

# Systeme de feedback sur des produits numeriques

## 1. Analyse du Problème

### Problème actuel

L'entreprise ne dispose d'aucun système structurant pour récolter les avis des utilisateurs sur ses produits numériques. Cela nuit au suivi de la satisfaction client et empêche une amélioration continue fondée sur les retours.

### Objectif

Mettre en place un système centralisé permettant : - De recueillir des notes et commentaires sur chaque produit - De visualiser tous les feedbacks - De filtrer les feedbacks par produit ou utilisateur

## 2. Conception du Système

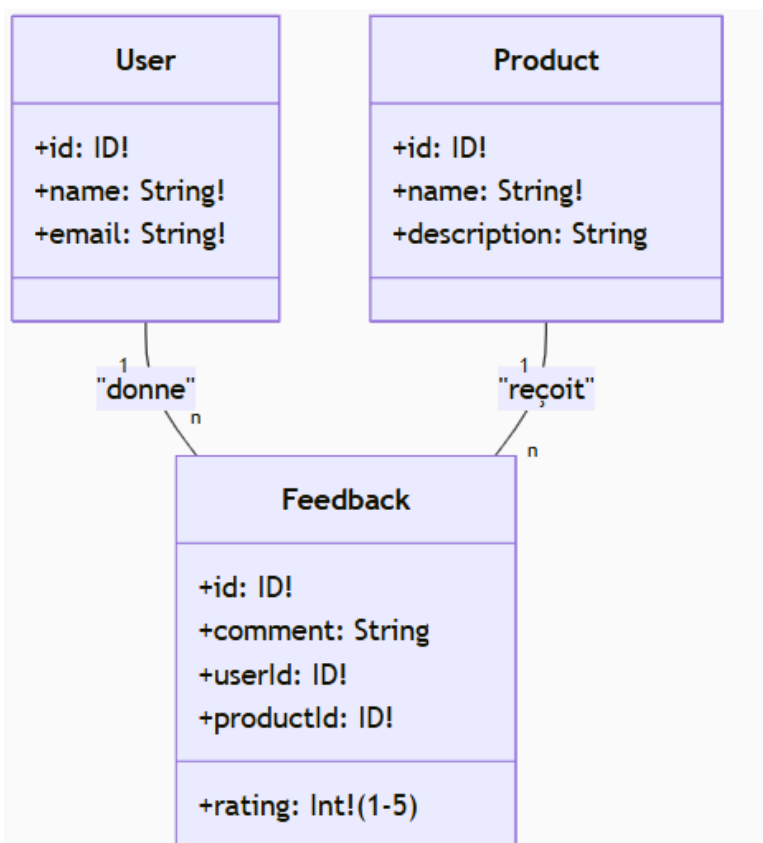
### Entités Principales

- **User** : personne donnant un avis
- **Product** : produit numérique évalué
- **Feedback** : avis d'un utilisateur sur un produit (note + commentaire + date)

### Relations

- Un utilisateur peut donner plusieurs feedbacks
- Un feedback est lié à un seul utilisateur et un seul produit

### Diagramme de classe



### 3. Fonctionnalités Disponibles (via GraphQL)

- Création d'utilisateur
- Création de produit
- Soumission de feedback (note + commentaire)
- Liste de tous les produits
- Liste de tous les feedbacks pour un produit donné
- Liste de tous les feedbacks d'un utilisateur (via champ `feedbacks`)

### 4. Schéma GraphQL (SDL)

```
type User {  
  id: ID!  
  name: String!  
  email: String!  
  feedbacks: [Feedback]  
}  
  
type Product {  
  id: ID!  
  name: String!  
  description: String  
  feedbacks: [Feedback]  
}  
  
type Feedback {  
  id: ID!  
  rating: Int!  
  comment: String  
  date: String!  
  user: User!  
  product: Product!  
}  
  
type Query {  
  getAllProducts: [Product]  
  getFeedbackByProduct(productId: ID!): [Feedback]  
}  
  
type Mutation {  
  createUser(name: String!, email: String!): User  
  createProduct(name: String!, description: String): Product  
  submitFeedback(userId: ID!, productId: ID!, rating: Int!, comment: String): Feedback  
}
```

## 5. Exemples de Requêtes et Réponses

### 1. Création d'un utilisateur

Requête :

```
mutation {  
  createUser(name: "Ali", email: "ali@example.com") {  
    id  
    name  
    email  
  }  
}
```

Réponse :

```
{  
  "data": {  
    "createUser": {  
      "id": "72ab1e4f-ed94-4d88-8b30-d941bad535db",  
      "name": "Ali",  
      "email": "ali@example.com"  
    }  
  }  
}
```

### 2. Création d'un produit

Requête :

```
mutation {  
  createProduct(name: "Application mobile", description: "Application de  
notes") {  
    id  
    name  
    description  
  }  
}
```

Réponse :

```
{  
  "data": {  
    "createProduct": {  
      "id": "279c19a6-df25-4907-84ee-593203b53e0a",  
      "name": "Application mobile",  
      "description": "Application de notes"  
    }  
  }  
}
```

### 3. Soumettre un feedback

Requête :

```
mutation {
  submitFeedback(
    userId: "72ab1e4f-ed94-4d88-8b30-d941bad535db",
    productId: "279c19a6-df25-4907-84ee-593203b53e0a",
    rating: 5,
    comment: "Très utile !"
  ) {
    id
    rating
    comment
    date
    user {
      name
    }
    product {
      name
    }
  }
}
```

Réponse :

```
{
  "data": {
    "submitFeedback": {
      "id": "5dc4f6a7-de81-4fb1-9d28-5ec01e5c7b02",
      "rating": 5,
      "comment": "Très utile !",
      "date": "2025-06-08T20:36:57.486Z",
      "user": {
        "name": "Ali"
      },
      "product": {
        "name": "Application mobile"
      }
    }
  }
}
```

## 4. Lister les produits

Requête :

```
query {  
  getAllProducts {  
    id  
    name  
    description  
  }  
}
```

Réponse :

```
{  
  "data": {  
    "getAllProducts": [  
      {  
        "id": "7b9db5e6-0dc5-4bd1-9b25-e0dc4a8cc3e7",  
        "name": "Application mobile",  
        "description": "Application de notes"  
      }  
    ]  
  }  
}
```

## 5. Lister les feedbacks d'un produit

Requête :

```
query {  
  getFeedbackByProduct(productId: "279c19a6-df25-4907-84ee-593203b53e0a") {  
    id  
    rating  
    comment  
    user {  
      name  
    }  
  }  
}
```

Réponse :

```
{  
  "data": {  
    "getFeedbackByProduct": [  
      {  
        "id": "5dc4f6a7-de81-4fb1-9d28-5ec01e5c7b02",  
        "rating": 5,  
        "comment": "Très utile !",  
        "user": {  
          "name": "Ali"  
        }  
      }  
    ]  
  }  
}
```

## 6. Implémentation Technique

- Utilisation de **Node.js** avec **Apollo Server**
- Données stockées en mémoire (tableaux `users`, `products`, `feedbacks`)
- Identifiants générés avec `uuid`
- Résolution des relations entre objets dans les `resolvers`

## 7. Conclusion

Ce projet démontre une solution fonctionnelle pour la collecte, la visualisation et la gestion des retours utilisateurs. Le service GraphQL est souple, extensible et permet de structurer efficacement les échanges entre utilisateurs et l'entreprise.