

Plateforme de Suivi Émotionnel pour Enfants Autistes

Application Web MERN Stack

Projet de Développement Web

Année Universitaire 2025-2026

Réalisé par :
Chayma Chetoui

8 janvier 2026

Table des matières

1	Introduction	2
1.1	Contexte du projet	2
1.2	Problématique	2
1.3	Objectifs	2
2	Analyse des Besoins	2
2.1	Besoins Fonctionnels	2
2.1.1	Gestion des Utilisateurs	2
2.1.2	Gestion des Enfants	2
2.1.3	Suivi Émotionnel	2
2.1.4	Analyses et Statistiques	3
2.2	Besoins Non Fonctionnels	3
3	Choix Techniques	3
3.1	Architecture MERN Stack	3
3.1.1	Justification du choix MERN	3
3.2	Technologies Utilisées	3
3.2.1	Frontend - React	3
3.2.2	Backend - Node.js + Express	4
3.2.3	Base de Données - MongoDB	4
4	Compréhension des Concepts MERN	5
4.1	Architecture Complète	5
4.1.1	Gestion d'État	5
5	Justification des Relations de Base de Données	5
5.1	Schéma Relationnel	5
6	Sécurité	5
6.1	Authentification JWT	5
6.2	Validation des Données	5
6.3	Protection des Routes	7
7	API RESTful	7
7.1	Endpoints Principaux	7
8	Tests et Validation	7
8.1	Tests Unitaires	7
8.2	Tests d'Intégration	7
9	Conclusion	7
9.1	Réalisations	7
9.2	Perspectives d'Évolution	8
9.3	Compétences Acquises	8

1 Introduction

1.1 Contexte du projet

Les troubles du spectre autistique (TSA) affectent la communication et le comportement social. Le suivi émotionnel régulier des enfants autistes permet aux parents, thérapeutes et éducateurs de mieux comprendre leurs besoins et d'adapter leurs interventions.

1.2 Problématique

Les outils actuels de suivi émotionnel pour enfants autistes sont souvent :

- Fragmentés et non centralisés
- Difficiles d'utilisation pour les parents
- Manquant d'analyses et de visualisations pertinentes
- Non accessibles en temps réel

1.3 Objectifs

Notre plateforme vise à :

- Centraliser le suivi émotionnel des enfants autistes
- Faciliter l'enregistrement des émotions par différentes sources
- Fournir des analyses statistiques et des visualisations
- Permettre la collaboration entre parents et thérapeutes
- Offrir une interface intuitive et accessible

2 Analyse des Besoins

2.1 Besoins Fonctionnels

2.1.1 Gestion des Utilisateurs

- Inscription et authentification sécurisée
- Gestion de profils (parents, thérapeutes)
- Gestion des sessions avec JWT

2.1.2 Gestion des Enfants

- Ajout et modification de profils d'enfants
- Informations détaillées (âge, diagnostic, préférences)
- Association avec les parents/tuteurs

2.1.3 Suivi Émotionnel

- Enregistrement d'émotions multiples (joie, tristesse, colère, peur, surprise, neutre, dégoût)
- Sources variées : observation parentale, caméra NLP, jeux éducatifs
- Contexte détaillé : lieu, déclencheurs, durée, intensité

2.1.4 Analyses et Statistiques

- Timeline des émotions
- Statistiques par période
- Heatmap des émotions par jour/heure
- Identification des patterns émotionnels

2.2 Besoins Non Fonctionnels

- **Performance** : Temps de réponse < 2 secondes
- **Sécurité** : Chiffrement des données sensibles, authentification robuste
- **Scalabilité** : Architecture modulaire permettant l'ajout de fonctionnalités
- **Accessibilité** : Interface responsive, compatible mobile
- **Fiabilité** : Disponibilité 99%, backup régulier des données

3 Choix Techniques

3.1 Architecture MERN Stack

3.1.1 Justification du choix MERN

Le stack MERN (MongoDB, Express.js, React, Node.js) a été choisi pour plusieurs raisons stratégiques :

Avantages du MERN Stack

1. **JavaScript Full-Stack** : Un seul langage pour frontend et backend
2. **Performance** : Architecture asynchrone et non-bloquante
3. **Communauté** : Large écosystème et support communautaire
4. **Flexibilité** : Adapté aux applications temps réel
5. **Scalabilité** : Architecture microservices facilitée

3.2 Technologies Utilisées

3.2.1 Frontend - React

Choix : React 18 avec Vite

Justifications :

- **Virtual DOM** : Performance optimale pour les mises à jour
- **Component-Based** : Réutilisabilité et maintenabilité
- **Hooks** : Gestion d'état moderne et élégante
- **Ecosystem** : Nombreuses bibliothèques (React Router, React Hot Toast)

Bibliothèques complémentaires :

- `react-router-dom` : Navigation SPA
- `lucide-react` : Icônes modernes

- **react-hot-toast** : Notifications utilisateur
- **recharts** : Visualisations de données

3.2.2 Backend - Node.js + Express

Choix : Node.js v18+ avec Express.js

Justifications :

- **Non-blocking I/O** : Gestion efficace de multiples requêtes
- **NPM** : Plus grand registre de packages open-source
- **Express** : Framework minimaliste et flexible
- **Middleware** : Architecture modulaire et extensible

Middleware utilisés :

- **express-validator** : Validation des données
- **jsonwebtoken** : Authentification JWT
- **bcryptjs** : Hachage sécurisé des mots de passe
- **cors** : Gestion Cross-Origin
- **helmet** : Sécurité HTTP headers

3.2.3 Base de Données - MongoDB

Choix : MongoDB Atlas (Cloud)

Justifications :

- **NoSQL** : Schéma flexible pour données évolutives
- **Document-Oriented** : Correspondance naturelle avec JSON
- **Scalabilité horizontale** : Sharding natif
- **Requêtes riches** : Agrégations complexes supportées
- **Mongoose** : ODM puissant pour validation et relations

4 Compréhension des Concepts MERN

4.1 Architecture Complète



4.1.1 Gestion d'État

- **useState** : État local des composants
- **useEffect** : Effets de bord (API calls, subscriptions)
- **Context API** : État global (authentification)
- **Custom Hooks** : Logique réutilisable

5 Justification des Relations de Base de Données

5.1 Schéma Relationnel

6 Sécurité

6.1 Authentification JWT

```

1 const token = jwt.sign(
2   { userId: user._id, role: user.role },
3   process.env.JWT_SECRET,
4   { expiresIn: '7d' }
5 );
  
```

6.2 Validation des Données

- **express-validator** : Validation côté serveur
- **Mongoose validation** : Contraintes au niveau DB
- **Sanitization** : Protection XSS

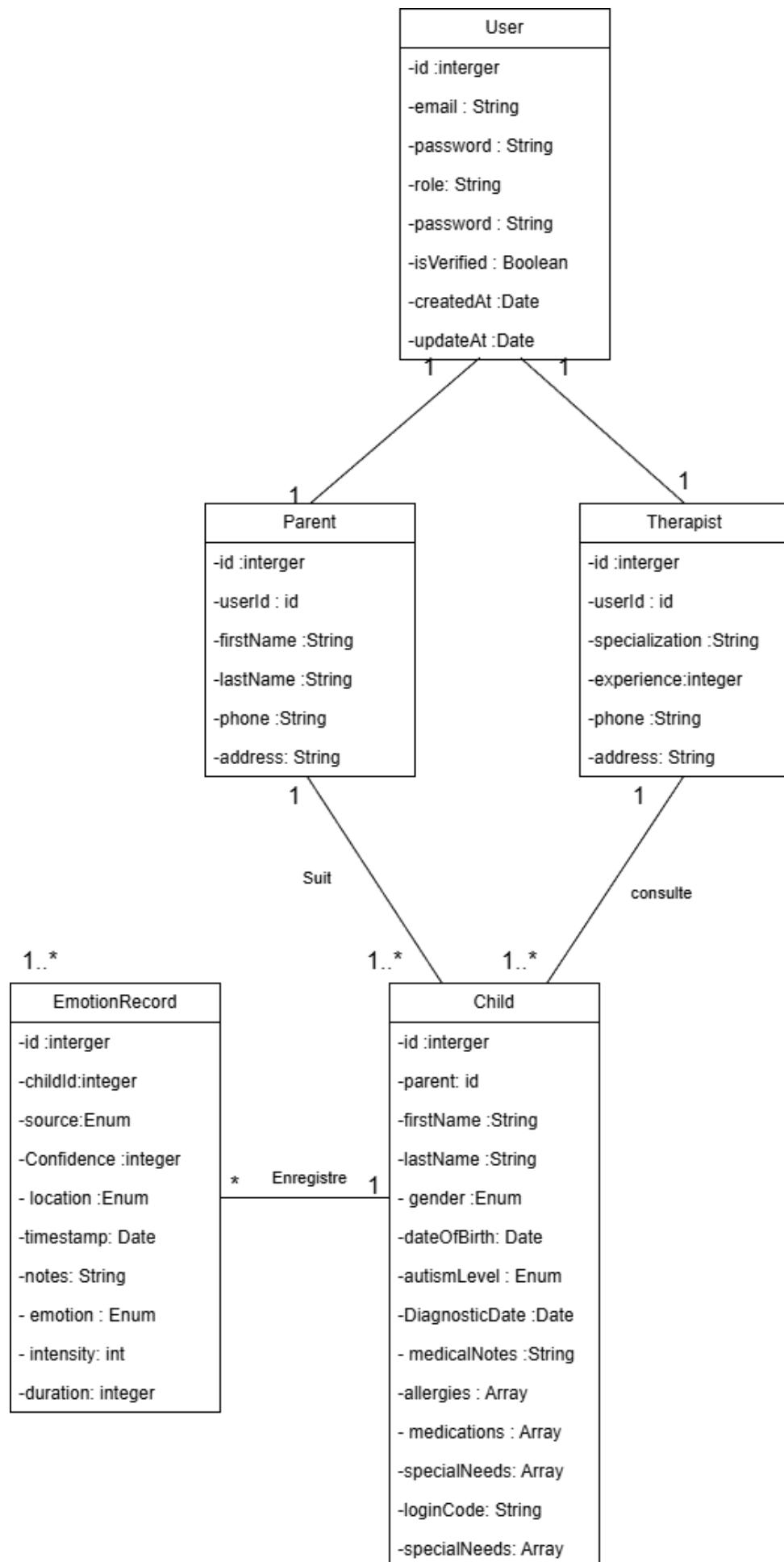


FIGURE 1 – diagramme de classe

6.3 Protection des Routes

- Middleware `authenticate` : Vérification JWT
- Middleware `authorize` : Contrôle des permissions
- Rate limiting : Protection contre bruteforce

7 API RESTful

7.1 Endpoints Principaux

Méthode	Endpoint	Description
POST	/api/auth/register	Inscription
POST	/api/auth/login	Connexion
GET	/api/children	Liste enfants
POST	/api/children	Créer enfant
GET	/api/emotions/child/ :id	Émotions enfant
POST	/api/emotions	Enregistrer émotion
GET	/api/emotions/child/ :id/stats	Statistiques

TABLE 1 – Principaux endpoints API

8 Tests et Validation

8.1 Tests Unitaires

- Controllers : Logique métier
- Services : Interactions DB
- Validators : Règles de validation

8.2 Tests d'Intégration

- Routes API complètes
- Flow authentification
- CRUD opérations

9 Conclusion

9.1 Réalisations

Ce projet a permis de développer une plateforme complète de suivi émotionnel pour enfants autistes, intégrant :

- Intégration IA pour reconnaissance émotionnelle via caméra
- Architecture MERN moderne et scalable
- Interface utilisateur intuitive et responsive

- Base de données optimisée et sécurisée
- API RESTful complète et documentée

9.2 Perspectives d'Évolution

- Application mobile native (React Native)
- Système de recommandations personnalisées
- Export de rapports PDF
- Messagerie entre parents et thérapeutes

9.3 Compétences Acquises

- Maîtrise du stack MERN
- Architecture d'applications web modernes
- Modélisation de bases de données NoSQL
- Sécurisation d'APIs