# **Spécifications Complètes BrainBattle - MVP1 (Version mise à jour)**

## **1. Vision du projet**

### **1.1 Concept principal**

Plateforme web permettant à des groupes d'amis de participer à des quiz interactifs ensemble à distance, avec un système à trois phases :

* Phase de questions séquentielles
* Phase de correction collaborative dirigée par le créateur
* Phase finale avec révélation des scores et podium animé

### **1.2 Objectifs principaux**

* Créer une expérience sociale amusante et synchrone entre amis
* Proposer une interface simple, intuitive et ludique (thème bleu/noir, animations)
* Implémenter un système varié de questions (QCM, texte, timeline, drag-n-drop)
* Permettre l'enrichissement de la base de données via soumission et validation de questions
* Développer un MVP fonctionnel, optimisé pour desktop

### **1.3 Public cible**

* Groupes d'amis (18-40 ans)
* Potentiellement des collègues pour team buildings (post-MVP)

### **1.4 Contraintes du projet**

* Budget maximal de 100€
* Utilisation de la stack technique existante
* Focalisation sur l'expérience desktop pour le MVP1

## **2. Spécifications fonctionnelles**

### **2.1 Système d'authentification et rôles**

#### **Hiérarchie des rôles**

**Invité**

* Peut rejoindre des parties existantes
* Ne peut pas créer de parties
* Ne peut pas soumettre de questions

**Utilisateur standard**

* Peut créer des parties et inviter des participants
* Peut soumettre des questions à validation
* Peut être désigné comme créateur de partie

**Validateur**

* Tous les droits d'un utilisateur standard
* Peut valider/refuser les questions soumises
* Peut modifier les questions soumises
* Accès à l'interface de validation

**Super-administrateur**

* Tous les droits précédents
* Peut créer/modifier des thèmes
* Peut gérer les utilisateurs et leurs rôles
* Accès complet au panneau d'administration

#### **Processus d'authentification**

* Informations minimales requises lors de l'inscription
* Pas de validation d'email pour MVP1 (prévu pour MVP2)
* Procédure standard de récupération de mot de passe
* Authentification via JWT

### **2.2 Système de questions et réponses**

#### **Types de questions**

* QCM à choix multiples
* Questions à réponse textuelle libre
* Questions avec timeline (barre horizontale pour dates/années)
* Questions de type drag-n-drop
* Support pour médias : images et sons associés aux questions

#### **Système de thèmes**

* Organisation par grandes catégories uniquement (MVP1)
* Création de thèmes réservée aux super-admins
* Option "Autre" temporaire lors de la soumission d'une question
* Minimum de 20 questions par thème pour qu'il soit disponible en partie
* Système de prévention des répétitions (marquage des questions posées dans la journée)

#### **Attributs des questions**

* Difficulté (échelle 1-5)
* Temps de réponse alloué (10-20 secondes)
* Thème associé
* Média optionnel (image ou son)
* Réponse correcte et options (pour QCM)

#### **Scoring**

* Points basés sur la difficulté (échelle 1-5)
* Bonus de rapidité basé sur le chronomètre
* Bonus modéré pour ne pas trop pénaliser les réponses lentes

### **2.3 Soumission et validation des questions**

#### **Processus de soumission**

* Accessible à tous les utilisateurs enregistrés
* Formulaire de création avec tous les attributs nécessaires
* Indication obligatoire du thème (avec option "Autre" temporaire)
* Possibilité d'ajouter des médias

#### **Processus de validation**

* Interface admin spécifique pour les validateurs
* Visualisation de la question, des réponses et de l'auteur
* Options pour accepter/refuser et modifier les questions
* Tolérance pour les erreurs orthographiques mineures dans les réponses
* Questions avec thème "Autre" nécessitant attribution d'un thème valide

### **2.4 Déroulement d'une partie complète**

#### **Création de partie**

* Réservée aux utilisateurs enregistrés
* Sélection des thèmes disponibles (avec au moins 20 questions cumulées)
* Génération d'un code d'invitation unique
* Limite de 10 participants maximum

#### **Phase de questions**

* 20 questions présentées séquentiellement
* Chronomètre spécifique à chaque question
* Affichage des médias associés
* Interface adaptée au type de question
* Réponses et scores cachés pendant cette phase
* Possibilité de se reconnecter (sans gagner de points pendant l'absence)

#### **Phase de correction**

* Correction dirigée par le créateur de la partie
* Affichage séquentiel des questions et réponses de chaque participant
* Système de vote (5 secondes max) pour les réponses litigieuses
* Décision finale appartenant au créateur
* Obligation de passer par toutes les questions
* Transfert automatique du rôle de créateur en cas de déconnexion

#### **Phase finale**

* Affichage des scores du plus bas au plus haut
* Animation ludique pour le podium
* Présentation des rangs et points totaux

#### **Gestion des incidents**

* Reconnexion possible avec conservation des points acquis
* Transfert du rôle de créateur si déconnexion
* Expulsion possible des participants par le créateur
* Impossibilité de rejoindre une partie en cours

## **3. Spécifications techniques**

### **3.1 Architecture technique**

#### **Stack Frontend**

* React.js 18.2.0
* React Router DOM 6.21.3
* Framer Motion 11.0.3 pour les animations
* React Icons 5.0.1
* Tailwind CSS 3.4.1 (sans framework UI additionnel)
* React DnD pour le drag-n-drop
* Testing Library pour les tests

#### **Stack Backend**

* Node.js avec Express.js
* Mongoose pour MongoDB
* JWT pour l'authentification
* Winston pour le logging
* Nodemailer pour les emails (future implémentation)

#### **Base de données**

* MongoDB avec cluster distant
* Modèles pour utilisateurs, questions, parties, thèmes
* Import d'une base de questions externe à déterminer

#### **Sécurité**

* Helmet, CORS, Rate Limiting
* XSS Clean, Express Mongo Sanitize, HPP
* Stockage sécurisé des mots de passe

### **3.2 Communication temps réel**

* WebSockets comme solution prioritaire
* Polling comme solution de backup
* Optimisation pour minimiser l'impact des latences réseau
* Gestion des déconnexions et reconnexions
* Synchronisation des états entre participants

### **3.3 Modèles de données principaux**

#### **Utilisateur**

* ID unique
* Nom d'utilisateur
* Email
* Mot de passe (hashé)
* Rôle (invité, utilisateur, validateur, admin)
* Date de création
* Dernière connexion

#### **Question**

* ID unique
* Type (QCM, texte, timeline, drag-n-drop)
* Contenu
* Réponse(s) correcte(s)
* Options (pour QCM)
* Thème associé
* Difficulté (1-5)
* Temps de réponse alloué
* Média associé (optionnel)
* Statut (en attente, validée, refusée)
* Auteur (référence Utilisateur)
* Validateur (référence Utilisateur)
* Date de création
* Date de dernière modification

#### **Partie**

* ID unique
* Code d'invitation unique
* Créateur (référence Utilisateur)
* Participants (liste de références Utilisateur)
* Thèmes sélectionnés
* Questions sélectionnées
* État de la partie (attente, questions, correction, terminée)
* Scores des participants
* Date de création
* Date de dernière activité

#### **Thème**

* ID unique
* Nom
* Description
* Nombre de questions associées
* Icône/couleur associée
* Date de création
* Créateur (référence Super-admin)

## **4. Stratégie d'implémentation**

### **4.1 Approche de développement**

* Approche UI-first (développement frontal prioritaire)
* Structure basée sur Atomic Design pour les composants
* Développement itératif avec livraison par sprints
* Développement desktop-first (adaptation mobile post-MVP)

### **4.2 Plan de tests**

* Tests unitaires avec Jest et React Testing Library
* Couverture de tests de 80% minimum
* Priorité aux composants UI et logique critique
* Tests d'intégration pour les flux principaux
* Tests manuels avec groupe d'utilisateurs pilotes

### **4.3 Déploiement**

* Frontend sur Vercel ou Render
* Backend sur solution économique compatible (ex: Render, Fly.io)
* MongoDB sur cluster distant (Atlas)
* Infrastructure minimale pour respecter le budget de 100€

### **4.4 Documentation**

* Documentation technique de l'API
* Documentation utilisateur minimale
* Documentation interne du code (JSDoc)

## **5. Évolutions futures (post-MVP)**

### **5.1 Fonctionnalités planifiées**

* Adaptation mobile et tablette
* Validation d'email obligatoire
* Statistiques utilisateurs et historique des parties
* Fonctionnalités sociales avancées
* Chat entre participants
* Génération de questions via IA

### **5.2 Améliorations techniques**

* Optimisation des performances
* CI/CD complet
* Monitoring et analytics
* Système de cache
* Tests end-to-end automatisés