

# Rapport de Projet Annuel

---



## NutriScanPro App

---

*Réaliser par :*

EL MEHDI EL AINE, NINO SAADI, ROBIN THOMAS

Année universitaire 2024/2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Contexte et Objectifs . . . . .	5
1.2	Problématique . . . . .	5
1.3	Solutions Apportées . . . . .	5
<b>2</b>	<b>Cahier des Charges Techniques et Fonctionnelles</b>	<b>6</b>
2.1	Objectif du Projet . . . . .	6
2.2	Contraintes Techniques . . . . .	6
2.3	Fonctionnalités attendues . . . . .	6
<b>3</b>	<b>Architecture Générale du Système</b>	<b>6</b>
3.1	Vue d'Ensemble . . . . .	6
3.2	Technologies Utilisées . . . . .	7
3.2.1	Stack Technique Backend . . . . .	7
3.2.2	Stack Technique Frontend . . . . .	7
<b>4</b>	<b>Architecture Backend - API NestJS</b>	<b>7</b>
4.1	Structure Modulaire . . . . .	7
4.2	Schéma de Base de Données . . . . .	8
4.3	Module d'Analyse IA . . . . .	9
4.4	Sécurité et Authentification . . . . .	9
<b>5</b>	<b>Architecture Frontend - Application Mobile</b>	<b>9</b>
5.1	Structure de Navigation . . . . .	9
5.2	Services et Hooks . . . . .	10
5.3	Internationalisation (i18n) . . . . .	11
<b>6</b>	<b>Fonctionnalités Principales</b>	<b>11</b>
6.1	Reconnaissance d'Images par IA . . . . .	11
6.2	Scan de Codes-barres . . . . .	11
6.3	Analyse Textuelle . . . . .	12
6.4	Suivi des Exercices . . . . .	12
<b>7</b>	<b>Interface Utilisateur et Expérience</b>	<b>12</b>
7.1	Design System . . . . .	12
7.2	Écrans Principaux . . . . .	12
7.2.1	Écran d'Accueil . . . . .	12
7.2.2	Historique des Repas . . . . .	13
7.2.3	Profil Utilisateur . . . . .	13
<b>8</b>	<b>Sécurité et Performance</b>	<b>13</b>
8.1	Sécurité . . . . .	13
8.2	Performance . . . . .	13
<b>9</b>	<b>Tests et Qualité</b>	<b>13</b>
9.1	Stratégie de Tests . . . . .	13
9.2	Qualité du Code . . . . .	14

<b>10 Déploiement et Infrastructure</b>	<b>14</b>
10.1 Environnements . . . . .	14
10.2 CI/CD . . . . .	14
<b>11 Évolutions Futures</b>	<b>14</b>
11.1 Fonctionnalités Planifiées . . . . .	14
11.2 Améliorations Techniques . . . . .	14
<b>12 Retours d'Expérience</b>	<b>15</b>
12.1 Points Positifs . . . . .	15
12.2 Difficultés Rencontrées . . . . .	15
12.3 Montée en Compétences de l'Équipe . . . . .	15
<b>13 Conclusion</b>	<b>15</b>
13.1 Points Forts . . . . .	15
13.2 Impact . . . . .	15
13.3 Perspectives . . . . .	16
<b>A Annexe A : Captures d'Écran</b>	<b>16</b>
A.1 Authentification et Page d'accueil . . . . .	16
A.2 Ajout de repas . . . . .	17
A.3 Résultats de l'analyse . . . . .	18
A.4 Bibliothèque Exercice Histoire . . . . .	19
A.5 Profil et sélection de la langue . . . . .	20

## Table des figures

1	Architecture générale du système NutriScanPro . . . . .	7
2	Architecture modulaire de l'API NestJS . . . . .	8
3	Schéma de base de données Prisma . . . . .	8
4	Structure de navigation de l'application mobile . . . . .	10
5	Processus de reconnaissance d'images par IA . . . . .	11
6	Captures d'Écran Authentification et Page d'accueil . . . . .	16
7	Captures d'Écran Choix de type d'ajout de repas . . . . .	17
8	Captures d'Écran Résultats de l'analyse . . . . .	18
9	Captures d'Écran Bibliothèque Exercice Histoire . . . . .	19
10	Captures d'Écran Profil et sélection de la langue . . . . .	20

## Résumé

Ce rapport technique présente l'architecture, l'implémentation et les fonctionnalités de NutriScanPro, une application mobile innovante de suivi nutritionnel. L'application combine des technologies modernes (React Native, Expo, NestJS) avec l'intelligence artificielle (Google Gemini) pour offrir une expérience utilisateur exceptionnelle. Le système comprend une API backend robuste construite avec NestJS et une application mobile cross-platform développée avec Expo. Les fonctionnalités principales incluent la reconnaissance d'aliments par image, le scan de codes-barres, l'analyse textuelle et un suivi complet de la nutrition et des exercices.

# 1 Introduction

## 1.1 Contexte et Objectifs

NutriScanPro est une application mobile révolutionnaire qui transforme la façon dont les utilisateurs suivent leur nutrition. L'objectif principal est de simplifier le processus de suivi alimentaire en utilisant l'intelligence artificielle pour automatiser la reconnaissance et l'analyse des aliments.

## 1.2 Problématique

Le suivi nutritionnel traditionnel présente plusieurs défis :

- Saisie manuelle fastidieuse des aliments
- Difficulté à estimer les portions
- Base de données limitée d'aliments
- Interface utilisateur complexe
- Manque de précision dans l'analyse nutritionnelle

## 1.3 Solutions Apportées

NutriScanPro résout ces problèmes grâce à :

- Reconnaissance d'images par IA (Google Gemini)
- Scan de codes-barres intégré
- Analyse textuelle intelligente
- Interface utilisateur intuitive
- Base de données complète d'aliments
- Suivi multi-plateforme

## 2 Cahier des Charges Techniques et Fonctionnelles

### 2.1 Objectif du Projet

Développer une application mobile multiplateforme permettant à un utilisateur de :

- Suivre ses repas et son activité physique
- Ajouter des aliments par scan de code-barres, photo ou description textuelle
- Visualiser ses apports nutritionnels et son historique
- Définir et suivre des objectifs personnalisés
- Accéder à une bibliothèque d'exercices
- Utiliser l'application en plusieurs langues (français, anglais)

### 2.2 Contraintes Techniques

- Application mobile compatible iOS, Android et Web
- Backend sécurisé, scalable, documenté (Swagger)
- Base de données relationnelle (MySQL)
- Authentification sécurisée (JWT)
- Intégration d'une IA pour l'analyse nutritionnelle
- Respect des bonnes pratiques UX/UI

### 2.3 Fonctionnalités attendues

- Inscription, connexion, gestion de profil
- Ajout de repas (scan, photo, texte)
- Analyse nutritionnelle automatisée
- Suivi des objectifs et de la progression
- Historique détaillé
- Bibliothèque d'exercices
- Internationalisation

## 3 Architecture Générale du Système

### 3.1 Vue d'Ensemble

L'architecture de NutriScanPro suit le modèle client-serveur avec une séparation claire des responsabilités :

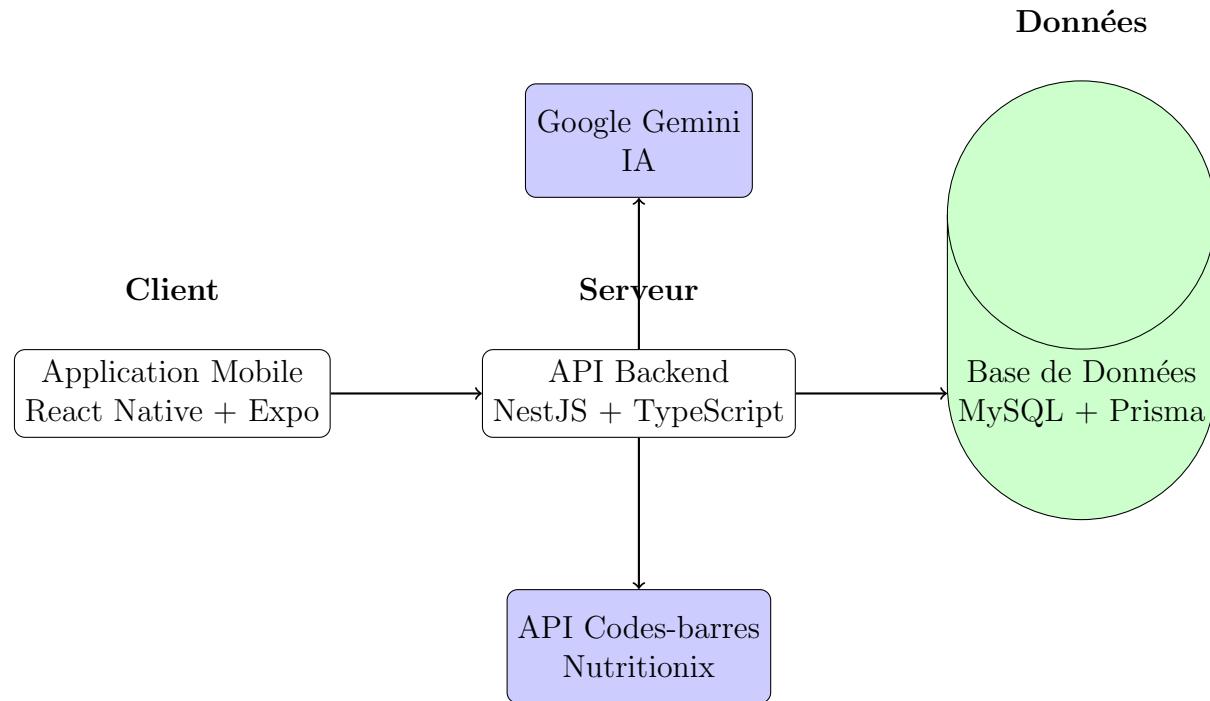


FIGURE 1 – Architecture générale du système NutriScanPro

### 3.2 Technologies Utilisées

#### 3.2.1 Stack Technique Backend

- **Framework :** NestJS 10.0.0
- **Langage :** TypeScript 5.2.2
- **Base de données :** MySQL avec Prisma ORM
- **Authentification :** JWT avec Passport
- **Documentation :** Swagger/OpenAPI
- **IA :** Google Gemini API

#### 3.2.2 Stack Technique Frontend

- **Framework :** React Native avec Expo SDK 53
- **Navigation :** Expo Router 5.1.2
- **Internationalisation :** i18next
- **Caméra :** Expo Camera + Vision Camera
- **Scan Codes-barres :** Expo Barcode Scanner
- **Stockage :** Expo Secure Store

## 4 Architecture Backend - API NestJS

### 4.1 Structure Modulaire

L'API NestJS suit une architecture modulaire avec une séparation claire des responsabilités :

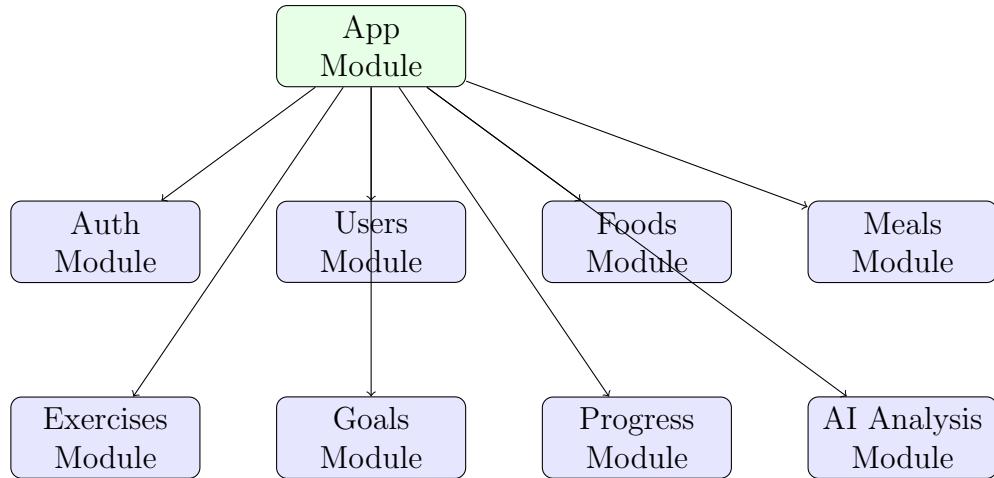


FIGURE 2 – Architecture modulaire de l'API NestJS

## 4.2 Schéma de Base de Données

Le schéma de base de données utilise Prisma ORM avec les entités suivantes :

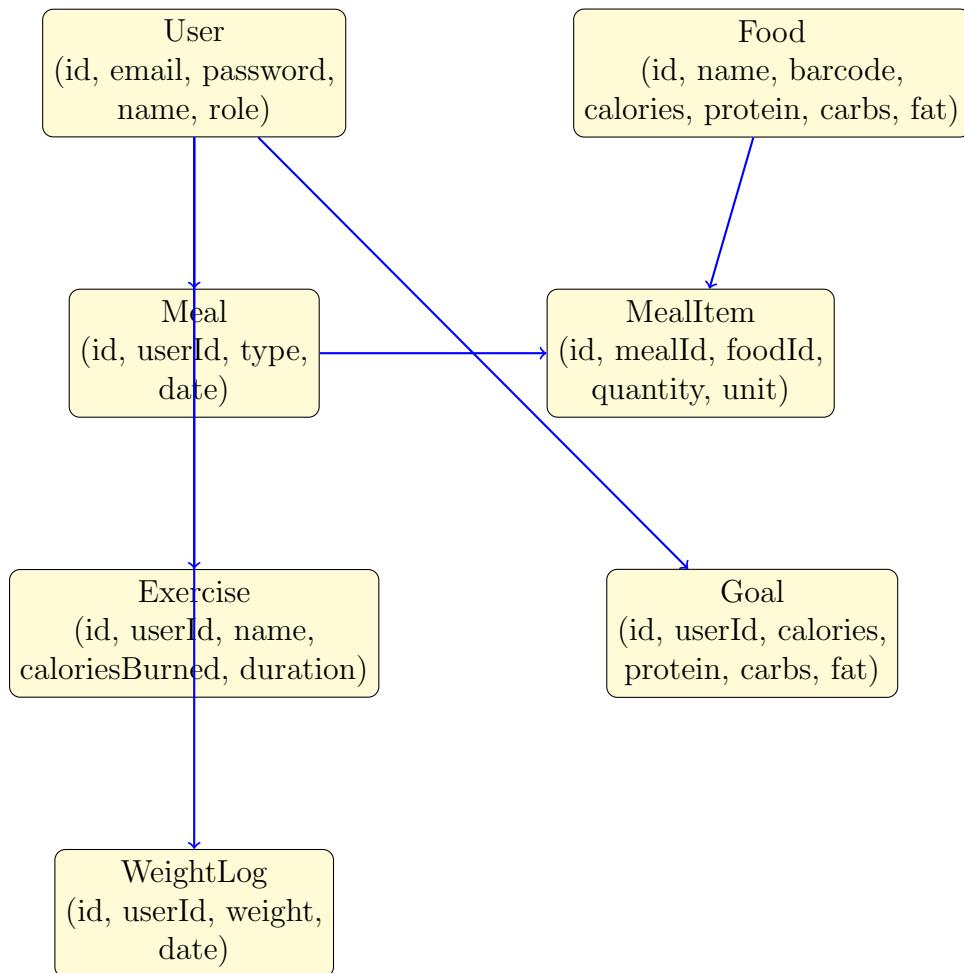


FIGURE 3 – Schéma de base de données Prisma

## 4.3 Module d'Analyse IA

Le module d'analyse IA utilise Google Gemini pour l'analyse d'images et de texte :

```

1  @Injectable()
2  export class AiAnalysisService {
3      private readonly geminiApiKey: string;
4
5      constructor(private readonly configService: ConfigService) {
6          this.geminiApiKey = this.configService.get<string>('GEMINI_API_KEY')
7          ;
8      }
9
10     async analyzeImage(base64Image: string): Promise<NutritionResponse> {
11         const prompt =
12             'Tu es un expert en nutrition.    partir de l'image fournie,
13             identifie les aliments visibles et estime leurs valeurs
14             nutritionnelles...
15             ';
16
17         const payload = {
18             contents: [
19                 parts: [
20                     { text: prompt },
21                     {
22                         inline_data: {
23                             mime_type: 'image/jpeg',
24                             data: base64Image,
25                         },
26                     ],
27                 ],
28             ];
29         }
30
31         return this.callGeminiApi('gemini-2.0-flash', payload);
32     }
33 }
```

Listing 1 – Service d'analyse IA

## 4.4 Sécurité et Authentification

L'API implémente une authentification JWT robuste :

- **Stratégie JWT** : Validation des tokens d'accès
- **Stratégie Local** : Authentification par email/mot de passe
- **Guards** : Protection des routes sensibles
- **Validation** : DTOs avec class-validator
- **Chiffrement** : Mots de passe hashés avec bcrypt

# 5 Architecture Frontend - Application Mobile

## 5.1 Structure de Navigation

L'application utilise Expo Router pour une navigation fluide :

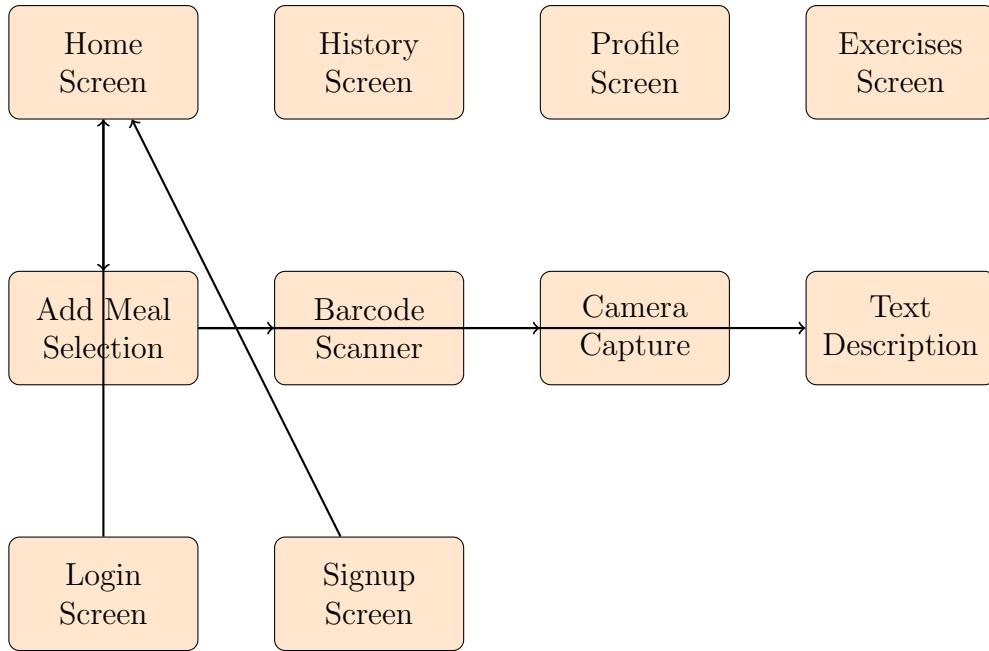


FIGURE 4 – Structure de navigation de l’application mobile

## 5.2 Services et Hooks

L’application utilise des services spécialisés et des hooks React :

```

1 class ApiService {
2   private token: string | null = null;
3
4   async makeRequest<T>(endpoint: string, options: RequestInit = {}): Promise<T> {
5     const url = `${API_BASE_URL}${endpoint}`;
6     const headers: { [key: string]: string } = {};
7
8     if (this.token) {
9       headers['Authorization'] = `Bearer ${this.token}`;
10    }
11
12    const response = await fetch(url, {
13      ...options,
14      headers: { ...headers, ...(options.headers as any) },
15    });
16
17    if (response.status === 401) {
18      this.clearToken();
19      router.push({ pathname: '/auth/login' });
20    }
21
22    return await response.json();
23  }
24 }
  
```

Listing 2 – Service API principal

### 5.3 Internationalisation (i18n)

L'application supporte plusieurs langues avec i18next :

```

1 export const i18nService = {
2   initialize: () => {
3     i18n
4       .use(initReactI18next)
5       .init({
6         resources: {
7           en: { translation: enTranslations },
8           fr: { translation: frTranslations },
9         },
10        lng: getLocales()[0]?.languageCode || 'en',
11        fallbackLng: 'en',
12        interpolation: { escapeValue: false },
13      });
14    },
15  };

```

Listing 3 – Configuration i18n

## 6 Fonctionnalités Principales

### 6.1 Reconnaissance d'Images par IA

La fonctionnalité phare de NutriScanPro utilise Google Gemini pour analyser les photos de repas :

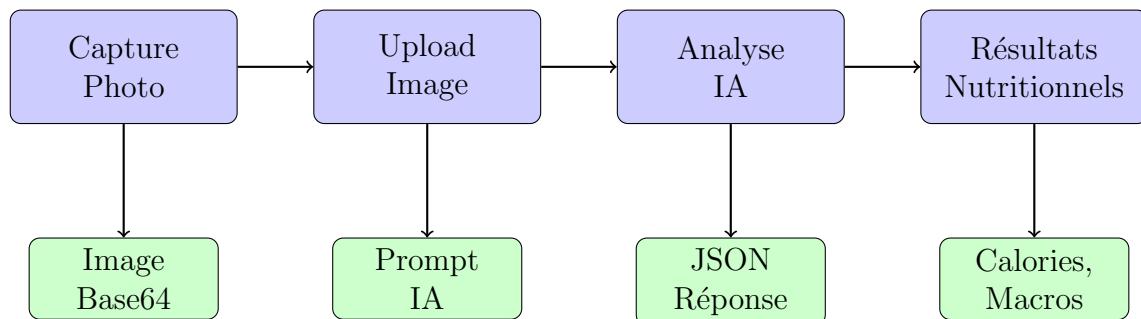


FIGURE 5 – Processus de reconnaissance d'images par IA

### 6.2 Scan de Codes-barres

Intégration avec les APIs de codes-barres pour une identification rapide :

- **Scanner natif** : Utilisation de la caméra du dispositif
- **Base de données** : Recherche dans Nutritionix
- **Informations complètes** : Calories, macronutriments, portions
- **Ajout automatique** : Intégration directe dans les repas

## 6.3 Analyse Textuelle

Analyse intelligente des descriptions de repas :

```

1  async analyzeText(description: string): Promise<NutritionResponse> {
2    const prompt = `
3      Tu es un nutritionniste expert. Voici une description d'un repas :
4      "${description}"
5
6      Estime les informations nutritionnelles pour chaque ingrédient...
7    `;
8
9    const response = await this.callGeminiApi('gemini-2.0-flash', {
10      contents: [{ parts: [{ text: prompt }] }], 
11    });
12
13    return this.extractJson(response);
14 }
```

Listing 4 – Analyse textuelle

## 6.4 Suivi des Exercices

Bibliothèque complète d'exercices avec filtres et détails :

- **Catalogue d'exercices** : Plus de 1000 exercices
- **Filtres avancés** : Par groupe musculaire, équipement
- **Détails complets** : Images, instructions, calories brûlées
- **Recherche intelligente** : Interface de recherche intuitive

# 7 Interface Utilisateur et Expérience

## 7.1 Design System

L'application utilise un design system cohérent :

- **Palette de couleurs** : Couleurs modernes et accessibles
- **Typographie** : Hiérarchie claire des textes
- **Composants** : Boutons, inputs, cartes réutilisables
- **Animations** : Transitions fluides et feedback haptique

## 7.2 Écrans Principaux

### 7.2.1 Écran d'Accueil

- Vue d'ensemble quotidienne des calories
- Progression des objectifs nutritionnels
- Accès rapide aux fonctionnalités principales
- Graphiques de tendances

### 7.2.2 Historique des Repas

- Navigation par date
- Détails complets des repas
- Statistiques nutritionnelles
- Filtres et recherche

### 7.2.3 Profil Utilisateur

- Gestion des informations personnelles
- Configuration des objectifs
- Paramètres de langue
- Statistiques personnelles

## 8 Sécurité et Performance

### 8.1 Sécurité

- **Authentification JWT** : Tokens sécurisés avec expiration
- **Validation des données** : Sanitisation des entrées utilisateur
- **HTTPS** : Communication chiffrée
- **Stockage sécurisé** : Expo Secure Store pour les tokens
- **Gestion des erreurs** : Messages d'erreur sécurisés

### 8.2 Performance

- **Optimisation des images** : Compression et cache
- **Lazy loading** : Chargement à la demande
- **Mise en cache** : Données fréquemment utilisées
- **Optimisation réseau** : Requêtes optimisées
- **Feedback utilisateur** : Indicateurs de chargement

## 9 Tests et Qualité

### 9.1 Stratégie de Tests

- **Tests unitaires** : Jest pour les services
- **Tests d'intégration** : API endpoints
- **Tests E2E** : Flux utilisateur complets
- **Tests de performance** : Temps de réponse

## 9.2 Qualité du Code

- **TypeScript** : Typage strict
- **ESLint** : Linting automatique
- **Prettier** : Formatage du code
- **Documentation** : JSDoc et Swagger

# 10 Déploiement et Infrastructure

## 10.1 Environnements

- **Développement** : Serveur local avec hot reload
- **Staging** : Environnement de test
- **Production** : Serveur cloud avec monitoring

## 10.2 CI/CD

- **Intégration continue** : Tests automatiques
- **Déploiement automatique** : Pipeline GitHub Actions
- **Monitoring** : Logs et métriques
- **Backup** : Sauvegarde automatique de la base de données

# 11 Évolutions Futures

## 11.1 Fonctionnalités Planifiées

- **Reconnaissance vocale** : Saisie par commande vocale
- **IA prédictive** : Suggestions de repas personnalisées
- **Intégration wearables** : Synchronisation avec montres connectées
- **Réseau social** : Partage et comparaison avec d'autres utilisateurs
- **Planification de repas** : Suggestions de menus hebdomadaires

## 11.2 Améliorations Techniques

- **Offline mode** : Fonctionnement hors ligne
- **Push notifications** : Rappels et notifications
- **Analytics avancés** : Statistiques détaillées
- **API publique** : Documentation pour développeurs tiers

## 12 Retours d'Expérience

### 12.1 Points Positifs

- Intégration fluide de l'IA pour l'analyse nutritionnelle
- Application réellement multiplateforme
- Interface moderne et intuitive
- Documentation API complète (Swagger)

### 12.2 Difficultés Rencontrées

- Gestion des dépendances natives sur Expo (caméra, scan)
- Précision de l'IA sur certains plats complexes
- Synchronisation des données entre mobile et backend

### 12.3 Montée en Compétences de l'Équipe

- Maîtrise de NestJS et Prisma pour le backend
- Approfondissement de React Native et Expo
- Découverte de l'intégration d'API IA (Google Gemini)
- Pratique avancée de l'internationalisation (i18next)
- Mise en place de pipelines CI/CD

## 13 Conclusion

NutriScanPro représente une innovation majeure dans le domaine du suivi nutritionnel. L'application combine avec succès les technologies modernes (React Native, NestJS, IA) pour offrir une expérience utilisateur exceptionnelle.

### 13.1 Points Forts

- Architecture modulaire et évolutive
- Intégration IA avancée pour la reconnaissance d'aliments
- Interface utilisateur intuitive et moderne
- Support multi-plateforme complet
- Sécurité et performance optimisées

### 13.2 Impact

L'application simplifie considérablement le suivi nutritionnel en automatisant les tâches fastidieuses et en offrant une précision accrue grâce à l'IA. Elle contribue à améliorer la santé des utilisateurs en facilitant l'adoption d'habitudes alimentaires saines.

### 13.3 Perspectives

Avec ses fonctionnalités innovantes et son architecture robuste, NutriScanPro est bien positionnée pour devenir une référence dans le domaine des applications de santé et de nutrition.

## A Annexe A : Captures d'Écran

### A.1 Authentification et Page d'accueil

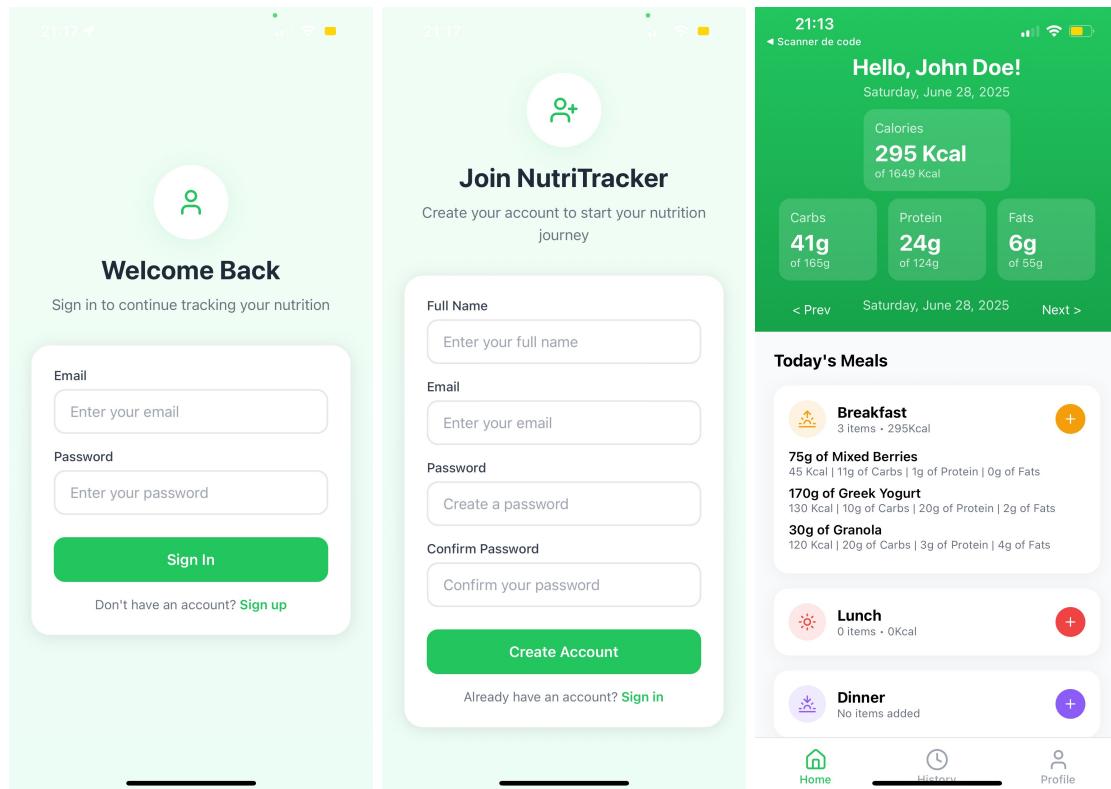


FIGURE 6 – Captures d'Écran Authentification et Page d'accueil

## A.2 Ajout de repas

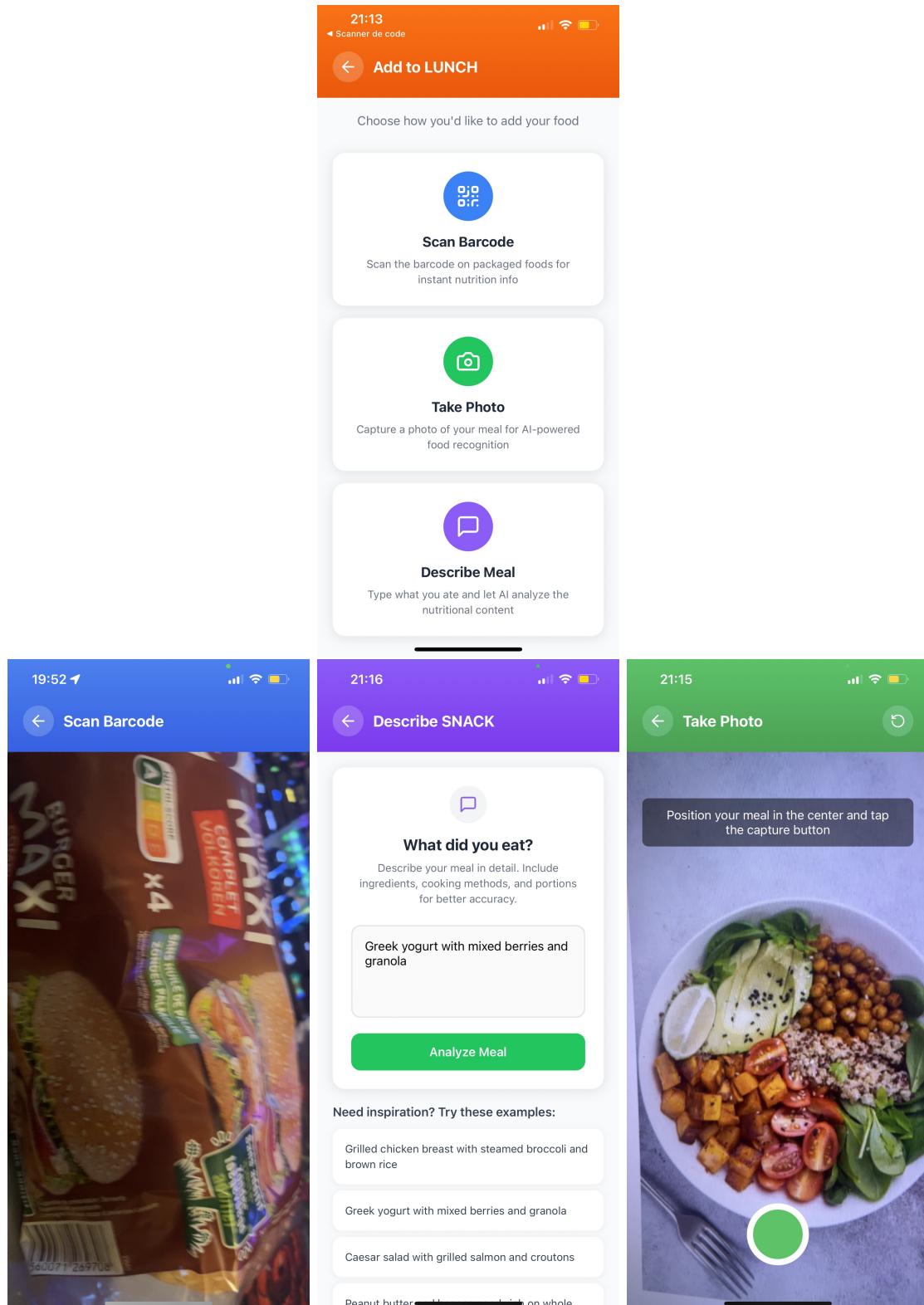


FIGURE 7 – Captures d’Écran Choix de type d’ajout de repas

### A.3 Résultats de l'analyse

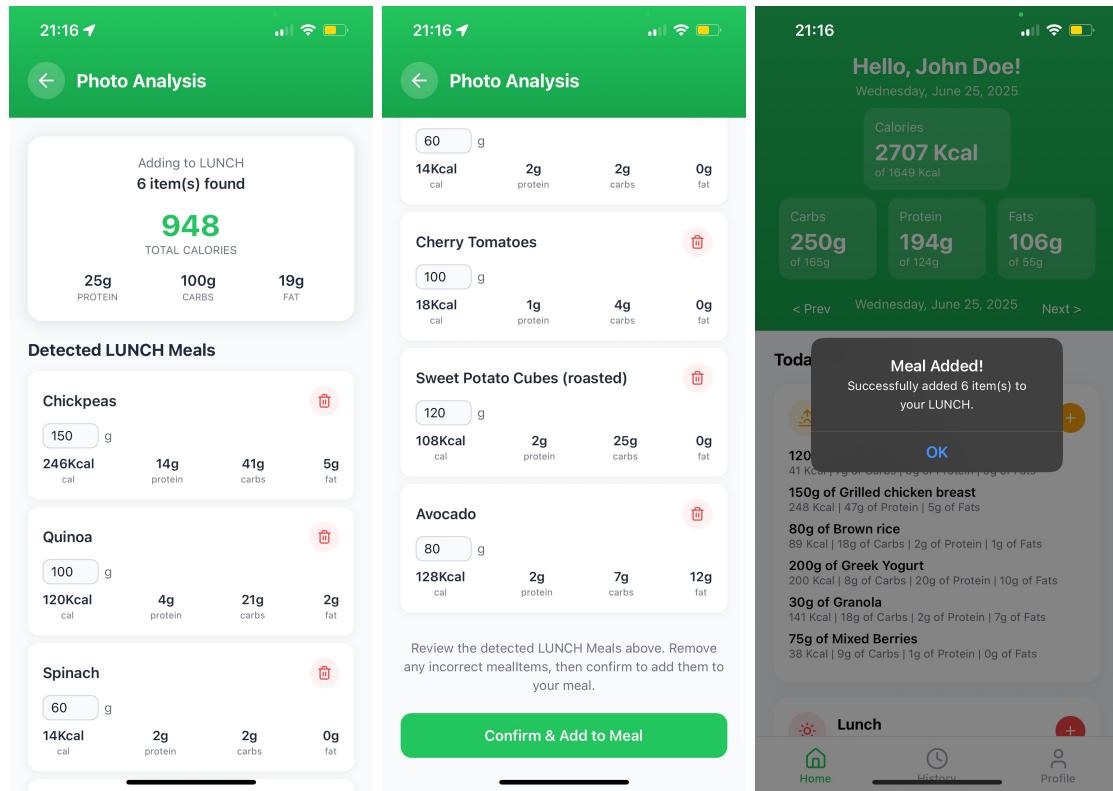


FIGURE 8 – Captures d’Écran Résultats de l’analyse

## A.4 Bibliothèque Exercice Histoire

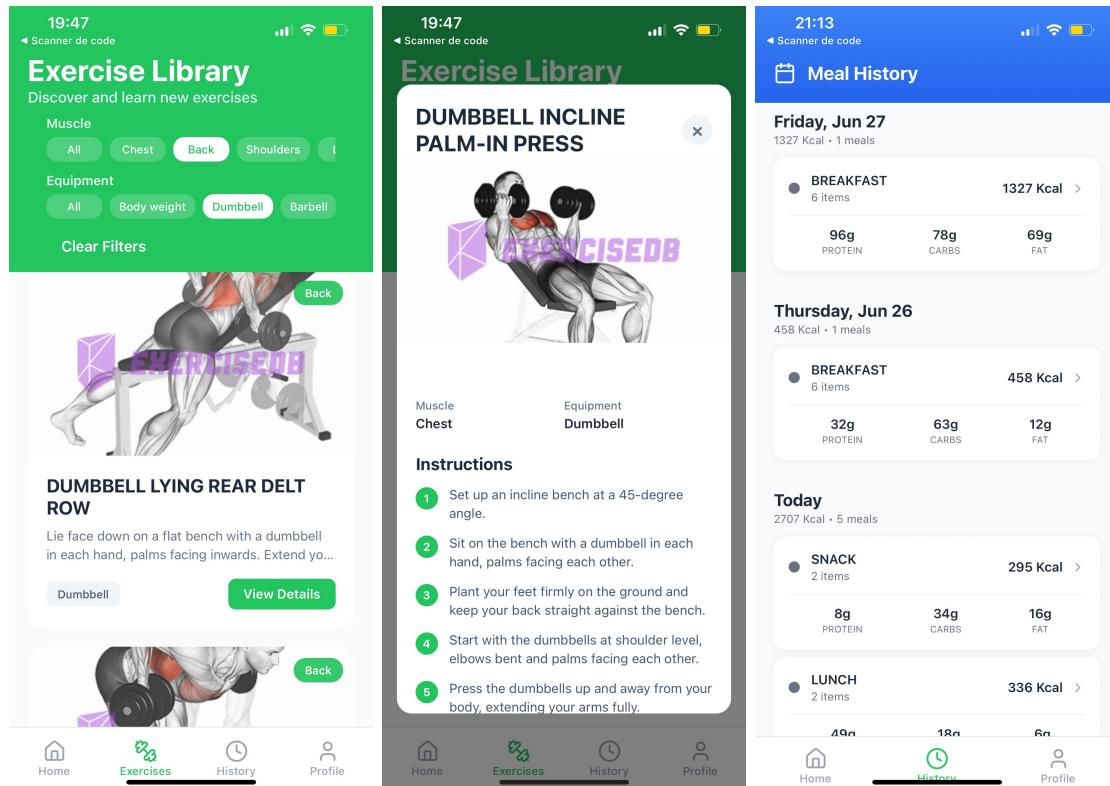


FIGURE 9 – Captures d’Écran Bibliothèque Exercice Histoire

## A.5 Profil et sélection de la langue

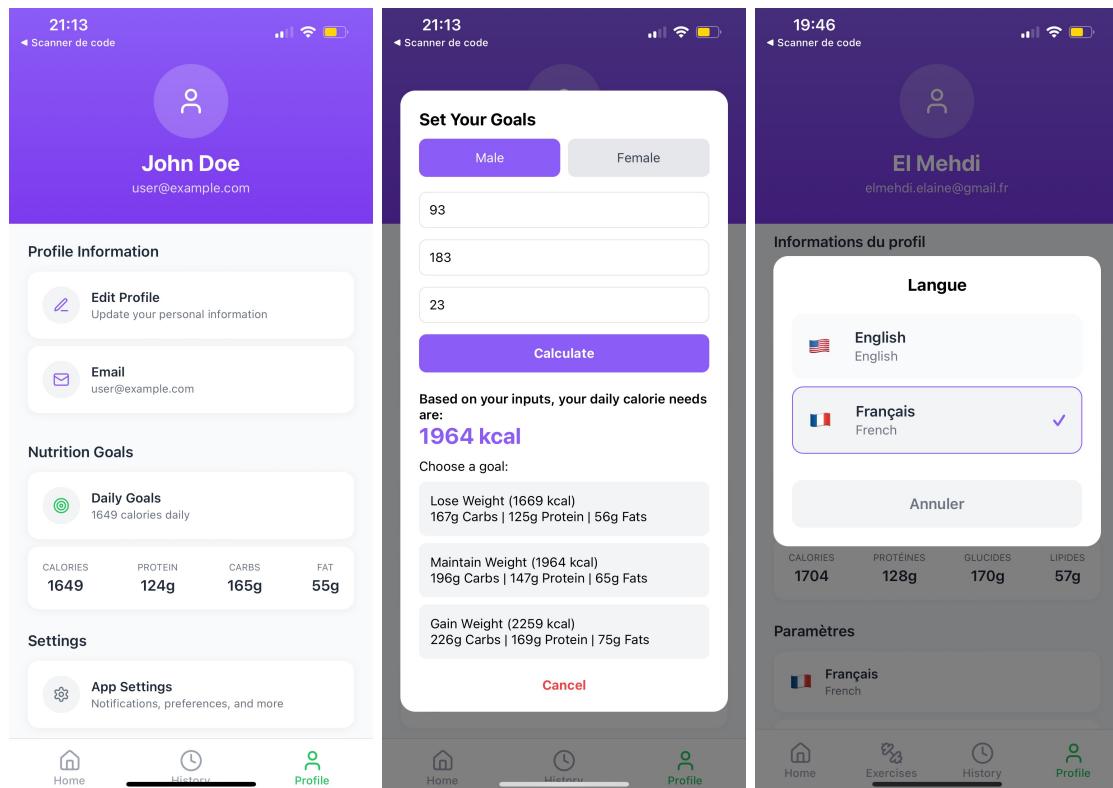


FIGURE 10 – Captures d’Écran Profil et sélection de la langue