Qwiktur

Cahier des charges

Sommaire

1. Spécifications générales 3
1.1. Présentation du projet 3
1.2. Règles du jeu 3
1.3. Objectifs 4
1.3.1. Objectif général 4
1.3.2. Objectifs fonctionnels 4
1.3.3. Cibles 4
1.4. Livraison 4
2. Spécifications détaillées 5
2.1. Technologies utilisées 5
2.2. Backend (API) 5
2.3. Frontend (Site web) 5
2.3.1. Front Office 5
2.3.2. Back Office (Admin) 6
2.4. Base de données 6
2.4.1. Entités 6
2.4.2. Modèle conceptuel de données (MCD) 7
3. Annexes 7

1. Spécifications générales

1.1. Présentation du projet

Le projet est de créer un jeu en ligne basé sur la confrontation de plusieurs joueurs sur énigme visuelle. Pour cela, une série de questions sur un thème particulier seront posées aux joueurs. Pour chaque bonne réponse, l'énigme à trouver (image) sera progressivement moins floutée. Dès lors qu'un joueur trouve l'intitulé de l'image, il gagne. Les thèmes seront aléatoires pour que les joueurs puissent avoir un classement global.
Le but du jeu : pour chaque réponse correcte aux questions affichées, l'effet de flou sur l'énigme (image) se dissipera proportionnellement. La partie sera composée de 5 joueurs et le classement de la partie se basera sur le temps des joueurs à résoudre l'énigme. Le premier joueur à résoudre l'énigme gagnera la première place et ainsi de suite.
Il y aura un système de classement des joueurs par Elo, ces points d'Elo sont remportés en partie. Pour remporter un maximum de points d'Elo il faut être le mieux classé en partie. L'Elo des joueurs adverses est pris en compte dans le gain et la perte de points d'Elo.
Une idée de parties personnalisées est également prise en compte, mais cette partie sera réalisée uniquement si le temps nous le permet. Le principe des parties personnalisées est de créer des parties avec ses amis via la génération d'un lien à partager. Les parties personnalisées proposeront l'option du choix de thème mais pourront également susciter l'aléatoire si les joueurs le désirent. Les joueurs ne gagnent pas de points d'Elo en partie personnalisée.

1.2. Règles du jeu

Le but du jeu est d'être le premier à découvrir l'image mystère.

Pour cela :

- Des questions vont s'afficher plus vous répondrez correctement aux questions plus le flou appliqué à l'image se dissipera.
- Répondre à un maximum de questions vous donnera un avantage certains sur vos adversaires.
- Les questions seront en lien avec la thématique de l'image.
- Vous pouvez tenter à tout moment de résoudre l'énigme.
- Plus vos classements en partie sont élevés et plus vos adversaires sont forts plus vous gagnerez de points d'Elo.
- Une partie classée se joue à 5 joueurs.

1.3. Objectifs

1.3.1. Objectif général

L'objectif principal est de monter en Elo en gagnant des parties dans divers domaines de culture générale. Pour cela les joueurs pourront se challenger au classement global et même s'affronter directement entre amis grâce aux parties personnalisées afin de savoir qui est le plus fort en culture générale.

1.3.2. Objectifs fonctionnels

Matchmaking
- Utiliser l'Elo pour trouver des joueurs ayant environs le même niveau.
- Adapter l'algorithme en fonction du nombre d'utilisateurs, surtout au lancement du projet car les utilisateurs seront peu nombreux.
- Système de création de partie automatique et de file d'attente géré au niveau du Backend.

Ranking
- Le ranking sera global sous forme d'Elo.
- L'algorithme de calcul des Elo se fera en fonction :
  - Du classement final des joueurs dans la partie
  - Du nombre d'Elo de chaque joueur

Image
- Utilisation d'une librairie pour modifier des images avec du flou / blur.
- Le traitement sera fait côté Backend pour empêcher les tricheries.
- Toutes les images pour le quizz seront ajoutées par nos soins.

Questions
- API : OpenQuizzDb
- Possibilité de pouvoir ajouter nos propres thèmes et questions en plus de l'API des questions (via le BackOffice).

Afficher les questions dans la langue de l'utilisateur
- Si langue indisponible (exemple : l'utilisateur habite en Syrie), afficher en anglais

Partie personnalisée (si le temps nous le permet)
- Créer une partie avec ses amis
- Génération d'un lien à partager
- Ne fais pas augmenter l'Elo
- Choix du thème ou aléatoire

1.3.3. Cibles

Nos cibles sont les personnes désirant jouer à un jeu axé sur la culture afin de tester ses connaissances ou les confirmer. Le projet est destiné aux joueurs qui aiment la compétition sur les jeux en général et de s'amuser entre amis.

1.4. Livraison

Le projet sera livré au plus tard le Vendredi 18 Décembre 2020.
La livraison comprend à la fois le développement du projet, l'installation et la mise en place du serveur (Backend, Frontend, base de données et nom de domaine), le

remplissage de la base de sonnées (images, thèmes et questions personnalisés), ainsi que la mise en place de tests fonctionnels.

2. Spécifications détaillées

2.1. Technologies utilisées

Les technologies utilisées pour le projet dépendent en fonction de chaque stack. Concernant le Backend, NodeJS est la technologie retenue. En effet, sa robustesse et ses performances permettront de traiter les requêtes API ainsi que les Websockets de façon rapide et efficace.
Côté Frontend, c'est React qui prendra le relais. Cette technologie est en effet réputée pour être réactive grâce à son DOM virtuel malgré sa complexité d'écriture en composants (il faut être très organisé).
Ces deux technologies ont donc été sélectionnées non seulement pour leurs avantages bien connus, mais également pour leur utilisation dans le langage JavaScript, qui donnera un même langage à la fois dans le Backend et dans le Frontend pour davantage de simplicité.
La base de données retenue sera MongoDB, une base orientée document, où les requêtes sont écrites en NoSQL (JavaScript).
Les détails des technologies utilisées seront indiqués ci-après.

2.2. Backend (API)

Additionnellement avec NodeJS qui servira de moteur, plusieurs modules seront ajoutés à celui-ci, à savoir :

- Express, qui permet de créer des routes (URIs) tout en gérant le pattern REST en incluant la gestion des méthodes http (GET, POST, PUT, PATCH et DELETE)
- Mongoose, qui sert d'intermédiaire et de validateur entre l'API et la base de données MongoDB.
- Socket.IO pour le déroulement des parties en temps réel.

2.3. Frontend (Site web)

Le Frontend sera développé à l'aide de la librairie React (orienté hooks), tout en incluant react-router pour la navigation ainsi qu'Axios et Socket.IO pour se connecter au Backend.

2.3.1. Front Office

Le FrontOffice comprend toute la partie du site web utilisable par tout le monde, à savoir :

- La page d'accueil
- La page de matchmaking
- La page de partie en cours
- La page utilisateur (dashboard, paramètres, etc…)

2.3.2. Back Office (Admin)

Le BackOffice comprend à l'inverse la partie administrative du site web. Elle est uniquement accessible par les utilisateurs ayant un rôle spécifique (administrateur et modérateur) leur permettant de gérer les données. Les parties comprenant le BackOffice sont :

- La page de dashboard général (affichage des diverses données, comme les dernières parties jouées, les nouveaux inscrits, les statistiques, etc…)
- La page de gestion des données, à savoir l'ajout d'images, de questions et thèmes personnalisés, etc…
- La page de gestion des utilisateurs

2.4. Base de données

La base de données utilisée est MongoDB. C'est une base écrite en NoSQL. Son interfaçage sera réalisé grâce au module Mongoose côté Backend.
Pour la gestion de la base de données, les outils utilisés seront MongoDB Atlas pour la centralisation, et MongoDB Compass pour gérer les données.

2.4.1. Entités

User {
    id : string
    username : string
    password : string
    role : string
    language : string
    elo : number
    createdAt : Date
    updatedAt : Date
}

Theme {
    id: string
    name: string
    createdAt: Date
    updatedAt: Date
}

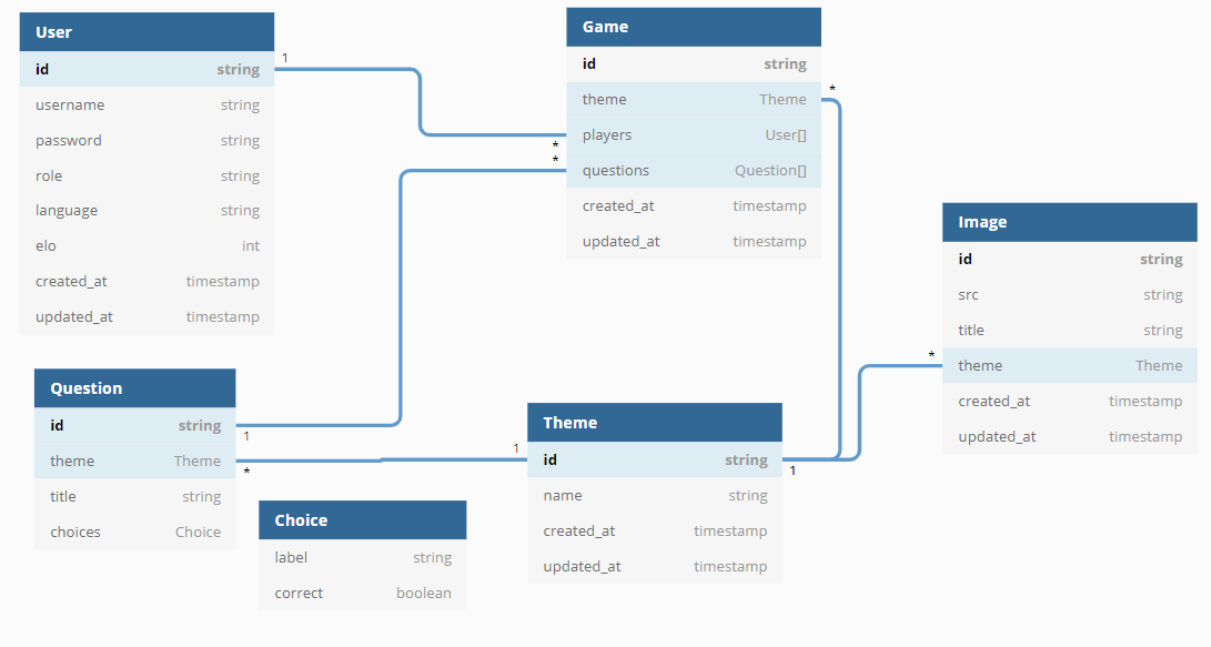
Image {
    id: string
    src: string
    title: string
    theme: Theme
    createdAt: Date
    updatedAt: Date
}

Question {
    id: string
    theme: Theme
    title: string
    choices: {
        label: string
        correct: boolean

```
        }
        createdAt: Date
        updatedAt: Date
}

Game {
        id: string
        theme: Theme
        players: User[]
        questions: [{
                target: Question
                history: [{
                        user: User
                        time: number
                        correct: boolean
                }]
        }]
        createdAt: Date
        updatedAt: Date
}
```

2.4.2. Modèle conceptuel de données (MCD)



3. Annexes

LOGO :


QWIKTUR