre

Le contenu généré par l’IA peut être incorrect.

### Use Rest 2 – Vallélian Thibaud

Une image contenant diagramme, texte, ligne, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## Activity Diagram d'un cas complet navigant dans les applications avec les explications

### Activity diagramme client 1 – Alparslan Ceyhan

1. Début du processus (User & Client)

* L'utilisateur commence en saisissant les informations de l'équipe.
* Le client vérifie si les informations saisies sont valides :
* Si non : Un message d'erreur est affiché et le processus s'arrête.
* Si oui : Les informations sont envoyées à l'API Gateway.

1. Transmission des informations (APIGateway & API Rest)

* L'API Gateway reçoit et transmet les données à l'API Rest.
* L'API Rest vérifie si une équipe avec le même nom existe déjà dans la base de données.

1. Vérification de l'existence de l'équipe (Database)

* La base de données cherche si le nom de l'équipe existe déjà :
* Si oui : Un message d'erreur est retourné à l'utilisateur.
* Si non : L'équipe est ajoutée à la base de données.

1. Confirmation de l'ajout (API Rest & Client)

* Si l'ajout de l'équipe échoue, un message d'erreur est affiché.
* Si l'ajout réussit, un message de confirmation est envoyé.
* L'utilisateur est redirigé vers la page de l'équipe nouvellement créée.

Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

### Activity diagramme client 2 – Vallélian Thibaud

1. Début du processus (User & Client)

* L'utilisateur commence en saisissant les informations du tournois.
* Le client vérifie si les informations saisies sont valides :
* Si non : Un message d'erreur est affiché et le processus s'arrête.
* Si oui : Les informations sont envoyées à l'API Gateway.

1. Transmission des informations (APIGateway & API Rest)

* L'API Gateway reçoit et transmet les données à l'API Rest.
* L'API Rest vérifie si un tournois avec le même nom existe déjà dans la base de données.

1. Vérification de l'existence du tournois (Database)

* La base de données cherche si le nom du tournois existe déjà :
* Si oui : Un message d'erreur est retourné à l'utilisateur.
* Si non : Le tournois est ajouté à la base de données.

1. Confirmation de l'ajout (API Rest & Client)

* Si l'ajout du tournois échoue, un message d'erreur est affiché.
* Si l'ajout réussit, un message de confirmation est envoyé.
* L'utilisateur est redirigé vers la page du tournois nouvellement créé.

Une image contenant texte, diagramme, Parallèle, Plan

Le contenu généré par l’IA peut être incorrect.

## Maquettes du projet avec la navigation intégrée et ses explications

La home page sera la première page, en tant que visiteur nous pouvons soit nous connecter soit voir les tournois et leur équipe et joueur. Si il se connecte en tant que utilisateur il pourra créer une équipe dans un tournois, si il est administrateur il pourra créer des tournois, en supprimer et il peut également supprimer des équipes.

Une image contenant texte, capture d’écran, logiciel, diagramme

Le contenu généré par l’IA peut être incorrect.

## Sequence System global entre les applications

### Séquence system client 1 – Alparslan Ceyhan

1. Vérification des données saisies

* L'utilisateur clique sur "Créer une équipe".
* Le Client vérifie si les données saisies sont correctes.
* Si les données sont incorrectes, un message d’erreur est affiché.

1. Envoi des données pour créer une équipe

* Si les données sont correctes, elles sont envoyées via l’APIGateway à l’API Rest1.
* L’API vérifie si l’équipe existe déjà dans la base de données.

1. Deux cas possibles :

* L’équipe n’existe pas :
* L’API ajoute l’équipe à la base de données.
* La réponse est renvoyée à l’APIGateway, qui la transmet au client.
* L’utilisateur voit la page de l’équipe nouvellement créée.
* L’équipe existe déjà :
* L’API renvoie une erreur indiquant que l’équipe est déjà prise.
* L’APIGateway transmet l’erreur au Client.
* Un message d’erreur s’affiche pour informer l’utilisateur.

Une image contenant texte, diagramme, Rectangle, Parallèle

Le contenu généré par l’IA peut être incorrect.

### Séquence system client 2 - Vallélian Thibaud

1. Vérification des données saisies

* L'admin clique sur "Créer un tournois".
* Le Client vérifie si les données saisies sont correctes.
* Si les données sont incorrectes, un message d’erreur est affiché.

1. Envoi des données pour créer un tournois

* Si les données sont correctes, elles sont envoyées via l’APIGateway à l’API Rest1.
* L’API vérifie si le tournois existe déjà dans la base de données.

1. Deux cas possibles :

* Le tournois n’existe pas :
* L’API ajoute le tournois à la base de données.
* La réponse est renvoyée à l’APIGateway, qui la transmet au client.
* L’utilisateur voit la page du tournois nouvellement créée.
* Le tournois existe déjà :
* L’API renvoie une erreur indiquant que le tournois est déjà créé.
* L’APIGateway transmet l’erreur au client.
* Un message d’erreur s’affiche pour informer l’utilisateur.

Une image contenant texte, diagramme, Parallèle, Rectangle

Le contenu généré par l’IA peut être incorrect.

# Conception à faire complétement avec EA -> à rendre uniquement le fichier EA

## Class Diagram complet avec les explications de chaque application

### Class Diagram client 1 – Alparslan Ceyhan

Lors du lancement de l’application, le fichier index.html est chargé, celui-ci va appeler le tournament.html qui va afficher les tournois, createTeam va s’occuper de créer des équipes pour un tournois spécifique. Le ctrl et le wrk .js vont ensuite faire le lien avec l’APIGateway pour échanger les informations et les données des tournois et des équipes.

Une image contenant texte, capture d’écran, diagramme, Rectangle

Le contenu généré par l’IA peut être incorrect.

### Class Diagram client 2 – Vallélian Thibaud

Lors du lancement de l’application, le fichier index.html est chargé, celui-ci va appeler le tournament.html qui va afficher les tournois, createTournament va s’occuper de créer des nouveaux tournois. Le ctrl et le wrk .js vont ensuite faire le lien avec l’APIGateway pour échanger les informations et les données des tournois et des équipes.

Une image contenant diagramme, Rectangle, texte, ligne

Le contenu généré par l’IA peut être incorrect.

### Class Diagram API Gateway

La structure du diagramme de classe de l’APIGateway se compose d’un worker, avec une interface pour envoyer les informations au controller, qui lui-même à aussi une interface pour communiquer avec le wrk. Le Servlet APIGateway peut lui récupérer les infos du ctrl pour les envoyer au userManager et à l’adminManager.

Une image contenant texte, Post-it, capture d’écran, Rectangle

Le contenu généré par l’IA peut être incorrect.

## Navigation Diagram complète avec les explications des applications

### Navigation Diagram client 1 – Alparslan Ceyhan

Si un utilisateur souhaite créer son équipe, dans un tournoi il doit, d’abord depuis l’index.html sélectionner un tournoi puis cliquer sur créer un tournoi dans la page tournament.html, ensuite il devra remplir les informations nécessaires à la création de son équipe cela sera vérifier si les informations ne sont pas complétées ou qu’une équipe possédant le même nom existe elle ne sera pas créée, si ce n’est pas le cas il sera amené dans la page de son équipe.

Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.

### Navigation Diagram client 2 – Vallélian Thibaud

Quand l’admin souhaite créer un tournoi, il doit, d’abord depuis l’index.html aller vers la pages des tournois, puis cliquer sur créer un tournoi dans la page tournament.html, ensuite il devra remplir les informations nécessaires à la création de son tournois cela sera vérifié et si les informations ne sont pas complètes ou qu’un tournois possédant le même nom celui-ci ne sera pas créé, si ce n’est pas le cas il sera amené dans la page du tournois.

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

# Bases de données

## Modèles ER

* **Partie Admin (Gestion des Tournois)**
  1. Admin - Tournois (1,N) :
  + Un admin peut gérer plusieurs tournois, mais un tournoi est administré par un seul admin.
  1. Tournois – Ranking (1,1) :
  + Chaque tournoi possède un classement unique des équipes.
  1. Tournois – Games (N,1) :
  + Un tournoi est basé sur un seul jeu, mais un jeu peut être utilisé dans plusieurs tournois.
  1. Tournois - Matches (1,N):
  + Un tournoi contient plusieurs matchs, mais un match appartient à un seul tournoi.
  1. Matches - Teams(2,N):
  + Un match à uniquement deux équipes et les équipes peuvent participer à plusieurs matchs
  1. Ranking – Teams (N,1):
  + Un classement possède plusieurs équipes, et une équipe peut apparaitre que dans un seul classement.
  1. Tournois - Teams (1,N):
  + Une équipe appartient à un seul tournoi, mais un tournoi peut avoir plusieurs équipes.
* **Partie Utilisateur (Gestion des Équipes)**
  1. User - Teams\_User (N,N):
  + Un utilisateur peut être membre de plusieurs équipes.
  1. Teams- Teams\_User (N,N):
  + Une équipe peut avoir plusieurs joueurs.
  1. User – Teams (N,1):
  + Un utilisateur peut être capitaine d’une ou plusieurs équipes, et l’équipe peut avoir qu’un seul capitaine.
  1. Teams - Matches (N,2):
  + Une équipe peut participer à plusieurs matchs, et un match inclus deux équipes.
  1. Teams - Tournois (1,N):
  + Une équipe appartient à un seul tournoi, mais un tournoi peut avoir plusieurs équipes.
  1. Teams - Ranking(1,N):
  + Un classement possède plusieurs équipes, et une équipe peut apparaitre que dans un seul classement.



## Modèles WorkBench MySQL

### Client 1

Une image contenant texte, capture d’écran, diagramme, Parallèle

Le contenu généré par l’IA peut être incorrect.

### Client 2

Une image contenant texte, capture d’écran, diagramme, Parallèle

Le contenu généré par l’IA peut être incorrect.

# Implémentation des applications <Le client Ap1> et <Le client Ap2>

## Une descente de code client

# Implémentation de l'application <API Gateway>

## Une descente de code APIGateway

# Implémentation des applications <API élève1> et <API élève2>

## Une descente de code de l'API REST

# Hébergement

# Installation du projet complet avec les 5 applications

# Outils, langages (versions, définitions, installations)

# Tests de fonctionnement du projet

# Auto-évaluations et conclusions

## Auto-évaluation et conclusion de Ceyhan Alparslan

## Auto-évaluation et conclusion de